



TESINA DE LICENCIATURA

Título: Técnicas de minería de procesos de negocio distribuidos con Bonita OS

Autores: Federico Gastón Madrid (8794/6)

Director: Dra. Patricia Bazán

Asesor profesional: Lic. José Nicolás Martínez Garro

Carrera: Licenciatura en Sistemas – Plan 2007

Resumen

La madurez de los BPMS (Business Process Management Systems) ha hecho que las organizaciones se vuelquen cada vez con más confianza hacia modelos de gestión por procesos de negocio.

La minería de procesos (Process Mining) es una poderosa tecnología para administrar procesos operacionales no triviales, cuyo objetivo es extraer conocimiento de los registros de eventos obtenidos de distintos sistemas de información. Así, constituye un buen soporte para cerrar el ciclo de mejora continua que propone BPM.

Por otra parte Cloud Computing ofrece almacenamiento y cómputo distribuido bajo demanda, permitiendo descomponer un proceso y ejecutar cada una de sus partes de manera distribuida.

La minería de procesos distribuidos presenta un desafío importante de abordar y de validar este análisis aplicado a un BPMS concreto. En este trabajo se enuncia un enfoque para la aplicación de Process Mining sobre procesos distribuidos utilizando Bonita Open Solution como BPMS.

Palabras Claves

- Process Mining
- Business Intelligence
- BPM
- Bonita Open Solution
- Web Services
- Cloud Computing
- Procesos distribuidos
- XES (eXtensible Event Stream)

Conclusiones

El desarrollo de Web Services que nos permiten obtener el registro de eventos de un proceso distribuido, materia prima para la aplicación de técnicas de minería de procesos, nos dan la posibilidad de seguir aplicando dichas técnicas en entornos distribuidos y así continuar analizando y optimizando procesos de negocio independientemente de dónde estén desplegados y ejecutándose.

Trabajos Realizados

Desarrollo de Web Services para obtener el registro de un proceso distribuido.

Extensión de la herramienta de monitoreo para interactuar con dichos Web Services.

Implementación un proceso de negocio como caso de estudio en la herramienta Bonita Open Solution para luego descomponerlo en varios servidores, obtener su registro de eventos y aplicarle las técnicas de Process Mining con ProM.

Trabajos Futuros

Generar un único registro de eventos enriquecido que contenga meta-datos de las tareas y servidores para permitirnos realizar análisis más completos de la situación distribuida.

Permitir pre-filtrar por algún criterio (tiempo, actor) el registro para optimizar flujos de red y uso de recursos por parte de la herramienta de minería.



Facultad de Informática
Universidad Nacional de La Plata

Técnicas de minería de procesos de negocio distribuidos con Bonita OS

Tesina de grado para Licenciatura en Sistemas

Alumno: Federico Madrid
Directora: Dra. Patricia Bazán
Asesor Profesional: Lic. José Nicolás Martínez Garro

Agradecimientos

Quisiera agradecer a mi familia, principalmente a mis abuelos Nelva y Vicente a quienes siento siempre conmigo y deseaban más que yo que lograra esto, lo comparto con ellos de manera muy especial. A mis padres Alicia y Jorge que me brindaron su amor, apoyo, contención y entendimiento en los momentos más difíciles de éste largo camino, sin ellos no lo hubiese logrado por lo que considero esto un logro compartido. A mis hermanos Francisco y Facundo, quienes estuvieron cada uno desde su lugar apoyándome durante todo este tiempo. A mis tíos Raúl, Ana, Susana y Chachi quienes siempre estuvieron pendientes y al servicio de lo que necesitaba. Todos colaboraron para que hoy llegue hasta acá y les voy a estar agradecido siempre.

A Patricia Bazán y José Martínez Garro, por su orientación, paciencia y buena predisposición que me ayudaron a realizar este trabajo.

Mis amigos del secundario: Conra, Bona, Mauro, Cabe, Mati, Rumi, Flor, Pepa, Lau y de la facultad: Ale, Facu, Aníbal y Juan, quienes estuvieron conmigo en todo momento brindándome su amistad y apoyo incondicional cuando más lo necesitaba.

Mis compañeros y amigos de trabajo Juli, Leo, Fulvio, Emi, Hunter, Virginia, Marcelo y Luciano con quienes crecí tanto a nivel profesional como personal.

Finalmente, a la Facultad de Informática y a la Universidad Nacional de La Plata que me dieron la oportunidad de estudiar esta carrera.

Índice General

Capítulo 1 - Introducción	4
1.1 - Motivación	5
1.2 - Estructura	6
Capítulo 2 - BPM y BPMS	7
2.1 - Business Process Management (BPM)	7
2.2 - Ciclo de vida BPM	7
2.3 - BPMS (Business Process Management Systems)	9
2.4 - Capacidades claves de un BPMS	10
Capítulo 3 - Process Mining	12
3.1 - Process Mining y definiciones	12
3.2 - Tipos de minería de procesos	12
3.3 - Dentro del ciclo de vida de BPM	15
3.4 - Relación modelo-realidad	17
3.5 - Business Intelligence y Process Mining	19
3.6 - Modelado y análisis de proceso	20
3.7 - Minería de datos	22
3.8 - Registros de eventos	24
3.9 - Descubrimiento de procesos	26
3.10 - Chequeo de concordancia	28
3.11 - Mejora y extensión	30
Capítulo 4 - Cloud Computing y BPaaS	32
4.1 - Cloud Computing	32
4.2 - Características esenciales	33
4.3 - Modelos de servicio	34
4.4 - Procesos de negocio como servicio	36
4.5 - BPMS en ambientes distribuidos	37
4.6 - Combinación de esquema embebido y cloud	38
4.7 - Descomposición de procesos	39
4.8 - Distribución óptima de las actividades	42
4.9 - Monitoreo de procesos de negocio distribuidos	43
Capítulo 5 - Software a utilizar y mecanismos de integración	45
5.1 - Bonita Open Solution	45
5.1.1 - Módulos de Bonita OS	46
5.1.2 - Capacidad de extensión de Bonita	47
5.1.3 - Conectores de Bonita OS	48
5.1.4 - API REST de Bonita Open Solution	50
5.2 - Proceso generador de log de eventos	51
5.3 - Sistema de ejecución y monitoreo de procesos de Bonita OS	52

5.4 - ProM	53
Capítulo 6 - Solución propuesta	55
Capítulo 7 - Validación con un caso de estudio	63
7.1 - Criterios de descomposición	74
7.2 - Análisis de minería del proceso con ProM	77
7.2.1 - Dotted Chart	80
7.2.2 - Minería de actores	84
7.2.3 - Diagramas de transición	86
Capítulo 8 - Conclusiones y trabajos futuros	97
Referencias bibliográficas	99

Capítulo 1 - Introducción

En la actualidad las organizaciones usan tecnologías de información para soportar sus procesos de negocio siendo BPM (Business Process Management o Gestión de procesos de negocio) la tecnología pionera [1].

Como consecuencia se necesitan herramientas que permitan analizar y optimizar dichos procesos y ayudar en la toma de decisiones. Aquí aparece Business Intelligence (BI) cuyo objetivo es el de generar información y conocimiento útil para el proceso de toma de decisiones complejas haciendo uso de los datos disponibles que deja el sistema a lo largo de su uso. Dentro de BPM el análisis de los rastros de ejecución conforman un punto clave dentro del ciclo de mejora continua. Técnicas de aplicación de BI son: BAM (Monitoreo de actividades de negocio), CPM (Gestión del rendimiento corporativo), CPI (Mejora continua de procesos), y BPI (Inteligencia de procesos de negocio) que permiten realizar reportes y tableros de mando. Luego para el análisis de cuestiones que son propias del proceso, los BPMS utilizan modelos de proceso para analizar los procesos operacionales. Pero estos modelos no son suficientes dado que se basan en un modelo idealizado y no en los datos de los eventos que ocurren con la ejecución de las instancias. Es en este escenario en el que Process Mining viene a dar una solución integral a la aplicación de Business Intelligence a BPM combinando datos de eventos y modelos de proceso [5] [4] [6] [7].

La idea básica de Process Mining es extraer conocimiento del registro de eventos producidos por los sistemas de información, aplicando las técnicas:

- Técnicas de descubrimiento de proceso.
- Técnicas de chequeo de concordancia del proceso con la realidad.
- Técnicas de extensión y mejora del proceso.

Para lograr aplicar las distintas técnicas que nos brinda Process Mining debemos generar el registro de eventos en un formato específico que pueda ser procesado por una herramienta de minería de procesos. Conociendo la ubicación de los datos de interés, ya sea en la base de datos del BPMS (si ya contamos con un proceso desplegado) o la/s base/s de datos propia del sistema de información de la empresa, se genera el registro. En este punto es donde XES estandariza el modelo de dicho registro.

Por otro lado, la decisión de utilizar un BPMS en una organización para dar soporte a sus procesos de negocio puede ser arriesgada desde el punto de vista de la inversión que debe realizarse en software y hardware, así como en la capacitación de los recursos humanos para el despliegue y mantenimiento de dicho sistema. La escalabilidad también se ve comprometida, dado que el motor de procesos es capaz de atender simultáneamente una cantidad limitada de instancias de procesos [9].

Cloud Computing aporta una solución brindando servicios de computación bajo demanda con una alta fiabilidad, escalabilidad y disponibilidad en un entorno distribuido. El poder de cómputo, almacenamiento de datos y servicios se contratan en una modalidad de “pago por uso”, evitando una gran inversión en software, hardware y mantenimiento, BPM basado en Cloud Computing (conocido como Business Process as a Service – BPaaS) es un modelo de

servicio en el cloud donde las aplicaciones que se ofrecen son del tipo procesos de negocio o workflows.

1.1 - Motivación

La motivación de este trabajo radica en la investigación de las tecnologías Process Mining y Cloud Computing en forma conjunta para poder identificar una arquitectura integradora de ambas partes, permitiendo así que se puedan adaptar en forma adecuada y por consiguiente aprovechar las funcionalidades que ofrece cada una de ellas. Conocemos bien lo poderosas que son ambas tecnologías para ofrecer soluciones en el campo de la mejora continua y la descomposición y escalabilidad de cómputo respectivamente, pero suelen analizarse una desconectada de la otra. Si bien la minería de procesos ya lleva un tiempo aplicándose en empresas que deciden orientar su funcionamiento a BPM, el cómputo distribuido está cada vez más accesible con mejores prestaciones, menores costos y mayor poder de personalización y, por lo tanto, con más chances que las empresas adopten este paradigma de procesamiento distribuido. Este nuevo escenario presenta un desafío para las clásicas técnicas de minería que dado el caso en que sus procesos estén distribuidos, sea más complejo recolectar la información histórica total y completa para poder seguir brindando sus servicios de manera transparente a en qué lugar físico estén ejecutando cada una de sus partes. Es por estos motivos que tanto el análisis arquitectónico, como la creación de herramientas que permitan la aplicación de técnicas de minería de procesos en entornos distribuidos es de alta relevancia en este ambiente empresarial cada vez más orientado a la nube.

Dejando el marco teórico y pasando al plano práctico, contamos con herramientas en ambas ramas. Por un lado tenemos un proceso implementado en Bonita Open Solution que permite obtener el log de eventos de un proceso dado en el formato estándar XES, lo cual es posible ya que dicho BPMS permite el acceso a su información histórica mediante su API[18]. El formato no es un detalle menor ya que este es soportado por la herramienta que utilizaremos para la aplicación de técnicas de minería de procesos: ProM.

En el otro rincón tenemos una aplicación web PHP que permite la ejecución y monitoreo de procesos distribuidos en BonitaOS [19]. Mediante el uso de la API REST de Bonita, una aplicación recolecta la información de los procesos y permite su monitoreo activo.

Dado este escenario tecnológico, este trabajo propone la extensión de dicha herramienta de monitoreo para poder obtener los logs de eventos de dichos procesos distribuidos y aplicarles técnicas de minería de procesos.

1.2 - Estructura

La estructura de este trabajo está dividida en los 8 capítulos en los que trataremos los siguientes temas:

- Capítulo 1 - Introducción: breve resumen de la situación, motivación y estructura del trabajo
- Capítulo 2 - BPM y BPMS: se verá el marco conceptual de BPMS y de las características principales de los mismos que se deben tener en cuenta a la hora de la integración de las tecnologías.
- Capítulo 3 - Process Mining: se verán los conceptos de la minería de procesos, su diferencia con Business Intelligence, su participación en el ciclo de vida BPM, se analizarán los registros de eventos que sirven como fuente de información y las distintas técnicas de minería.
- Capítulo 4 - Cloud Computing y BPaaS: se verán las características de la computación en la “nube” y su orientación a BPM que da como resultado lo que se conoce como Business Process as a Service. Más adelante se tratarán temas relacionados a los procesos distribuidos como descomposición y monitoreo.
- Capítulo 5 - Software a utilizar y mecanismos de integración: se detallarán las principales características del BPMS Bonita Open Solution, un proceso generador de eventos y un sistema de ejecución y monitoreo de procesos también de BonitaOS y la herramienta de minería de procesos ProM.
- Capítulo 6 - Solución propuesta: se detallará la extensión de la herramienta de monitoreo para proveer la funcionalidad de obtención del log de eventos de un proceso distribuido mediante el uso de servicios web en PHP.
- Capítulo 7 - Validación con un caso de estudio: se plantea el detalle de un caso de estudio como ejemplo de obtención del log de eventos de un proceso distribuido y su posterior análisis en ProM
- Capítulo 8 - Conclusiones y trabajos futuros: se detallan los conflictos y resoluciones tomadas durante la investigación y el desarrollo , conclusiones obtenidas y trabajos hacia el futuro.

Capítulo 2 - BPM y BPMS

La Gestión de Procesos de Negocio (en inglés: Business Process Management o B.P.M.) es una metodología corporativa y disciplina de gestión, cuyo objetivo es mejorar el desempeño (eficiencia y eficacia) y la optimización de los procesos de negocio de una organización, a través de la gestión de los procesos que se deben diseñar, modelar, organizar, documentar y optimizar de forma continua. Por lo tanto, puede ser descrito como un proceso de optimización de procesos. [1][2][3]

En este capítulo se introducirán los conceptos básicos de BPM y BPMS que nos servirán de base para abordar los temas posteriores de Process Mining y Cloud Computing.

2.1 - Business Process Management (BPM)

Un proceso de negocio consiste en un conjunto de actividades que son realizadas en forma coordinada en un entorno organizacional y técnico, cumpliendo una función específica del dominio. Cada proceso de negocio es implementado a lo largo de una única organización pero puede llegar a interactuar con otros procesos implementados en distintas organizaciones.

La gestión de procesos de negocio incluye conceptos, métodos y técnicas que incluyen el diseño, administración, configuración y análisis de procesos, partiendo de la restricción de ejecución entre los mismos y la representación explícita de sus actividades. Una vez que los procesos son definidos, estos pueden ser sujetos a análisis y mejora.

Un modelo de proceso de negocio consiste en un conjunto de actividades modelo y restricciones de ejecución entre las mismas. Una instancia de proceso representa un caso concreto en el dominio operacional de una compañía, esta consiste en instancias de actividades. Los modelos de procesos son los elementos principales a la hora de la puesta en funcionamiento, que puede ser realizada a través de la ejecución de reglas y políticas organizacionales o utilizando un sistema de gestión de procesos de negocio. [1]

2.2 - Ciclo de vida BPM

Como se muestra en la Figura 2.1, el ciclo de vida de los procesos de negocio se compone de fases cíclicas que poseen una dependencia lógica, comenzando con la de diseño y análisis en la que se realizan entrevistas sobre los procesos de negocio organizacionales y del ambiente técnico. En base a los datos recolectados, los procesos de negocio son identificados, revisados, validados y representados por modelos explícitos que son expresados en notaciones gráficas que facilitan la comunicación sobre estos procesos de tal forma que los diferentes involucrados puedan comunicarse en forma eficiente para así poder redefinirlos y mejorarlos. En esta etapa ya podemos notar que conceptos relativos a la distribución de procesos tales como descomposición (agrupar tareas creando subprocessos) y coreografía (interacción de procesos entre sí) entran en juego. Es aquí donde se toman decisiones sobre en qué partes se va a dividir el proceso completo y que se van a comunicar, independientemente de la herramienta tecnológica que se decida utilizar.

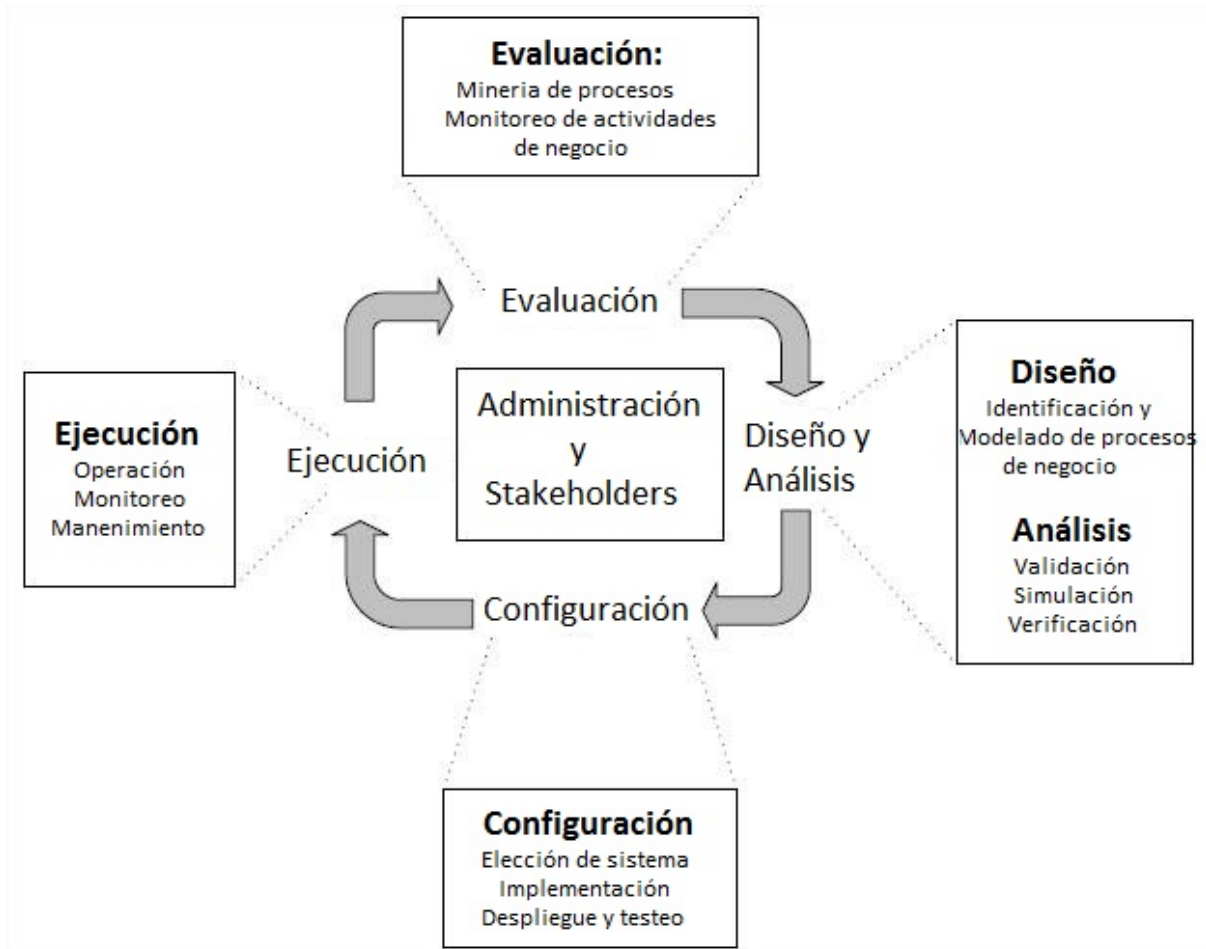


Figura 2.1. Ciclo de vida de los procesos de negocio [1]

Una vez que el modelo de proceso de negocio es diseñado y validado se pasa a la fase de configuración. Existen diferentes formas de realizar tal etapa, se puede implementar mediante un conjunto de políticas y procedimientos que los empleados de la empresa necesitan cumplir o puede ser ejecutada mediante un sistema gestor de procesos de negocio. Eligiendo como alternativa esta última, se deberá elegir una plataforma de implementación. El modelo de proceso de negocio se verá incrementado por información técnica que facilita su representación por el sistema gestor de procesos de negocio, que requiere ser configurado acorde al ambiente organizacional de la empresa y al proceso de negocio cuya representación debe controlar. Esta configuración incluye interacciones de los empleados con el sistema así como también la integración de los sistemas existentes con el mismo. En esta etapa siguen entrando los conceptos que remarcamos en la etapa previa (descomposición y coreografía) pero ya desde el punto de vista tecnológico. Quiere decir que definimos, modelamos e implementamos dichos subprocesos en los cuales decidimos descomponer el proceso original pero esta vez sobre un BPMS concreto, teniendo en cuenta las facilidades y restricciones que nos plantea la herramienta en sí. Luego, establecemos las interfaces de comunicación entre dichas partes, definiendo qué tecnología usaremos (Web Services, API REST, etc) e implementando los métodos necesarios para lograrla.

Una vez cumplida la fase de configuración las instancias de procesos de negocio pasan por la etapa de representación, es decir, pasan a ser ejecutadas. Iniciar una instancia normalmente

se produce por un evento definido como puede ser el recibir una orden por un cliente. El BPMS controla activamente la ejecución de las instancias de los procesos de negocio conforme a como están definidas en el modelo, para esto se requiere una correcta orquestación. Un componente de monitorización de un sistema gestor de procesos de negocio visualiza el estado de las instancias. La monitorización es un mecanismo importante para proveer información acertada del estado de nuestros procesos, tal información es de utilidad por ejemplo para responder a pedidos del cliente que necesita conocer el estado actual de una instancia en curso. Durante la fase de representación se recolectan datos relevantes de la ejecución de las instancias, normalmente en alguna forma de archivo de log, estos archivos consisten en conjuntos ordenados de entradas que indican eventos que ocurrieron durante la ejecución de las instancias. El inicio y fin de las actividades produce información que es almacenada en logs de ejecución. Estos archivos de log son vitales para el campo de estudio que plantea este trabajo, ya que son nuestro principal recurso para aplicar las técnicas de minería de procesos. Decidir qué variables y en qué formato se va a guardar afectará de forma directa a los resultados que podremos llegar a obtener luego y, por lo tanto, qué conclusiones estaremos en condiciones de tomar.

La fase de evaluación utiliza información disponible para evaluar y mejorar los modelos de procesos de negocio y sus implementaciones. Los logs de ejecución son analizados utilizando monitorización de actividades del negocio y técnicas de process mining. Estas técnicas apuntan a identificar la calidad de los modelos de proceso de negocio y la adecuación del entorno de ejecución. Por ejemplo una actividad de monitoreo de proceso de negocio puede identificar que una cierta actividad toma demasiado tiempo debido a la escasez de recursos requeridos para llevarlos a cabo. El objetivo es obtener un seguimiento de los procesos, incluyendo información sobre los actores que intervienen, los tiempos involucrados y además otorgando la posibilidad de descubrir nuevos procesos, controles e información. Dado que la información es útil también para la simulación de procesos de negocio, estos pasos están estrechamente relacionados. Esta es la fase en la que más está enfocada esta investigación. Aquí es donde se aplica específicamente la minería de procesos, tomando la información desde los archivos de logs que fueron dejando las distintas ejecuciones de lo procesos en la etapa previa. Es en esta instancia donde nos topamos con conceptos tales como la mejora continua que buscan, en base a resultados ya obtenidos, lograr una mejor performance (tanto a nivel tecnológico como de dominio).

Al tratarse de un ciclo, como observamos en la Figura 2.1, este final (en caso de determinar cuáles son las mejoras que se pueden aplicar) vuelve a disparar el inicio de una nueva vuelta a la etapa de análisis y diseño, esta vez para la versión mejorada. [1]

2.3 - BPMS (Business Process Management Systems)

La adopción de BPM y técnicas de administración de procesos por las organizaciones ha originado un fuerte interés en el mundo tecnológico para desarrollar productos y soluciones para soportar BPM. La tecnología BPM ha madurado al punto donde la administración de procesos en tiempo real es posible. Los BPMS representan un quiebre en el uso e implementación de sistemas de información. La metodología de implementación de los sistemas tradicionales se focaliza en funciones y objetos. Los procesos se relegan al flujo de

trabajo, que por lo general no recibe mayor atención durante la implementación. Los BPMS rompen con la mentalidad de diseño de objetos. Pone al proceso en un foco central de diseño de soluciones. Esto alinea a las soluciones de IT a estar más en línea con la realidad de proceso para las organizaciones de BPM. Además de las funciones de diseño, un BPMS, permite que una vez implementado se puedan medir, monitorear, controlar y analizar el tiempo real de los procesos. En pocas palabras, los BPMS sirven como el centro de control sobre las personas, aplicaciones de empresa, y datos. Como centro de control los BPMS reciben datos en tiempo real de todas las tareas que se realizan en los procesos que controla. Es posible el monitoreo con la información de proceso en tiempo real. BPMS puede detectar variaciones en el proceso, una vez que se detecta una variación, un BPMS tiene la capacidad de resolver esa variación. Las mediciones en el proceso de negocio proveen datos de proceso que se pueden analizar. El análisis de los datos de proceso es un paso esencial para identificar qué proceso debe mejorar y las mejoras a qué área pueden traer mayor valor al proceso.[3]

2.4 - Capacidades claves de un BPMS

Los BPMS son una clase de software que permite a las organizaciones crear soluciones de tecnología de la información centradas en el proceso. Que se centre en el proceso quiere decir que las soluciones de un BPMS permiten integrar personas, sistemas y datos. Las organizaciones que utilicen un BPMS para llevar a cabo el cambio de procesos de negocio habilitado por IT ganaran las siguientes capacidades:

- Involucrarse mejor en el negocio en el diseño de soluciones de procesos de negocio habilitado por IT.
- Habilidad de integrar personas y sistemas que participan en los procesos de negocio.
- Habilidad de simular procesos de negocio para diseñar el proceso más óptimo para la implementación.
- Habilidad de monitorear, controlar y mejorar procesos de negocio en tiempo real.
- Habilidad de efectuar cambios en procesos de negocio existentes en tiempo real sin esfuerzo de conversión de proceso elaborado.

BPMS permite que los propietarios de los procesos de negocio se involucren directamente con la solución IT. El diseñador de procesos que provee un BPMS, es una herramienta que permite a los propietarios de los procesos de negocio o a los analistas de negocio diseñar los procesos de negocio en detalle. Los diseñadores de procesos de negocio pueden generar automáticamente el código que a veces se puede desplegar sin ayuda del departamento de IT. En el caso en el que se requiera un desarrollo, el BPMS ya contiene el significado del proceso y las definiciones de la solución. Los desarrolladores de IT pueden utilizar el mismo proceso diseñado por el personal de negocios, y embeber la lógica al proceso utilizando el diseñador. Esto se puede realizar utilizando el lenguaje de script que viene empaquetado con el diseñador de procesos o la herramienta de desarrollo en un lenguaje de programación utilizado extensamente. Tanto el personal de negocio como el personal de IT trabajan desde el mismo diseño utilizando la misma herramienta. Esto reduce la brecha comunicacional entre IT y los negocios. [3]

Ahora que tenemos una mejor idea sobre los conceptos de BPM y BPMS pasemos a analizar cómo Process Mining apunta hacia la mejora continua en base a los “rastros” que dejan los procesos ejecutados.

Capítulo 3 - Process Mining

Ante la creciente situación en la que los sistemas de información se entrelazan cada vez más con los procesos operacionales que soportan, se cuenta a su vez con cada vez más eventos registrados en dichos sistemas de información. No obstante, a las organizaciones les cuesta extraer valor (lo que denominamos información) de dichos datos. Es sobre este aspecto que Process Mining viene a dar soporte, permitiendo obtener información relacionada con el proceso en base a los eventos registrados en el sistema. Mediante la aplicación de técnicas que nos provee la minería de procesos se pueden descubrir procesos, chequear la concordancia del proceso con la realidad y extender y mejorar un proceso.

El crecimiento del universo digital que está bien alineado con los procesos en una organización hace posible almacenar y analizar eventos. El desafío está en explotar los datos de manera significativa, por ejemplo para proveer puntos de vista, identificar cuellos de botella, anticipar problemas, registrar violaciones de políticas, sobre todo lo dicho anteriormente se encarga Process Mining. [5]

En este capítulo se analizará la minería de procesos y nos permitirá ver su importancia dentro del ciclo BPM.

3.1 - Process Mining y definiciones

Process Mining es una disciplina de investigación joven que se encuentra entre el aprendizaje automático y la minería de datos por un lado y el modelado de procesos y análisis por el otro. Su propósito es descubrir, monitorear, y mejorar los procesos reales, mediante la extracción de conocimiento del log de eventos que ya se encuentra disponible en los sistemas de hoy.

Las capacidades en expansión de los sistemas de información y otros sistemas que dependen de la computación, están bien caracterizadas por la ley de Moore. Gordon Moore, el co-fundador de Intel, vaticinó en 1965 que el número de componentes en los circuitos integrados se duplicaría todos los años. Durante los últimos 50 años el crecimiento ha sido de hecho exponencial, si bien es cierto que a un paso ligeramente más lento. Estos avances resultaron en un crecimiento espectacular del “universo digital” (i.e., todos los datos almacenados y/o intercambiados electrónicamente). Además, el universo digital y el real continúan acercándose a estar más y más alineados. [26]

3.2 - Tipos de minería de procesos

Como se muestra en la Figura 3.1, los registros de eventos pueden ser utilizados para realizar tres tipos de minería de procesos. El primer tipo de minería de procesos es el descubrimiento. Una técnica de descubrimiento toma un registro de eventos y produce un modelo sin usar ninguna información a-priori. El descubrimiento de procesos es la técnica de minería de procesos más destacada. Para muchas organizaciones es sorprendente ver que las técnicas existentes son realmente capaces de descubrir los procesos reales meramente basado en las muestras de ejecución en los registros de eventos. El segundo tipo de minería de procesos es la conformidad. Aquí, se compara un modelo de proceso existente con un registro de eventos

del mismo proceso. La verificación de conformidad puede ser usada para chequear si la realidad, tal como está almacenada en el registro de eventos, es equivalente al modelo y viceversa. Note que distintos tipos de modelos pueden ser considerados: la verificación de conformidad puede ser aplicada a modelos procedurales, modelos organizacionales, modelos de procesos declarativos, políticas/reglas de negocio, regulaciones, etc. El tercer tipo de minería de procesos es el mejoramiento. Aquí, la idea es extender o mejorar un modelo de proceso existente usando la información acerca del proceso real almacenada en algún registro de eventos. Mientras la verificación de conformidad mide el alineamiento entre el modelo y la realidad, este tercer tipo de minería de procesos busca cambiar o extender el modelo a-priori. Por ejemplo, al usar marcas de tiempo en el registro de eventos, uno puede extender el modelo para mostrar cuellos de botella, niveles de servicio, tiempos de procesamiento, y frecuencias. [26]

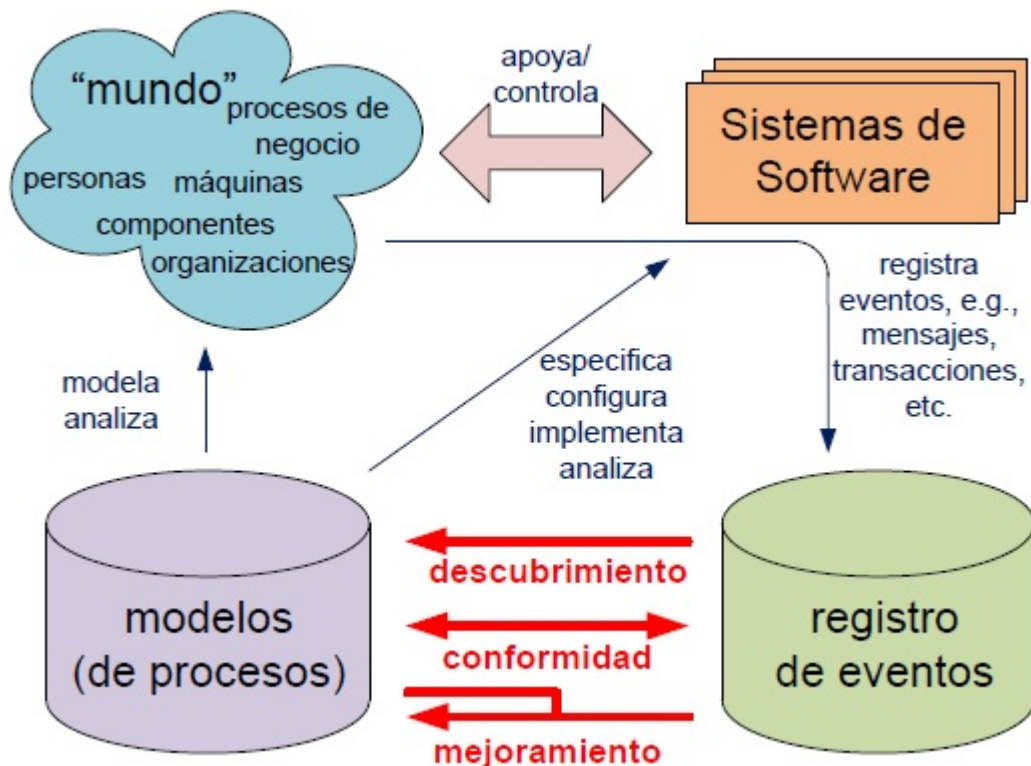


Figura 3.1. Posicionamiento de los tres tipos principales de minería de procesos: (a) descubrimiento, (b) verificación de conformidad, y (c) mejoramiento. [26]

La Figura 3.2 describe los tres tipos de minería de procesos en términos de entradas y salidas. Las técnicas para descubrimiento toman un registro de eventos y producen un modelo. El modelo descubierto es típicamente un modelo de proceso (e.g., una red de Petri, un BPMN, un EPC, o un diagrama de actividad UML), sin embargo, el modelo podría también describir otras perspectivas (e.g., una red social). Las técnicas de verificación de conformidad necesitan un registro de eventos y un modelo como entrada. La salida consiste en información de diagnóstico mostrando las diferencias y elementos en común entre el modelo y el registro

de eventos. Las técnicas para mejoramiento de modelos (reparar o extender) también necesitan un registro de eventos y un modelo como entrada. La salida es un modelo mejorado o extendido.[26]

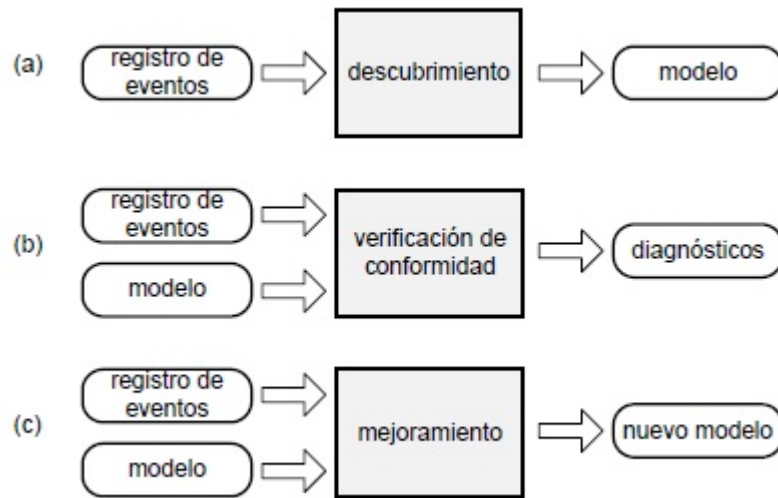


Figura 3.2. Los tres tipos básicos de minería de procesos explicados en términos de entradas y salidas: (a) descubrimiento, (b) verificación de conformidad, y (c) mejoramiento. [26]

A su vez, a los tres tipos de minería de procesos antes enunciados se les puede identificar tres tipos de perspectivas. La perspectiva de control de flujo se enfoca en el control de flujo, i.e., el orden de ejecución de las actividades. El objetivo de explorar esta perspectiva es encontrar una buena caracterización de todos los caminos posibles. El resultado se expresa típicamente en términos de una red de Petri o alguna otra notación de procesos (e.g., EPCs, BPMN, o diagramas de actividad UML). actores (e.g., personas, sistemas, o departamentos) están involucrados y cómo se relacionan. El objetivo es ya sea estructurar la organización clasificando a las personas en términos de roles y unidades organizacionales, o mostrar la red social. La perspectiva de casos se enfoca en las propiedades de los casos. Obviamente, un caso puede ser caracterizado por su ruta en el proceso o por los actores que trabajan en él. Sin embargo, los casos también pueden ser caracterizados por los valores de los correspondientes elementos de datos. Por ejemplo, si un caso representa un pedido de reposición, podrá ser interesante conocer el proveedor o la cantidad de productos solicitados. La perspectiva de tiempo se relaciona con la ocurrencia y frecuencia de los eventos. Cuando los eventos tienen asociados marcas de tiempo, es posible descubrir cuellos de botella, medir niveles de servicio, monitorear la utilización de recursos, y predecir el tiempo de procesamiento restante de casos en ejecución.[26]

Hay algunas ideas erradas en relación a minería de procesos. Algunos proveedores, analistas e investigadores limitan el alcance de la minería de procesos a una técnica especial de minería de datos para el descubrimiento de procesos que puede sólo ser usada para análisis offline. Esto no es así, por lo tanto, enfatizamos las siguientes tres características.

- La minería de procesos no está limitada al descubrimiento del control de flujo. El descubrimiento de modelos de procesos desde los registros de eventos llena la

imaginación tanto de profesionales como de académicos. Por lo tanto, el descubrimiento del control de flujo es a menudo visto como la parte más emocionante de la minería de procesos. Sin embargo, la minería de procesos no se limita al descubrimiento del control de flujo. Por una parte, el descubrimiento es solo una de las tres formas básicas de minería de procesos (descubrimiento, conformidad, y mejoramiento). Por otra parte, el alcance no está limitado al control de flujo; las perspectivas organizacionales, de casos y de tiempo también cumplen un rol importante.

- La minería de procesos no es solo un tipo específico de minería de datos. La minería de procesos se puede ver como el “eslabón perdido” entre la minería de datos y el BPM tradicional basado en modelos. La mayoría de las técnicas de minería de datos no están en absoluto centradas en procesos. Los modelos de proceso que potencialmente exhiben concurrencia son incomparables a las estructuras de minería de datos simples, tales como los árboles de decisión y las reglas de asociación. Por lo tanto, se necesitan nuevos tipos de representación y de algoritmos.
- La minería de procesos no está limitada al análisis offline. Las técnicas de minería de procesos extraen conocimiento de los datos de eventos históricos. Aunque se utilizan datos “post mortem”, los resultados pueden ser aplicados a casos en ejecución. Por ejemplo, se puede predecir el tiempo de finalización de un pedido del cliente parcialmente realizado, usando un modelo de proceso descubierto. [26]

3.3 - Dentro del ciclo de vida de BPM

Para posicionar la minería de procesos, usamos el ciclo de vida de BPM mostrado en la Figura X. El ciclo de vida de BPM muestra las siete fases de un proceso de negocio y sus correspondientes sistemas de información. En la fase de (re)diseño se crea un nuevo modelo de proceso o se adapta un modelo de proceso existente. En la fase de análisis se analiza un modelo candidato y sus alternativas. Después de la fase de (re)diseño, se implementa el modelo (fase de implementación) o se (re)configura un sistema existente (fase de (re)configuración). En la fase de ejecución se ejecuta el modelo diseñado. Durante la fase de ejecución el proceso es monitoreado. Además, se podrían realizar pequeños ajustes sin rediseñar el proceso (fase de ajuste). En la fase de diagnóstico se analiza el proceso ejecutado y la salida de esta fase podría gatillar una nueva fase de rediseño del proceso. La minería de procesos es una herramienta valiosa para la mayoría de las fases mostradas en la Figura 3.3. Obviamente, la fase de diagnóstico puede beneficiarse de la minería de procesos. Sin embargo, la minería de procesos no está limitada a la fase de diagnóstico. Por ejemplo, en la fase de ejecución, las técnicas de minería de procesos se pueden usar para el soporte operacional. Se pueden utilizar predicciones y recomendaciones basadas en modelos aprendidos usando información histórica para influenciar los casos en ejecución. Se pueden utilizar formas similares de apoyo a la toma de decisiones para ajustar los procesos y guiar la (re)configuración de procesos. [26]

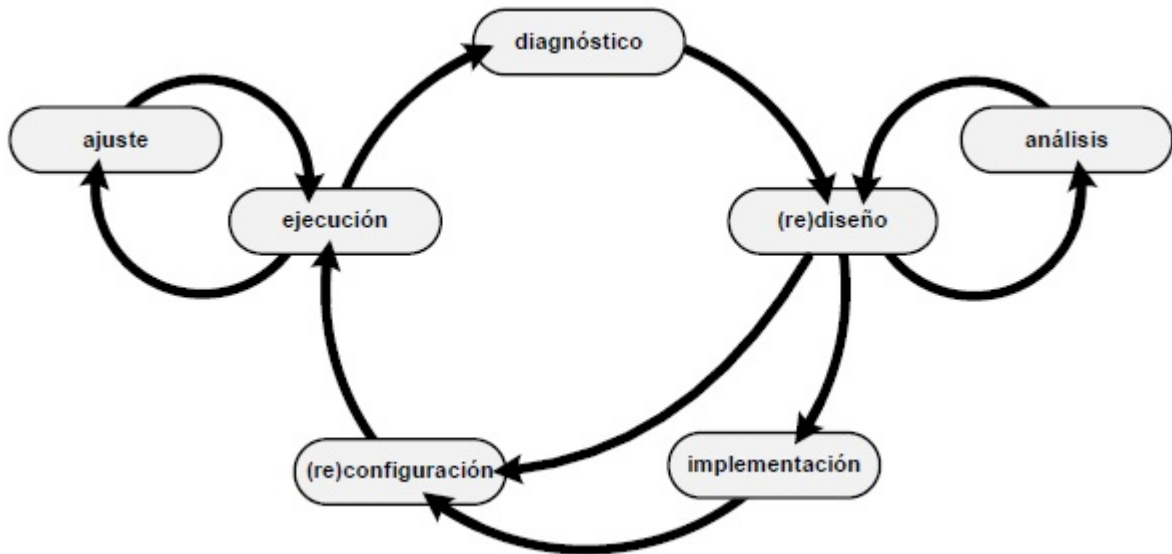


Figura 3.3. El ciclo de vida de BPM identificando las diferentes fases de un proceso de negocio y sus correspondientes sistemas de información; la minera de procesos cumple potencialmente un rol en todas las fases (excepto la fase de implementación). [26]

Mientras la Figura 3.3 muestra el ciclo de vida BPM como un todo, la Figura 3.4 se enfoca en las actividades y artefactos concretos de minería de procesos. Ésta describe las etapas posibles en un proyecto de minería de procesos. Cualquier proyecto de minería de procesos comienza con una planificación y una justificación para esta planificación (Etapa 0). Después de iniciado el proyecto, se necesita extraer los datos de eventos, modelos, objetivos, y preguntas a partir de los sistemas, expertos del dominio, y la gestión (Etapa 1). Esto requiere un entendimiento de los datos disponibles (“¿Qué puede ser usado para el análisis?”) y un entendimiento del dominio (“¿Cuáles son las preguntas importantes?”) y tiene como resultado los artefactos mostrados en la Fig. 5 (i.e., datos históricos, modelos hechos a mano, objetivos, y preguntas). En la Etapa 2, se construye el modelo de control de flujo y se le relaciona con el registro de eventos. Aquí, se pueden utilizar técnicas automáticas de descubrimiento de procesos. El modelo de procesos descubierto ya podrá proveer respuestas a algunas de las preguntas y gatillar acciones de rediseño o ajuste. Además, se podrá filtrar o adaptar el registro de eventos usando el modelo (e.g., eliminando actividades poco frecuentes o casos atípicos, e insertando eventos faltantes). Algunas veces se necesitan significativos esfuerzos para correlacionar eventos que pertenecen a la misma instancia de un proceso. Los eventos restantes están relacionados con entidades del modelo de proceso. Cuando el proceso es relativamente estructurado, el modelo de control de flujo podrá ser extendido con otras perspectivas (e.g., datos, tiempo, y recursos) durante la Etapa 3. La relación entre el registro de eventos y el modelo establecido en la Etapa 2 se utiliza para extender el modelo (e.g., se utilizan las marcas de tiempo de los eventos asociados para estimar los tiempos de espera para las actividades). Esto podrá utilizarse para responder preguntas adicionales y podría gatillar acciones adicionales. En última instancia, los modelos construidos en la Etapa 3 podrá ser utilizado para apoyar las operaciones (Etapa 4). El conocimiento extraído de los datos de eventos históricos se combina con la información acerca de los casos en ejecución. Esto podría utilizarse para intervenir, predecir, y recomendar. Las Etapas 3 y 4 solo se pueden alcanzar si el proceso es suficientemente estable y estructurado. [26]

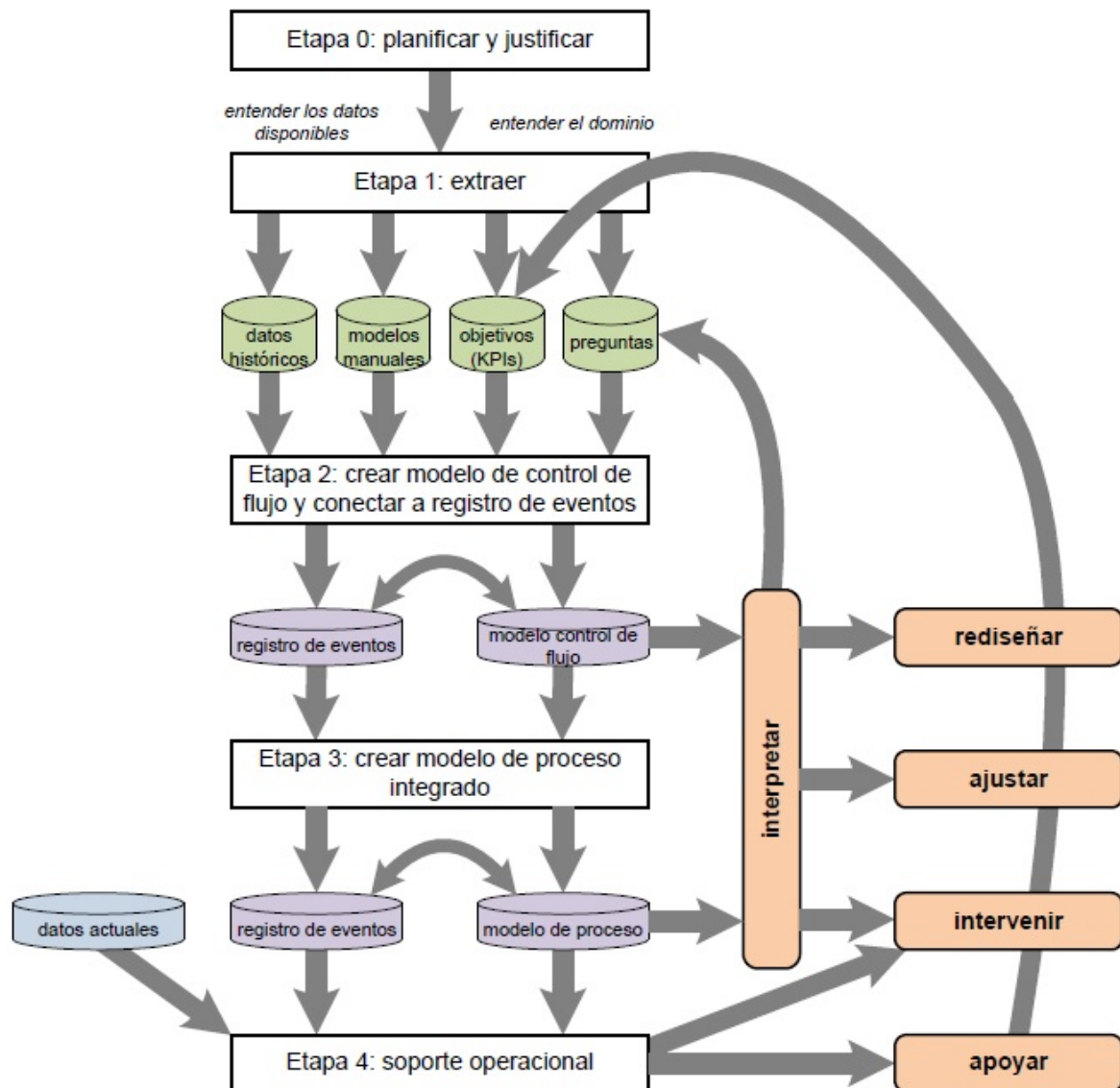


Figura 3.4. El modelo de ciclo de vida L describe un proyecto de minería de procesos consistente de cinco etapas: planificar y justificar (Etapa 0), extraer (Etapa 1), crear un modelo de control de flujo y conectarlo con el registro de eventos (Etapa 2), crear un modelo de proceso integrado (Etapa 3), y proveer soporte operacional (Etapa 4). [26]

3.4 - Relación modelo-realidad

Uno de los elementos principales de Process Mining es el énfasis en establecer una relación fuerte entre el modelo de proceso y la "realidad" capturada en la forma de un log de eventos. Utilizamos los términos Play-in, Play-out, y Replay para reflejar esta relación. La Figura 3.5 expresa estas tres relaciones:

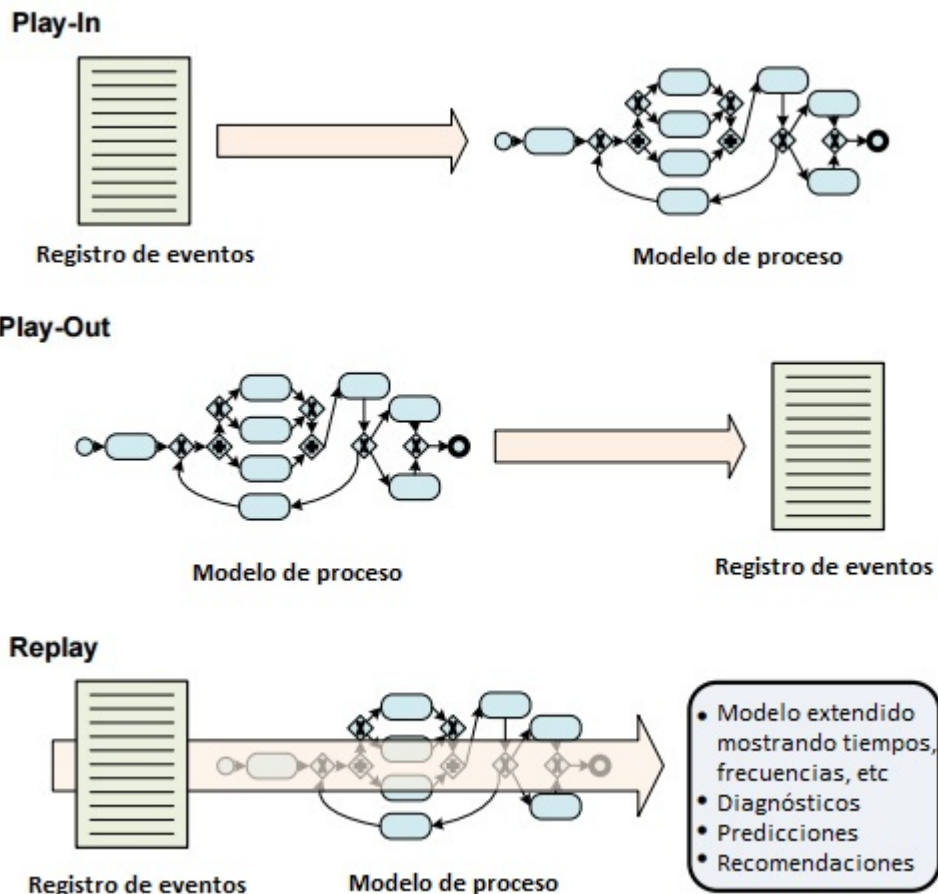


Figura 3.5. Tres formas de relacionar log de eventos (u otras fuentes de información que contengan comportamiento de ejemplo) con modelos de procesos: *Play-in*, *Play-out*, *Replay*. [8]

- Play-in dado un log de eventos se puede deducir un comportamiento y generar un modelo. Play-in también se refiere como deducción. La mayoría de las técnicas de data mining utilizan Play-in un modelo se diseña en base a los ejemplos (al log de eventos).
- Play-out se refiere al uso clásico de modelos de procesos. Dado un modelo se puede generar comportamiento que se almacena en un log de eventos. Play-out se puede usar para aprobar procesos operacionales utilizando algún modelo ejecutable. Las herramientas de simulación también utilizan un motor Play-out para conducir experimentos. La idea principal de la simulación es correr repetidamente un modelo y recorrer las estadísticas y los intervalos confidenciales. Un motor de simulación es similar a un motor de workflow. La diferencia principal es que el motor de simulación interactúa con un ambiente modelado mientras que el motor de workflow interactúa con un ambiente real.
- Replay utiliza tanto un log de eventos como un modelo como entrada. Un log de eventos puede volverse a correr por diferentes propósitos:
 - Chequeo de concordancia: las diferencias entre el modelo y el log se pueden detectar volviendo a correr el log sobre el modelo.
 - Extender el modelo con frecuencias e información temporal: Cuando se vuelve a correr el log se puede ver que partes del modelo se visitan con frecuencia. También se puede utilizar para detectar cuellos de botella.

- Construir modelos predictivos: para diferentes estados del modelo se pueden realizar predicciones particulares.
- Soporte operacional: Replay no está limitado a datos de eventos históricos. Se pueden volver a correr trozos parciales de casos que se encuentran corriendo. Esto se puede usar para detectar desviaciones en momento de ejecución.[4][5]

3.5 - Business Intelligence y Process Mining

Si bien, Process Mining es una herramienta poderosa dentro del amplio contexto de BPM, se suele considerar bajo el campo de Business Intelligence, sin embargo, existen diferencias entre estas dos tecnologías. Las detallaremos a continuación.

Business Intelligence es un concepto amplio que incluye todo lo que tenga como objetivo proveer información accionable que se utilice para la toma de decisiones. Una de sus definiciones lo enmarca como un conjunto de modelos matemáticos y metodologías de análisis que aprovechan los datos disponibles para generar información y conocimiento útil para procesos de toma de decisiones complejos. Process Mining se puede ver como una colección de técnicas de BI. Sin embargo, la mayoría de las herramientas de BI no son realmente inteligentes y no proveen ninguna de las capacidades de Process Mining, se focalizan en consultar y generar reportes, combinados con técnicas de visualización que muestran dashboards (gráficos).

Podemos destacar productos BI tales como: IBM Cognos Business Intelligence (IBM), Oracle Business Intelligence (Oracle), MS SQL Server Business Intelligence (Microsoft), MicroStrategy (MicroStrategy), SAS Enterprise Business Intelligence (SAS), Jaspersoft (Jaspersoft) y Pentaho BI Suite (Pentaho) entre otros.

Estos productos suelen ofrecer típicamente las siguientes funcionalidades: la extracción de datos de varias fuentes mediante procesos ETL (Extract, Transform, and Load). Los datos extraídos se transforman a un formato de datos estándar (por lo general una tabla multidimensional) y cargados en un sistema de BI. La posibilidad de realizar consultas Ad-hoc por parte de los usuarios para explorar los datos de manera personalizada. La definición de reportes estándares y dashboards interactivos que permiten la definición de tableros de control que consisten de datos tabulares y una variedad de gráficos (que el usuario puede cambiar, refinar, agregar y filtrar la vista actual utilizando controles predefinidos). Finalmente brindan la posibilidad de definir eventos y condiciones que puedan disparar una alerta por ejemplo, cuando las compras caen a un determinado índice un e-mail se manda al gerente de ventas.

Los enfoques administrativos, como Continuous Process Improvement (CPI), Total Quality Management (TQM) y Six Sigma también están relacionados con Process Mining. Estos enfoques tienen en común que los procesos son minuciosamente analizados para identificar si se pueden incorporar posibles mejoras. Claramente Process Mining contribuye en el análisis de desviaciones e ineficiencias. [5]

3.6 - Modelado y análisis de proceso

Podemos diferenciar dos grandes pilares dentro de Process Mining. Estos son el modelado y análisis de procesos por un lado, y por otra parte la minería de datos.

La cantidad de notaciones de modelado de procesos existentes hoy en día nos demuestra la relevancia de los modelos de proceso. Algunas organizaciones pueden usar solo modelos de procesos informales para estructurar las discusiones y para documentar los procedimientos. Sin embargo, las organizaciones que operan en un nivel más alto de BPM usan modelos que se pueden analizar y usar para aprobar modelos operacionales. [5]

Hoy en día, las innovaciones en computación y en comunicación son los mayores productores de cambios en los procesos de negocio. Los procesos de negocio se han vuelto más complejos. Los modelos de procesos asisten en la administración de la complejidad proveyendo puntos de vista y documentando los procedimientos. Los sistemas de información se deben configurar y manejar por instrucciones precisas. Los procesos que atraviesan la organización pueden funcionar adecuadamente si se hace un acuerdo común en las interacciones requeridas. Como resultado, los modelos de procesos se utilizan ampliamente en las organizaciones. Los modelos se usan para razonar sobre los procesos (rediseño) y para tomar decisiones dentro del proceso (planificar y controlar). Un modelo de proceso expresado en BPMN se puede utilizar para discutir responsabilidades, analizar el cumplimiento, predecir la performance utilizando simulación y configurar un sistema Work Force Management, cuyo objetivo es optimizar la productividad.

Al no ser una ciencia cierta, durante el modelado de procesos uno puede detectar cierto tipo de errores como, por ejemplo, que dicho modelo puede estar describiendo una versión idealizada de la realidad. Otro tipo de error puede ser el de la incapacidad de capturar adecuadamente el comportamiento humano. No se puede modelar el proceso aislado sino que se tiene que tener en cuenta las distintas intervenciones que tiene el usuario en distintas tareas del proceso, o en otros procesos. El modelo se puede encontrar en un nivel de abstracción equivocado. Finalmente se debe elegir un nivel de abstracción adecuado dependiendo de los datos de entrada y las preguntas que se requieren responder. Puede ser que el modelo sea muy abstracto y q sea incapaz de responder preguntas muy concretas, o puede ser muy detallado y así costar la comprensión

Estos son algunos de los problemas que una organización enfrenta cuando se hacen modelos a mano. Solo diseñadores y analistas con experiencia pueden hacer modelos que tengan un valor predictivo bueno y se pueden usar como punto de entrada para (re) implementación o rediseño. Un modelo inadecuado puede llevar a conclusiones incorrectas. Por lo tanto se avoca al uso de datos de eventos. Process Mining permite la extracción de modelos basados en hechos. Además, Process Mining no tiene como objetivo crear un solo modelo del proceso. En cambio provee varias vistas de la misma realidad en diferentes niveles de abstracción. Por ejemplo se puede inspeccionar el comportamiento más frecuente. Process Mining puede también revelar que las personas en una organización no funcionan como máquinas. Por un lado puede mostrar que se producen muchos tipos de ineficiencias, y por otro puede mostrar

la flexibilidad remarcable de algunos empleados para enfrentarse con problemas y distintos trabajos. [5]

Modelos de procesos

Modelar procesos no es una tarea fácil, pero sin embargo es una tarea muy importante y necesaria. Process Mining puede facilitar la construcción de modelos mejores en menos tiempo. Los algoritmos de descubrimiento de procesos como el algoritmo α pueden generar automáticamente un modelo de proceso. Si bien ciertas técnicas de Process Mining dan como resultado un modelo de proceso específico, se pueden convertir fácilmente un modelo en de una notación a otra. Por ejemplo el algoritmo α produce una red de Petri como resultado y se puede convertir fácilmente en un modelo BPMN, BPEL, o un diagrama de actividades UML.

Process Mining utiliza modelos de procesos en distintas notaciones, las cuales son: sistemas de transiciones, redes de Petri, redes de trabajo social. YAWL (Yet Another Workflow Language), BPMN (Business Process Management Notation), redes casuales entre otros.

Para el modelado de procesos nos concentramos en la perspectiva de control de flujo (mencionada en la sección Process Mining y definiciones). Asumimos que existe un conjunto de nombres de actividades A. El objetivo de un modelo de proceso es decidir cuáles actividades deben ejecutarse y en qué orden. Las actividades pueden ejecutarse secuencialmente, algunas pueden ser opcionales, otras concurrentes y es posible la repetición de la misma actividad. [5]

Análisis de proceso basado en modelo

Los principales enfoques para análisis basado en modelos son: verificación y análisis de performance. La verificación se interesa en la correctitud de un sistema o proceso. El análisis de performance se focaliza en los tiempos de flujos, los tiempos de espera, la utilización y niveles de servicios. Verificación: Cuando una red de trabajo tiene caminos permitidos erróneos, se dice que el modelo tiene ruido, una tarea de verificación puede ser detectar el ruido en el modelo. Otra tarea de verificación puede ser la comparación entre dos modelos, por ejemplo se puede comparar la implementación de las necesidades de un proceso con una especificación de más alto nivel del mismo proceso. Análisis de performance: por lo general se identifican 3 dimensiones de performance: tiempo, costo y calidad. Para cada uno de estas dimensiones de performance se pueden definir distintos Key Performance Indicator (KPI). Cuando miramos a la dimensión tiempo se pueden definir los siguientes indicadores de performance:

- El tiempo de flujo: es el tiempo total desde la creación del caso hasta la finalización del mismo. Se pueden medir el promedio del tiempo de flujo de todos los casos. Sin embargo el nivel de varianza puede ser importante y debe medirse también.
- El tiempo de servicio es el tiempo que realmente requiere un caso. Por lo general el tiempo de servicio es una fracción del tiempo de flujo (o tiempo legado).
- El tiempo de espera, es el tiempo que un caso espera para que un recurso esté disponible. Este tiempo se puede calcular para cada actividad o para un caso entero.

- El tiempo de sincronización es el tiempo que una actividad no está completamente habilitada y espera un disparador externo o algún acontecimiento paralelo. El caso espera por sincronización en vez de por un recurso.

Los indicadores de performance se definen también para la dimensión de costo. El costo de ejecutar una actividad puede depender del tipo de recurso utilizado, la utilización, o la duración de una actividad. El costo de los recursos puede depender de la utilización de los mismos. Un índice indicador de performance en la mayoría de los procesos es el la utilización promedio de los recursos sobre un período de tiempo.

La dimensión de calidad por lo general se focaliza en el producto o el servicio que se provee al cliente. Como los costos se puede medir de diferentes formas. Se puede medir por ejemplo el número promedio de quejas por caso o el número de defectos por producto.

Mientras que la verificación se focaliza en la correctitud lógica del proceso modelado, el objetivo del análisis de performance es mejorar los procesos con respecto al tiempo, costo y calidad.

Por lo general los modelos analíticos requieren muchas suposiciones y pueden solo utilizarse para cuestiones particulares. Por lo tanto a veces se debe acudir a la simulación. La mayoría de las herramientas de BPM proveen simulación.

Muy pocas herramientas utilizan la simulación de manera efectiva, esto puede suceder por falta de entrenamiento y limitaciones de las herramientas existentes. Pero también hay problemas fundamentales en la simulación, los modelos de simulación tienden a sobre simplificar las cosas, por ejemplo las personas no trabajan a velocidades constantes, y muchas veces requieren de distribuir su atención entre muchos procesos. Esto puede traer efectos dramáticos a la performance de un proceso y por eso esos aspectos se deben tener en cuenta.

Los modelos deben coincidir con la realidad, porque de no ser así se estarían analizando y realizando simulaciones con modelos que son “idealizados” y hay poca alineación entre los modelos realizados y la realidad. El objetivo de Process Mining es solucionar estos problemas estableciendo una conexión directa entre los modelos y los datos de eventos de bajo nivel actuales del proceso. Las distintas técnicas de descubrimiento, permiten ver la misma realidad en diferentes ángulos y en diferentes niveles de abstracción. [5]

3.7 - Minería de datos

Las ideas originarias del campo de data mining son utilizadas para la evaluación de resultados de Process Mining. Por ejemplo uno puede adoptar varias aproximaciones de data mining para medir la calidad de modelos de procesos mejorados o descubiertos. Un entendimiento básico de data mining puede ayudar al entendimiento completo de las técnicas de Process Mining.

Data mining se define como “el análisis de (por lo general grandes) conjuntos de datos para encontrar relaciones no sospechadas y resumir los datos a maneras nobles que sean comprensibles y útiles para los dueños de los datos”. La entrada es típicamente una tabla y la salida puede ser las reglas, los clusters, estructuras de árboles, gráficos, ecuaciones, patrones, etc. El crecimiento del universo digital es el principal conductor de la popularidad de data mining.

Las tablas están compuestas por filas que representan instancias, individuos, entidades, casos objetos o registros. Las columnas son llamadas variables. Las variables son referidas por lo general como atributos, características, o elementos de datos. Se hace una distinción entre variables categóricas y variables numéricas. Las variables categóricas tienen un conjunto limitado de valores posibles, y pueden enumerarse fácilmente por ejemplo una variable booleana que puede ser true o false. Las variables numéricas tienen un orden y no se pueden enumerar fácilmente. Por ejemplo temperatura, edad, peso, número de tiempos y altitud. Las variables categóricas se dividen en ordinales y nominales. Las variables nominales no tienen orden lógico. Por ejemplo los booleanos los colores, países, no tienen un acuerdo común para un orden lógico. Las variables ordinales tienen un orden asociado. Por ejemplo la calificación (excelente, bueno, malo). Por lo general, antes de aplicar cualquier técnica de data mining los datos son pre procesados. Es decir las columnas y las filas pueden ser removidas por varias razones. Las instancias que están corruptas deben ser removidas también. Además el valor de una variable para una instancia particular puede estar perdido o tener el valor equivocado. Las técnicas de data mining no realizan tantos supuestos sobre el formato de la entrada como las técnicas de Process Mining. En Process Mining los eventos se ordenan de acuerdo al tiempo, mientras que en data mining las filas no tienen ningún orden. Para casos particulares es posible convertir un log de eventos en un conjunto de datos simples para data mining. Nos referiremos a esto como extracción de características. Luego utilizaremos la extracción de características para varios propósitos como ser análisis de decisiones, en un modelo de proceso descubierto y clusterización de casos antes del descubrimiento de proceso para que de esa forma cada clúster tenga un modelo de proceso dedicado. Luego de estudiar la estructura básica de entrada para data mining, clasificamos las técnicas de data mining en dos categorías principales: aprendizaje supervisado, y aprendizaje no supervisado. [4][5]

Aprendizaje supervisado: clasificación y regresión

El aprendizaje supervisado asume datos etiquetados, es decir, hay una variable respuesta que etiqueta cada instancia. Las otras variables son variables predictivas y estamos interesados en explicar las variables respuesta en términos de las variables predictivas. A veces las variables de respuesta son llamadas variables dependientes y las variables predictivas se llaman variables independientes. Por ejemplo, queremos predecir el resultado final de un estudiante en términos de las notas de los estudiantes del curso. Las técnicas de aprendizaje supervisado se pueden dividir en clasificación y regresión dependiendo del tipo de variable respuesta (categórica o numérica) Las técnicas de clasificación asumen una respuesta categórica y el objetivo es clasificar instancias basadas en las variables predictivas. Las técnicas de regresión asumen como respuesta una variable numérica. El objetivo es encontrar una función que encaje los datos con menos errores. La técnica de regresión más frecuente es la regresión lineal. Dada una variable de respuesta y y las variables predictivas x_1, x_2, \dots, x_n

un modelo lineal $y = f(x_1, x_2, \dots, x_n) = a_0 + \sum_{i=1}^n a_i x_i$ se aprende sobre el conjunto de datos. Por cada instancia en el conjunto de datos hay un error de $|y - \hat{y}|$. Un enfoque es minimizar la suma de los errores al cuadrado, es decir dadas m instancias el objetivo es encontrar una función f para que $\sum_{j=1}^m (y_j - \hat{y}_j)^2$ sea mínimo. La clasificación requiere una variable de respuesta categórica. En algunos casos tiene sentido transformar una variable de respuesta numérica en una categórica. Por ejemplo transformar la variable numérica edad en una variable categórica que las divida en infante, joven, adulto, etc. [5]

Aprendizaje no supervisado: Clustering y descubrimiento de patrones

El aprendizaje no supervisado asume datos no etiquetados, es decir, las variables no se separan en variables de respuesta o de predicción. Analizaremos dos tipos de aprendizaje no supervisado: clustering y descubrimiento de patrones. Los algoritmos de clustering examinan los datos para encontrar grupos de instancias que son similares. A diferencia de la clasificación, no se focaliza en algunas variables de respuesta, sino que en las instancias como un todo. Por ejemplo el objetivo puede ser encontrar grupos homogéneos de estudiantes, o clientes. Algunas técnicas bien conocidas de clustering son k-means clustering y clustering jerárquico aglomerativo. Hay muchas técnicas para descubrir patrones en los datos. Por lo general el objetivo es encontrar reglas de la forma: IF X THEN Y donde X e Y relacionan valores de variables diferentes. La técnica más conocida para estas cuestiones es minería de reglas de asociación.

Los árboles de decisión también se pueden convertir en reglas. Sin embargo, un árbol de decisión se construye para una variable de respuesta particular. Por lo tanto las reglas que se extraen de un árbol de decisión solo dicen algo sobre las variables de respuesta en términos de algunas de las variables predictivas. Las reglas de asociación se descubren utilizando aprendizaje no supervisado, esto es no se requiere seleccionar una variable de respuesta.

Los resultados de data mining pueden ser o descriptivos o predictivos. Los árboles de decisiones, reglas de asociación, y funciones de regresión dicen algo del conjunto de datos usado para aprender el modelo. Sin embargo también se pueden utilizar para hacer predicciones para nuevas instancias es decir, predecir la performance total de los estudiantes basándose en las calificaciones del curso obtenidas en el primer semestre. [5]

3.8 - Registros de eventos

Como se mencionó previamente el objetivo de Process Mining es descubrir, monitorear, y mejorar los procesos reales, mediante la extracción de conocimiento del log de eventos que ya se encuentra disponible en los sistemas de hoy. La estructura del log de eventos puede variar de acuerdo a la técnica de data mining que se utilizará. El desafío es extraer estos datos de diferentes variedades de fuentes de datos, como ser bases de datos, sistemas de archivos planos, logs de mensajes y sistemas de administración documentados. Cuando se mezcla y se extraen los datos la sintaxis y la semántica juegan un rol fundamental. Dependiendo de las preguntas que se buscan contestar se requieren distintas vistas de los datos disponibles.

La idea de Process Mining es analizar datos de eventos desde una perspectiva orientada a procesos. El objetivo es responder a preguntas de procesos operacionales. Como por ejemplo ¿Cómo controlar mejor el proceso? En el proceso general de Process Mining mostrado en la Figura 3.6 se realiza un especial énfasis en lo rol de los datos de eventos. El punto de comienzo son los datos crudos escondidos en todo tipo de fuentes de datos. Una fuente de dato puede ser un archivo plano, una planilla de Excel, un log transaccional, o una tabla de una base de datos. Obviamente todos los datos se encuentran en diferentes formatos y formas de estructuración. Se requiere un poco de esfuerzo recolectar de todas estas fuentes la información relevante en un formato adecuado.

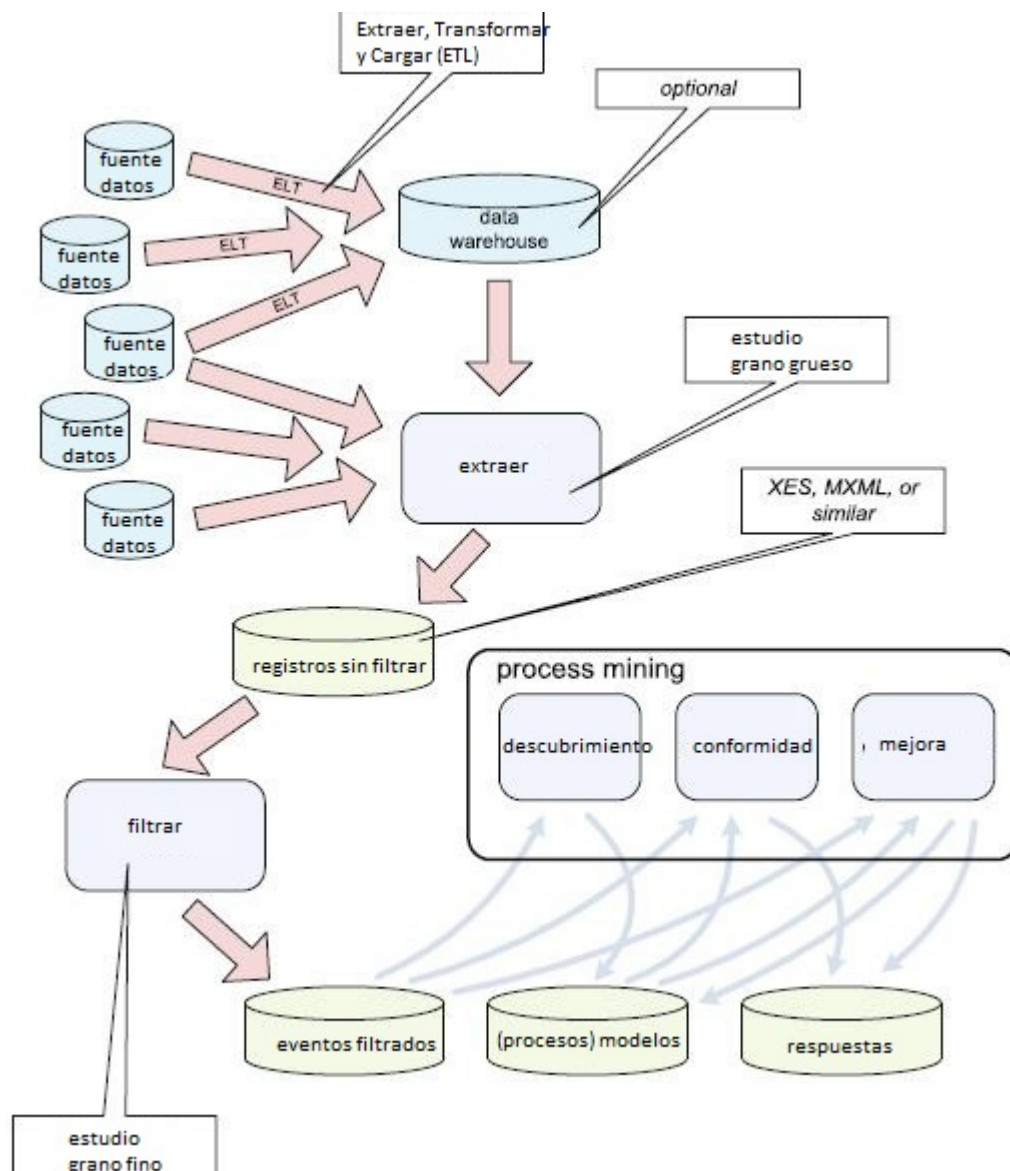


Figura 3.6 Visión global describiendo el flujo de trabajo desde fuentes de datos heterogéneas hasta los resultados de Process Mining [5]

Tanto en el contexto de BI como en el de data mining existe la fase de ETL “extract, Transform, and Load” que se utiliza para describir el proceso que involucra:

- Extracción de datos de una fuente externa

- Transformación de los datos para adecuarse a necesidades operacionales (tratando con temas de sintáctica y semántica mientras se asegura los niveles de calidad predefinidos)
- Carga en un sistema como un data warehouse o una base de datos relacional.

Un data warehouse es un repositorio lógico único de los datos transaccionales y operacionales de la organización. El data warehouse no produce datos simplemente los almacena de sistemas operacionales. El objetivo es unificar información como esa para que pueda ser usada para reporte, análisis, etc. Muchas organizaciones proveen su data warehouse pero el mismo no es orientado a procesos, por lo general el data warehouse se utiliza para Online Analytical Processing (OLAP) que no poseen mucha información relacionada al proceso, y Process Mining requiere de eventos relevantes de procesos y el orden de los mismos. Haya o no un data warehouse se deben extraer los datos y deben convertirse en un log de eventos. Aquí el alcance es el más importante. Por lo general el problema no es de conversiones sintácticas sino la selección de los datos adecuados. Los formatos típicos para almacenar el log de eventos son XES (eXtensible Event Stream) and MXML (Mining eXtensible Markup Language). Un log de eventos corresponde a un solo proceso. Dependiendo de las preguntas y los puntos de vista elegidos, se pueden extraer distintos logs de eventos del mismo conjunto de datos. Una vez que se crea el log de eventos por lo general se filtra. El filtrado es un proceso iterativo. El alcance de grano grueso se realiza cuando se extraen los datos en un log de eventos. El filtrado corresponde al alcance de grano fino que se basa en los resultados de análisis inicial. Basándose en el log filtrado se pueden aplicar los distintos tipos de técnicas de Process Mining: descubrimiento, concordancia, y mejora. [4] [5].

Hasta hace poco el estándar por defecto para almacenar e intercambiar logs de eventos era el MXML (mining eXtensible Markup language). MXML emergió en el 2003 y luego se adoptó por ProM. Utilizando MXML se puede almacenar un log de eventos utilizando una sintaxis basada en XML. ProMimport es una herramienta que soporta la conversión de diferentes fuentes de datos a MXML, como por ejemplo MS Access, Aris PPM, CSV, Apache, PeopleSoft, Subversion, SAP R/3. Protos, Cognos. MXML tienen una notación estándar para almacenar timestamps, recursos y tipos de transacciones. Además se pueden agregar arbitrariamente atributos a eventos y casos. El resultado posterior es un MXML ad-hoc con ciertos atributos de datos que se interpretan de una forma específica. XES es el sucesor de MXML. Se basa en muchas experiencias prácticas que se tuvieron con MXML, el formato XES se hizo menos restrictivo y realmente extensible. Es el formato adoptado por la IEEE Task Force on Process Mining. El formato es adoptado por herramientas como ProM, Nitro, XESme, y OpenXES. [5]

3.9 - Descubrimiento de procesos

Una vez que se obtiene el log de eventos desde los distintos sistemas de información nos encontramos en condiciones de aplicar las distintas técnicas de Process Mining: descubrimiento de procesos, chequeo de concordancia y extensión.

Las técnicas de descubrimiento de procesos son técnicas de tipo Play-in que se encargan de la perspectiva de control de flujo de proceso, es decir toma como entrada un log de eventos, y construye un modelo de proceso, capturando el comportamiento que se ve en el log.

A continuación la definición del algoritmo de descubrimiento de procesos: sea L el log de eventos, especificado por el estándar XES. Un algoritmo de descubrimiento de proceso es una función que mapea L en un modelo de proceso, por esto es que el modelo es representativo del comportamiento que se ve en el log de eventos. El desafío es encontrar tal algoritmo. No se especifica qué tipo de modelo se generará si BPMN, YAWL o un modelo de red de Petri.

La definición anterior no especifica el formato de destino y se utiliza un log de eventos como entrada, sin especificar los requerimientos tangibles. Para hacer los requerimientos más concretos definimos el destino como una red de Petri. Además utilizaremos un log de eventos simple L , que es un conjunto múltiple de trazes sobre un conjunto de actividades A , es decir, $L \in B(A^*)$. Por ejemplo: $L1 = [(a, b, c, d)3, (a, c, b, d)2, (a, e, d)]$.

El objetivo es descubrir una red de Petri que pueda reproducir el log de eventos. Definición (problema de descubrimiento de procesos específico) un algoritmo de descubrimiento de procesos es una función γ que mapea un log $L \in B(A^*)$ en una red de Petri $\gamma(L) = (N, M)$. Idealmente, N es una red de workflow y todos los trazes de L corresponden a una posible secuencia de disparo de (N, M) .

La función γ define una técnica Play-in. Basándonos en el $L1$, un algoritmo de descubrimiento de proceso puede descubrir la red de workflow de la figura 3.7.

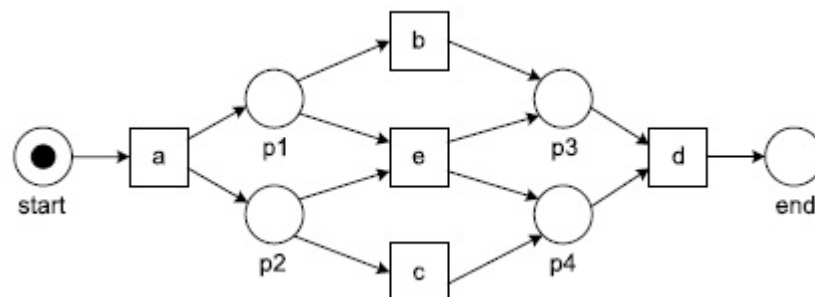


Figura 3.7. Red-WF N1 descubierta del $L1 = [(a, b, c, d)3, (a, c, b, d)2, (a, e, d)]$

Se puede ver como la red de workflow puede reproducir todas las instancias del log $L1$.

La mayoría de las notaciones de modelado asumen el criterio de correctitud y para todos los fenómenos como livelocks y deadlocks no son deseados, independientemente de la notación. El formalismo compacto de la semántica formal de las redes de Petri puede ser el más indicado para explicar los algoritmos de descubrimiento de procesos. Pero hay muchas técnicas para transformar redes de Petris en otras notaciones como ser BPMN, YAWL, etc. El modelo descubierto debe ser representativo del comportamiento que se observa en el log de eventos. Esto se realiza bajo el requerimiento de que el modelo sea capaz de reproducir todo el comportamiento en el log, es decir cada rastro en el log de eventos es un posible disparo

de secuencia del el WF-net (work-Flow net). Esto es lo que se llama requerimiento de adecuación. En general, hay un equilibrio entre los siguientes cuatro criterios de calidad:

- Fitness: el modelo descubierto debería permitir el comportamiento observado en el log de eventos.
- Precisión: El modelo descubierto no debería permitir comportamiento que no esté relacionado con lo que se vea en el log de eventos.
- Generalización: el modelo descubierto debe generalizar el comportamiento ejemplo que se observa en el log.
- Simplicidad: El modelo descubierto debe ser simple.

Un modelo que tenga buen fitness es capaz de reproducir la mayoría de los rastros en un log. La precisión está relacionada con la noción de underfitting presentada en el contexto de data Mining. Un modelo con precisión pobre tiene underfitting, es decir, permite un comportamiento que es muy diferente al que se observa en el log de eventos. La generalización está relacionada con la noción de overfitting. Un modelo con overfitting no generaliza lo suficiente, es decir, es muy específico y se deja llevar por los ejemplos en el log de eventos. El cuarto criterio está relacionado con la navaja de Occams que citaba que “ no se debe incrementar, más de lo necesario, el número de entidades requeridas para explicar algo”. Siguiendo este principio buscamos al modelo de proceso más simple.

Es un desafío balancear los cuatro criterios. Por ejemplo, un modelo muy simplificado tiende a tener menos fitness o falta de precisión. Además, hay una compensación obvia entre underfitting y overfitting. [5].

3.10 - Chequeo de concordancia

Las técnicas de chequeo de concordancia son técnicas del tipo Replay que utiliza tanto el log de eventos como el modelo de proceso como entrada, es decir la historia se vuelve a reproducir utilizando el modelo para analizar distintos fenómenos. Replay se puede utilizar para descubrir cuellos de botellas y análisis de decisiones. Para el chequeo de concordancia ya se posee el modelo de proceso y el log de eventos. El modelo puede haberse construido a mano o puede haber sido descubierto. El chequeo de concordancia relaciona eventos en el log de eventos con actividades en el modelo de proceso y las compara. El objetivo es encontrar las cosas en común y las discrepancias entre el comportamiento modelado y el comportamiento observado detectando desviaciones indeseables que provocan ineficiencias. . Además las técnicas de chequeo de concordancia pueden utilizarse también para medir la performance de los algoritmos de descubrimiento de procesos y para reparar modelos que no estén alineados con la realidad. La idea principal de chequeo de concordancia se ilustra en la figura 3.8.

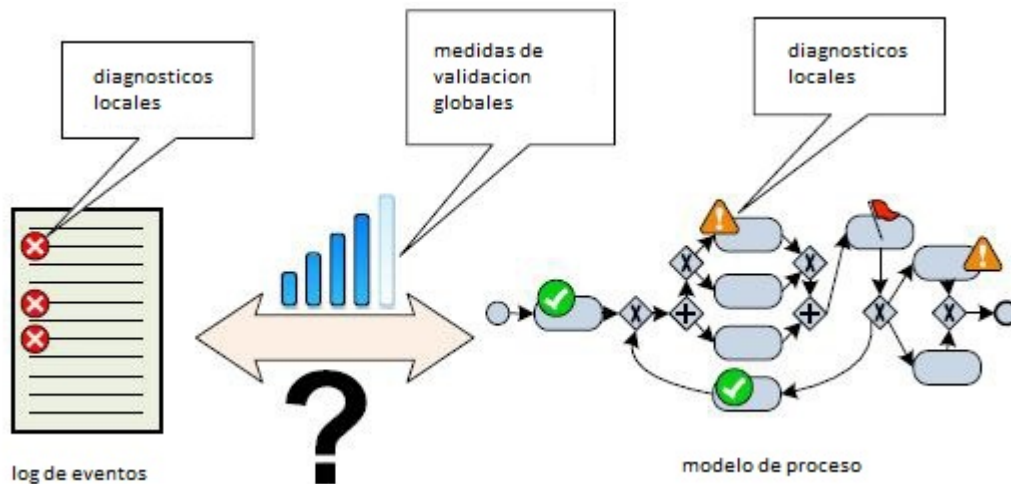


Figura 3.8. Chequeo de concordancia: Las medidas globales de chequeo de concordancia cuantifican la concordancia total del modelo y el log. Se realizan diagnósticos locales sobresaltando los nodos del modelo donde el modelo y el log no concuerdan. [5]

El comportamiento de un modelo de proceso y el comportamiento almacenado en un log de eventos se comparan para encontrar cosas en común y discrepancias. Ese análisis resulta en medidas de concordancia globales (por ejemplo 85% de los casos en el log de eventos se pueden reproducir con el modelo) y diagnóstico local (por ejemplo la actividad x fue ejecutada 15 veces a pesar de que no era permitido de acuerdo al modelo). Esta interpretación de no concordancia depende del propósito del modelo. Si el modelo intenta ser descriptivo, entonces las discrepancias entre el modelo y el log indican que el modelo necesita ser mejorado para adecuarse mejor a la realidad. Si el modelo es normativo, entonces las discrepancias encontradas puede exponer desviaciones indeseables, es decir, la necesidad de un mejor control de los procesos. Otras discrepancias pueden revelar desviaciones deseables. Por ejemplo, los trabajadores pueden derivarse a servir clientes mejor o a manejar circunstancias que no fueron previstas por el modelo de proceso. De hecho, la flexibilidad con la no concordancia por lo general correlaciona positivamente. Cuando se hace el chequeo de concordancia es importante ver las desviaciones desde dos ángulos:

- El modelo está mal y no refleja la realidad (Cómo mejorar un modelo?)
- Los casos se desvían con respecto al modelo y se requieren acciones correctivas (Cómo mejorar el control para hacer cumplir una mejor concordancia?)

Las técnicas de chequeo de concordancia deben contemplar estos dos puntos de vista.

El objetivo del alineamiento de negocios es asegurarse que la información de los sistemas y los procesos de negocio reales estén bien alineados. Las personas deben trabajar en conjunto con los sistemas de información antes que trabajar a su espalda para realizar las cosas. Process Mining puede mejorar el alineamiento entre los sistemas de información, procesos de negocios y la organización. Mediante el análisis de los procesos reales y las discrepancias diagnosticadas, nuevos puntos de vista se pueden reunir mostrando cómo mejorar el soporte por sistemas de información.

El término auditoría se refiere a la evaluación de las organizaciones y sus procesos. La auditoría se realiza para comprobar la validez y la confianza de la información sobre estas organizaciones y los procesos asociados. Este procedimiento se realiza para chequear si los procesos de negocio se ejecutan dentro de ciertos límites seteados por los administradores, gobernadores y otros stakeholders. Por ejemplo, las reglas específicas pueden ser establecidas por la ley o políticas de compañías y el auditor se debe encargar de chequear si estas reglas se cumplen o no. La violación de esas reglas puede indicar fraude, mala práctica, riesgos e ineficiencias.

Hoy en día con el nivel de detalle que se tiene de los procesos almacenados en un log de eventos ya no debería ser aceptable chequear solo un pequeño conjunto de ejemplos fuera de línea, en cambio todos los eventos en un proceso de negocio pueden ser evaluados y esto se puede hacer mientras el proceso siga corriendo. La capacidad de registrar datos y las técnicas avanzadas de Process Mining permite nuevas formas de auditoría, a través de Process Mining. [5]

3.11 - Mejora y extensión

Mientras el foco principal del descubrimiento de procesos es en la perspectiva de control de flujo, el log de eventos puede contener información valiosa relacionada con otras perspectivas como la perspectiva organizacional, la perspectiva de caso, y la perspectiva de tiempo. La perspectiva organizacional se puede usar para obtener puntos de vistas de patrones de trabajo típicos, estructuras organizacionales y redes de trabajo social. Los timestamps y la frecuencia de las actividades se pueden utilizar para detectar cuellos de botella y diagnosticar otros problemas relacionados con la performance. Los datos de los casos se pueden usar para tener mejor entendimiento en la toma de decisiones y analizar diferencias sobre los casos. Además, las diferentes perspectivas se pueden mezclar en un único modelo que provea una visión integrada del proceso. Ese modelo integrado se puede utilizar para análisis “what if” utilizando simulación.

En esta sección no focalizamos en la mejora para extender y mejorar un modelo de proceso existente utilizando información sobre el proceso actual almacenado en el log de eventos. Hay dos tipos de mejoras: reparación y extensión. Por medio de la extensión agregamos una nueva perspectiva al modelo de proceso mediante la correlación cruzada con el log. [5]

Relativo a la decisión

La perspectiva de casos se focaliza en las propiedades de las distintas instancias de proceso. Cada caso se caracteriza por sus atributos, los atributos de sus eventos, el camino que se tomó e información de la performance (ejemplo tiempos de flujos). Primero nos focalizamos en la influencia de atributos de casos y eventos en el ruteo de los casos. La minería de decisión tiene como objetivo encontrar reglas explicando tales decisiones en términos de las características del caso.

Relativo al tiempo

La perspectiva tiempo se concibe con la sincronización y la frecuencia de eventos. En la mayoría de los log de eventos hay un timestamp. La granularidad de estos timestamps puede variar. En algunos log solo se da información de la fecha, otros logs tienen información de los timestamps con precisión de milisegundos. La presencia de timestamps posibilita el descubrimiento de cuellos de botella, el análisis de niveles de servicios, el monitoreo de la utilización de los recursos y la predicción del tiempo que procesamiento de los casos corridos. Todos los diagnósticos que se pueden hacer sobre el tiempo y probabilidades son posibles sólo porque los eventos en el log se acoplan a elementos del modelo por medio de Replay. Luego de Replay para cada lugar una colección de visitas de tokens se registra. Cada visita de token tiene un tiempo de comienzo y de fin. Por lo tanto se puede derivar un conjunto múltiple de duraciones. Para un log de eventos grande ese conjunto múltiple contendrá cientos de elementos. Por lo que es posible ajustar una distribución y computar un estándar de estadísticas como, desviación estándar, máxima y mínima. Lo mismo sucede con instancias de actividad. Cada instancia de actividad tiene un tiempo de inicio y de fin. De esta forma se puede derivar un conjunto múltiple de tiempos de servicios. También de aquí se puede computar estadísticas estándar. Las estadísticas también se pueden computar para tiempos de espera. Es posible computar intervalos confidentes para derivar declaraciones como “el 90% de los intervalos confidentes para el tiempo medio de espera para la actividad x está entre 40 y 50 minutos”.

Si el log de eventos tiene eventos como asignado, suspendido, retomado, salteo manual, caso abortado, se pueden recolectar más estadísticas durante la reproducción. Por ejemplo si los eventos de comienzo se proceden de los eventos de asignación, es posible analizar cuanto se toma en comenzar a ejecutar una actividad después de haber sido asignada a un recurso específico. Si el log de eventos no tiene información transaccional las actividades se suponen atómicas, sin embargo, todavía se puede analizar el tiempo que pasa entre esas actividades atómicas. Además se puede aplicar heurística para “adivinar” la duración de las instancias de actividad. [5]

Relativo a los recursos

Las actividades, las entidades organizacionales y los recursos se pueden relacionar. Desde que los eventos en el log se refieren a actividades y recursos (e indirectamente a las entidades organizacionales), se pueden extraer del log mediciones de performance y se pueden proyectar en los modelos. Por ejemplo, se pueden proyectar frecuencias en las actividades, entidades organizacionales y recursos. Se puede ver cuántas veces un recurso ejecuto alguna actividad y de la misma manera saber cuántas veces fue usada una entidad organizacional. Si el log de eventos contiene alta calidad de información incluyendo timestamps y tipos transaccionales, se puede analizar en detalle el comportamiento de los recursos. [5]

Ya que hemos hecho un recorrido por BPM y Process Mining, resaltando el importante papel que cumple la aplicación de dichas técnicas en la evolución de los procesos de una organización, analizaremos el último concepto que es de interés para este trabajo que es Cloud Computing y su aplicación para BPM.

Capítulo 4 - Cloud Computing y BPaaS

Los avances tecnológicos en materia de capacidad de cómputo, velocidad de comunicaciones y disponibilidad para el acceso a través de Internet han hecho de Cloud Computing una tecnología más que al alcance tanto para el usuario promedio (ofreciendo servicios, almacenamiento, etc) como para las empresas ofreciendo servicios de computación bajo demanda con una alta fiabilidad, escalabilidad y disponibilidad en un entorno distribuido. No pasó mucho tiempo para que las tecnologías referentes a BPM tomen nota de esto y se comience con el estudio e implementación de BPMS en la “nube”. La gran capacidad computacional de los sistemas en el cloud y el “pago por uso” en lugar de enfrentar grandes inversiones en software y hardware, son dos grandes ventajas de la combinación de estas técnicas.

Esto conlleva a que se introduzcan conceptos como descomposición de procesos y un cambio en el concepto de monitoreo de procesos que se tenía hasta ahora.

En este capítulo analizaremos Cloud Computing y su relación con BPM viendo como se comporta BPM en ambientes distribuidos.

4.1 - Cloud Computing

La Computación en la Nube (Cloud Computing) está cambiando la forma en que las industrias y las organizaciones hacen sus negocios, en que los recursos dinámicamente escalables y virtualizados se proporcionan como un servicio a través de Internet. Este modelo crea una oportunidad nueva para las organizaciones que quieren brindar servicios bajo demanda con alta confiabilidad, escalabilidad y disponibilidad en ambientes distribuidos. [10]

El Instituto Nacional de Estándares y Tecnología (NIST) define a Cloud Computing en [11] como un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente aprovisionados y publicados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. Este modelo de nube promueve la disponibilidad y está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue (Figura 4.1).

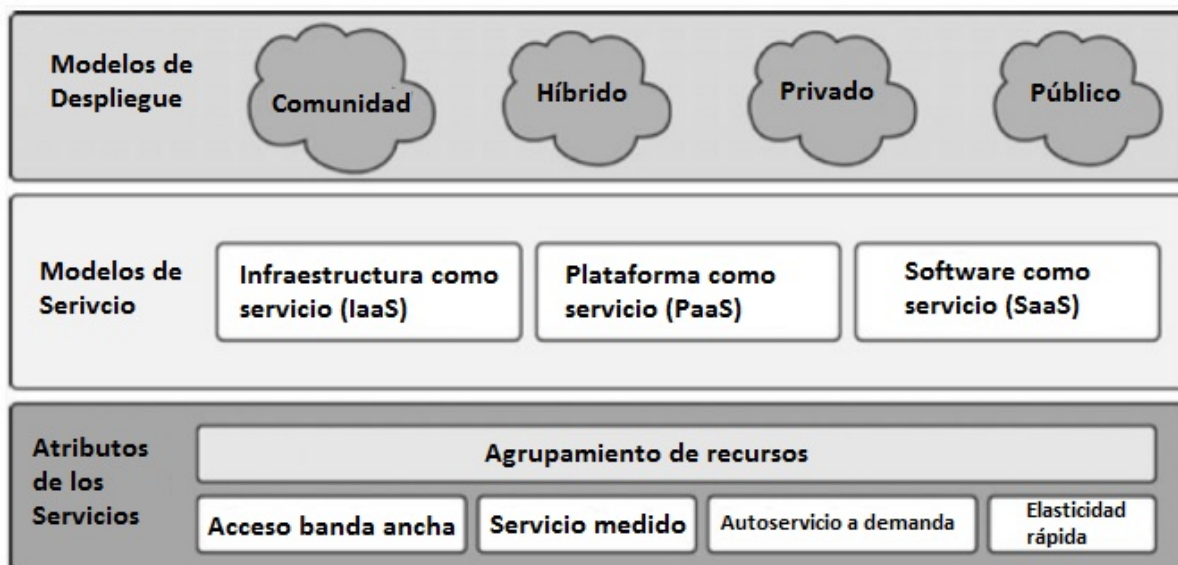


Figura 4.1. Modelo de Cloud Computing por el NIST [12]

4.2 - Características esenciales

Dentro de las cinco características esenciales de Cloud Computing según el NIST [11][12][13] tenemos:

1. **Bajo demanda y Autoservicio:** El usuario dispone automáticamente de las distintas necesidades de recursos que el proveedor de servicios de cloud brinda sin necesidad que éste último tenga que realizar intervenciones manuales. El usuario del cloud puede configurar a su criterio las capacidades de cómputo tales como tiempo de servidor, almacenamiento, memoria RAM, cantidad de procesadores, velocidad de red, etc. Esta característica aporta un gran beneficio al usuario dado que reduce en gran medida las complicaciones y costos que normalmente conllevan la adquisición de recursos propios IT, siendo así uno de sus grandes beneficios.

2. **Amplio acceso a la red:** El Cloud Computing permite el acceso a los datos desde cualquier lugar. Solo se necesita un navegador web y conexión a Internet para disfrutar de los servicios en la nube, no hace falta tener un sistema operativo determinado o instalar un software específico en cada cliente. La combinación de dispositivos móviles (tablets y smartphones) y fijos crea nuevas oportunidades en el desarrollo de la actividad empresarial, permitiendo plena operatividad. Esta característica es especialmente importante en organizaciones distribuidas geográficamente, permitiendo el acceso a los recursos con independencia de aspectos como la ubicación y la jornada laboral. Es importante puntualizar que esta característica establece una limitación, ya que no es posible utilizar las aplicaciones en la nube si no hay conexión a Internet.

3. **Pooling de recursos:** Esta característica permite a los distintos proveedores compartir sus recursos tanto físicos como virtuales entre los distintos usuarios, disminuyendo costes, maximizando la disponibilidad y de acuerdo a los requerimientos de los usuarios. Para poder habilitar este modelo multi-tenant de forma óptima, es necesario que los recursos disponibles

(capacidad de computación, almacenamiento, velocidad de red, etc.) se asignen y balanceen de forma automática en base a las distintas peticiones de los usuarios. Los usuarios pueden ignorar el origen y la ubicación de los recursos a los que acceden, aunque sí es posible que sean conscientes de su situación a determinado nivel, como el país donde se localiza el CPD (centro de procesamiento de datos).

4. Escalabilidad y rapidez: La sencillez con la que se pueden añadir o eliminar recursos supone una ventaja frente al modelo tradicional. En Cloud Computing es posible añadir o eliminar recursos en cuestión de minutos, aumentando o disminuyendo el almacenamiento o el número de procesadores sin que la aplicación se vea afectada. En el ámbito software, la flexibilidad es muy alta, pudiendo incorporar nuevas funcionalidades a todos los usuarios de forma más rápida que sobre sistemas tradicionales.

5. Servicio medido: Los sistemas en la nube controlan automáticamente y optimizan el uso de recursos mediante una capacidad de medición a algún nivel de abstracción adecuado al tipo de servicio; por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas. El uso de estos recursos puede ser monitoreado, controlado y reportado, proporcionando transparencia tanto para el proveedor como para el consumidor por el servicio utilizado. Esto posibilita diferentes modalidades de pago como:

- Pago por disponibilidad del servicio: se acuerda un precio por el tiempo en el que los recursos contratados están habilitados al usuario.
- Pago por uso: se basa específicamente en los servicios consumidos por el usuario (almacenamiento realizado, transacciones comerciales realizadas, etc.), de forma análoga a como se paga el servicio de electricidad, gas o agua potable.
- Pago por paquetes escalables: el pago se realiza por reservas de slots fijos que se pueden incrementar en unidades acotadas.

4.3 - Modelos de servicio

Desde que se ha desarrollado la computación en la nube, los proveedores de cloud ofrecen diferentes modelos de servicio asociados a ellos. De esta forma, en la actualidad existe una gran cantidad de modelos de servicio que siguen la siguiente forma: XaaS, or <Something> as a Service, o en español, <Cualquier cosa> como Servicio. [12]



Figura 4.2. Pila de Servicios en Cloud Computing [14]

Sin embargo, existen tres tipos de servicios que han sido aceptados universalmente (conocido como modelo SPI) y que se pueden ver en la Figura 4.2, estos son:

1. **Software como Servicio (Software as a Service - SaaS):** en este modelo, el proveedor pone a disposición una aplicación completa como servicio para el usuario final. Este servicio se ejecuta como una instancia única de software que corre en la infraestructura que el proveedor gestiona. Los clientes acceden a estas aplicaciones a través de interfaces de cliente livianas como navegadores web, dispositivos móviles o interfaces de programas. Entre los servicios más comunes que podemos encontrar están: Google Maps, Google Docs, Microsoft Windows Live, Twitter, Facebook, entre otras.

2. **Plataforma como Servicio (Platform as a Service - PaaS):** este servicio cuenta con un ambiente de programación y un ambiente de ejecución que está dirigido más que nada a desarrolladores en lugar de usuarios finales. Entre algunos de los servicios más conocidos podemos encontrar la App Engine de Google, Azure de Microsoft, o Facebook Platform de Facebook, donde cada uno de estos servicios tienen un lenguaje de programación y herramientas que brinda el proveedor. El cliente no controla ni gestiona la infraestructura, sólo es responsable de instalar y desplegar las aplicaciones que desarrolla.

3. **Infraestructura como Servicio (Infrastructure as a Service - IaaS):** el proveedor del servicio le brinda a los clientes procesamiento, almacenamiento, dispositivos de red y otros recursos

fundamentales, en la mayoría de los casos a través de la tecnología de virtualización, en donde cliente es el responsable de instalar el sistema operativo y aplicaciones que le sean necesarias, y gestionar el almacenamiento y la interacción del usuario con el sistema.

Como se mencionó anteriormente, los modelos de servicio que ofrece la nube siguen el esquema XaaS (“Cualquier cosa como Servicio”) como regla general, sin embargo, cualquier modelo de servicio que una organización desee establecer gira en torno al modelo SPI (SaaS, PaaS, e IaaS), esto significa que las organizaciones utilizan algún servicio del modelo SPI como base para ofrecer los servicios propios. Entre algunos ejemplos de modelos de servicios que generan las organizaciones podemos encontrar: StaaS (Storage as a Service), se brinda espacio de almacenamiento como servicio; IaaS (Identity as a Service), pensado como un inicio de sesión único para la nube; SaaS (Security as a Service), modelo de tercerización para la gestión de la seguridad; entre muchos más.

Para BPM también existe un modelo de servicio en cloud computing llamado BPaaS, o Business Process as a Service, que consiste en un modelo de servicio donde las aplicaciones que se ofrecen en cloud son del tipo procesos de negocio o workflows. Este modelo se utiliza como uno de los temas centrales de este trabajo, y se presenta en la siguiente sección.

4.4 - Procesos de negocio como servicio

Procesos de negocio como Servicio (BPaaS) es un modelo de servicio en el cloud donde las aplicaciones que se ofrecen son del tipo procesos de negocio o workflows.

BPaaS, al igual que varios modelos de servicios en cloud, permite a los usuarios utilizar un BPMS situado en la nube bajo una modalidad de “pago por uso”, en lugar de enfrentarse a grandes costos de inversión en software, hardware y mantenimiento. Por esto, con la popularización de la computación en la nube, se espera que BPaaS sea una solución para aquellas pequeñas y medianas organizaciones que no pueden afrontar los costos de un software empresarial. [19][15][16]

Es común ver a BPaaS como un modelo especial situado por sobre la capa del modelo SaaS, en el que los proveedores de la nube proporcionan métodos para el modelado, utilización, personalización, y ejecución distribuida de procesos de negocios [16].

Dado que BPM es una disciplina que ayuda continuamente a las organizaciones a optimizar los procesos operacionales que tienen un gran impacto en los objetivos de negocio, las organizaciones utilizan cada vez más los BPMS para modelar y ejecutar los procesos de negocios, administrar tareas humanas y de los sistemas, monitorear y auditar el rendimiento de los procesos de negocio y mejorarlos continuamente. Por esto, la integración de un BPMS en un modelo SaaS provee un ambiente flexible y a bajo costo, no solo para aquellas pequeñas y medianas organizaciones que desean tener una infraestructura de IT que pueda incrementar el nivel de servicio que se les brinda a sus clientes, sino también para organizaciones de gran magnitud que se ven envueltas en grandes costos de instalación, despliegue, mantenimiento y actualización de su infraestructura IT.

Básicamente, este modelo de servicio necesita de dos componentes principales para el desarrollo y ejecución de sus procesos, que son utilizados para lograr las funcionalidades de un workflow tal como se ejecutaría en un sistema embebido:

- BPMS (Business Process Management System): herramientas para diseñar, ejecutar, monitorear y optimizar los procesos de negocios.
- BAM (Business Activity Monitoring): herramienta para monitorización de ejecución de procesos de negocio en tiempo real.

Sin embargo, una desventaja en la subcontratación de la ejecución de los procesos de negocio, es que los usuarios pierden el control de los datos sensibles que hacen al funcionamiento de sus organizaciones. También, al mantener toda la aplicación en la nube, se estaría desaprovechando el poder de cómputo del cloud cuando las actividades de los procesos que se despliegan en ese entorno no requieren de alta capacidad en su performance pero sí requieren un frecuente acceso a datos, de esta manera se intensifica el tráfico y hace más costoso el servicio.

Una de las soluciones a estas desventajas que se han nombrado anteriormente es crear un sistema híbrido, formado por un motor de procesos de negocio en el cloud y otro local, en el cuál los procesos se descomponen en dos tipos de actividades, aquellas que requieren mayor poder de procesamiento y no manejan información sensible, y aquellas que manejan información sensible, o gran volumen de información. Al descomponer los procesos de negocio en estos dos tipos de actividades podemos situar a las primeras en la nube, y a las segundas en un BPMS local. [15] [20]

4.5 - BPMS en ambientes distribuidos

Si bien los BPMS basados en Cloud Computing pueden ayudar a las pequeñas y medianas organizaciones a mejorar la eficiencia de los procesos y reducir sus gastos de infraestructura, todavía existen algunas barreras al utilizar este servicio.

Los dos temas principales de estudio en la adopción y del uso de un BPMS en el cloud son:

-Protección de la privacidad: algunos datos gestionados por los procesos de negocios pueden ser confidenciales y las organizaciones no están dispuestas a correr el riesgo de un posible robo de información. Si bien los servicios de cloud se encuentran ocultos detrás de barreras de seguridad provistas por la infraestructura de red y tienen un avanzado mecanismo para mantener la división de información entre sus clientes, los dueños de la información sensible temen perder el control de estos datos, o puede ocurrir que por reglas contractuales con sus clientes no se les permita almacenar información de éstos en otros lugares que no sean servidores propios.

- Actividades que no son altamente computacionales: por otro lado, la eficiencia y efectividad de las actividades que no son altamente computacionales puede disminuir debido a que la transferencia de los datos puede tomar más tiempo que el procesamiento mismo de los datos. Además, los costos de la actividad pueden incrementarse debido a que la transferencia de los datos es uno de los elementos de facturación en un sistema de cloud computing.

Hoy en día, en la mayoría de las soluciones donde se utilizan BPMS, el motor de proceso, las actividades y los datos de los procesos se encuentran alojados en un mismo sitio, ya sea en un sistema embebido o en un ambiente de cloud computing, donde presentan las ventajas e inconvenientes de cada uno de estos. Para poder utilizar los principales beneficios de ambos sitios, esto es, privacidad de datos en un sistema embebido, y poder de cómputo y ubicuidad relacionado a un sistema de cloud, se ha investigado en [21] un modelo de distribución, denominado PAD (Proceso-Actividad-Datos), en el que el arquitecto de procesos de negocio puede separar uno de estos procesos de acuerdo a los beneficios que brindan cada una de estas ubicaciones. Los ambientes distribuidos en el contexto de los procesos de negocio favorecen el rendimiento y proponen incorporar el concepto de descomposición de procesos, permitiendo que los mismos se ejecuten tanto en un entorno cloud o embebido. [19][15][20][21]

4.6 - Combinación de esquema embebido y cloud

Al diseñar una arquitectura descentralizada basada en un BPM cloud y uno embebido, se debe tener en cuenta en dónde desplegar los procesos, dónde ejecutar las actividades, como así también donde almacenar los datos producidos y consumidos por las aplicaciones. Para analizar la combinación entre un esquema embebido y uno de cloud se plantean tres aspectos: arquitectura, control de flujo y optimización de la distribución. [20] [21]

Arquitectura: en la mayoría de las soluciones BPM el motor de procesos, las actividades y los datos del proceso se localizan en el mismo lado, tanto en un sistema embebido como en el cloud. Existen investigaciones que introducen el modelo PAD (Proceso-Actividad-Datos) de la Figura 4.3 como una posibilidad de distribución de BPM en el cloud. En el mismo el motor de procesos, las actividades involucradas en el mismo y sus datos están distribuidos.

El modelo PAD define cuatro posibilidades de distribución:

1. El primer patrón de solución es el enfoque tradicional de BPM donde todos los elementos están alojados en el usuario final.
2. El segundo patrón es útil cuando el usuario ya tiene un sistema BPM, pero las actividades con intensidad de cómputo se localizan en el cloud para incrementar su performance.
3. El tercer patrón es útil para los usuarios que aún no poseen un sistema BPM; en este caso podrían adoptar un esquema de cloud de manera de pago por uso, donde las actividades sin intensidad de cómputo y los datos sensibles se pueden localizar en el usuario final.
4. El cuarto patrón es el modelo basado en cloud donde todos los elementos se localizan en la nube.

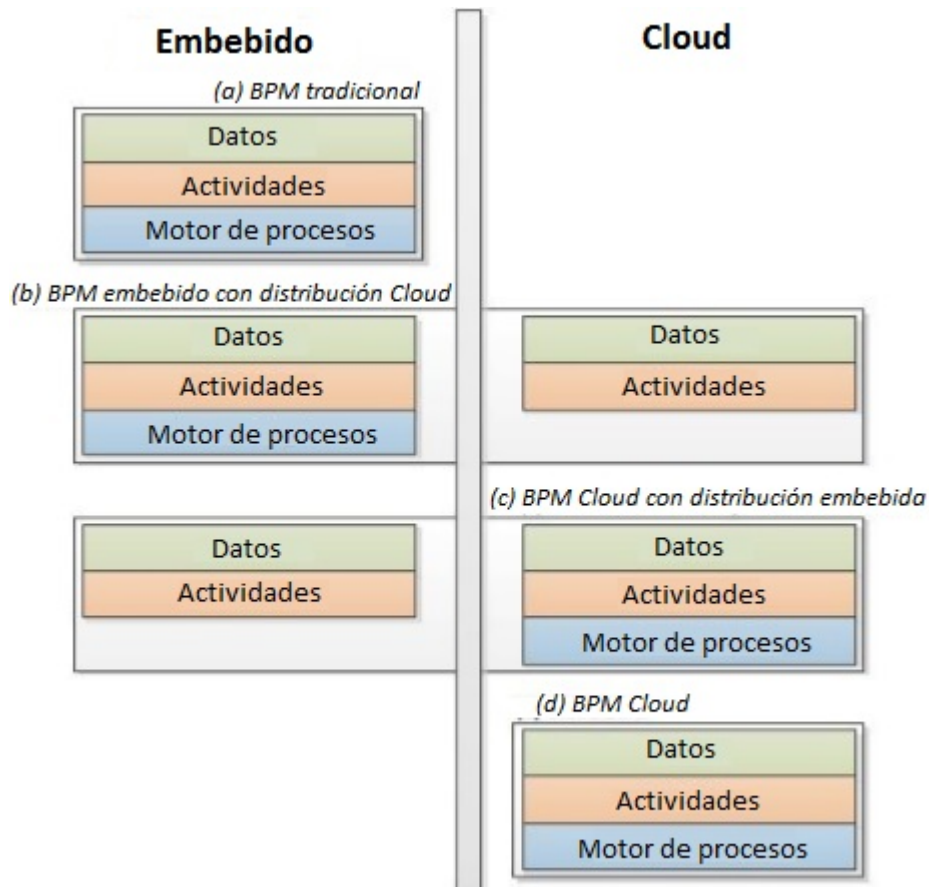


Figura 4.3. Esquema de distribución Proceso-Actividad-Datos [15][21]

Control de Flujo: Los procesos de negocio consisten de dos tipos de flujos: de control y de datos. Los flujos de control regulan las actividades que se ejecutan y cuál es la siguiente actividad en ejecutarse, mientras que los flujos de datos determinan cómo los datos se envían de una actividad a la otra dentro del proceso, y cómo se realiza el mapeo de estos datos. Los motores de BPM deben lidiar con el control de ambos flujos. Un flujo de datos puede contener datos sensibles, por lo tanto, cuando se despliega un motor de BPM en el cloud, se debe proteger el contenido de los mismos.

Distribución óptima: los costos de un sistema de cloud han sido propósito de estudio en diversos artículos tales como [22] y [23], en los cuales se investiga a los proveedores de cloud como Amazon, Google, eBay, entre otros, y se plantea como encontrar el balance de costo y rendimiento utilizando una aplicación que se basa en cloud computing para la realización de las tareas que requieren mayor poder de procesamiento.

4.7 - Descomposición de procesos

Es posible generalizar la distribución PAD e identificar un quinto patrón en el cual el motor de procesos, las actividades y los datos se despliegan en la nube y en el usuario final, tal como se puede observar en la Figura 4.4.

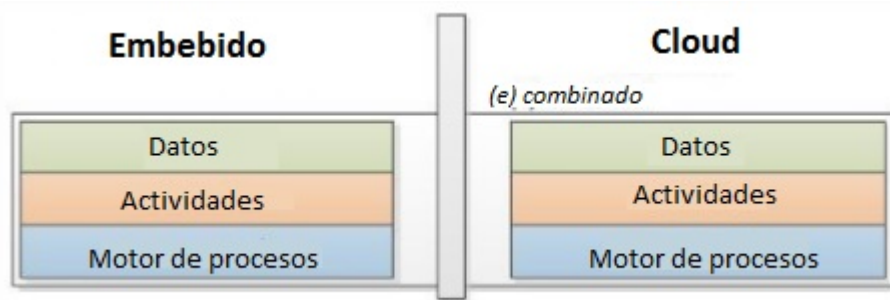


Figura 4.4. Quinto patrón de distribución PAD [15]

Esta solución presenta dos beneficios potenciales:

1. El motor de procesos regula el flujo de control y el flujo de datos. Una actividad recibe datos del motor de procesos y luego de su ejecución los datos que son producidos se pasan de nuevo al motor de procesos.
2. Cuando la nube no se encuentra accesible, los usuarios pueden ejecutar los procesos de negocio en forma completa en el sistema embebido hasta que el primero vuelva a estar disponible.

Para observar la diferencia entre un motor de procesos situado en un solo lugar, como por ejemplo en el patrón 2 del esquema de distribución PAD, y motores de procesos situados en ambos sitios, veamos el siguiente ejemplo.

Consideremos que una secuencia de actividades se ubica en la nube, mientras que el motor de procesos se despliega en el usuario final. Cada actividad utiliza los datos de salida de la actividad previa como entrada. Los datos no se envían directamente de una actividad a la otra sino que son enviados al motor de proceso primero (Figura 4.5(a)).

Debido a que la transferencia de datos es uno de los factores de facturación en el modelo de cloud computing, estas situaciones pueden volverse más caras cuando se transfieren grandes cantidades de datos entre actividades.

Para evitar este problema se puede agregar un motor de procesos al cloud, el cual regula el flujo de control y el flujo de datos entre las actividades situadas en él (Figura 4.5(b)). Cuando una secuencia de actividades se localiza en el cloud, los datos se regulan por el motor de procesos en el cloud, lo cual reduce la cantidad de datos a ser transferidos entre este y el sistema embebido.

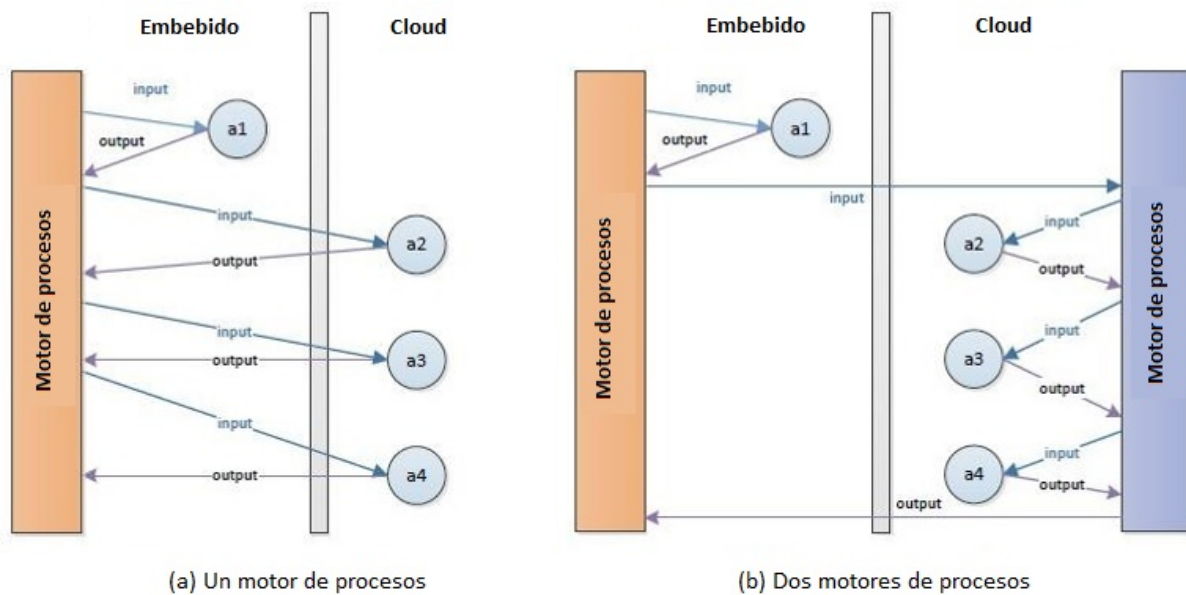


Figura 4.5. Datos enviados entre actividades coordinadas por motores de procesos [15]

Para poder correr un mismo proceso de negocio en dos motores de proceso separados, el mismo debe ser dividido en dos procesos individuales. Puede llegar a ser conveniente para los usuarios del BPMS tomar una lista de distribución del proceso de negocio y sus actividades, la cual puede ser automáticamente transformada en dos procesos de negocio, uno en el cloud y otro en el sistema embebido. [19][15][20]

Una aproximación posible para manejar la descomposición del proceso es identificar la estructura y la semántica del mismo (Figura 4.6). Esto significa que se deben identificar aquellas actividades del proceso de negocio en donde la ejecución de las mismas dan la pauta de atomicidad, y que su ejecución, ya sea en el sistema embebido o en el cloud, no modificaría el comportamiento del proceso de negocio original. Una estrategia para detectar aquellas actividades atómicas es tratar de observar las dependencias de control y de datos que tienen con otras actividades y con el motor de procesos en el cual se ejecutan. Por ejemplo, si existe una actividad que utiliza información confidencial que obtiene del sistema embebido, lo más probable es que no sea recomendable ejecutar la actividad en el motor de procesos situado en el cloud, ya que correríamos peligro de exponer tales datos. Al identificar las dependencias de control y de datos, se pueden investigar las consecuencias de mover ciertas actividades del sistema embebido al cloud y viceversa.

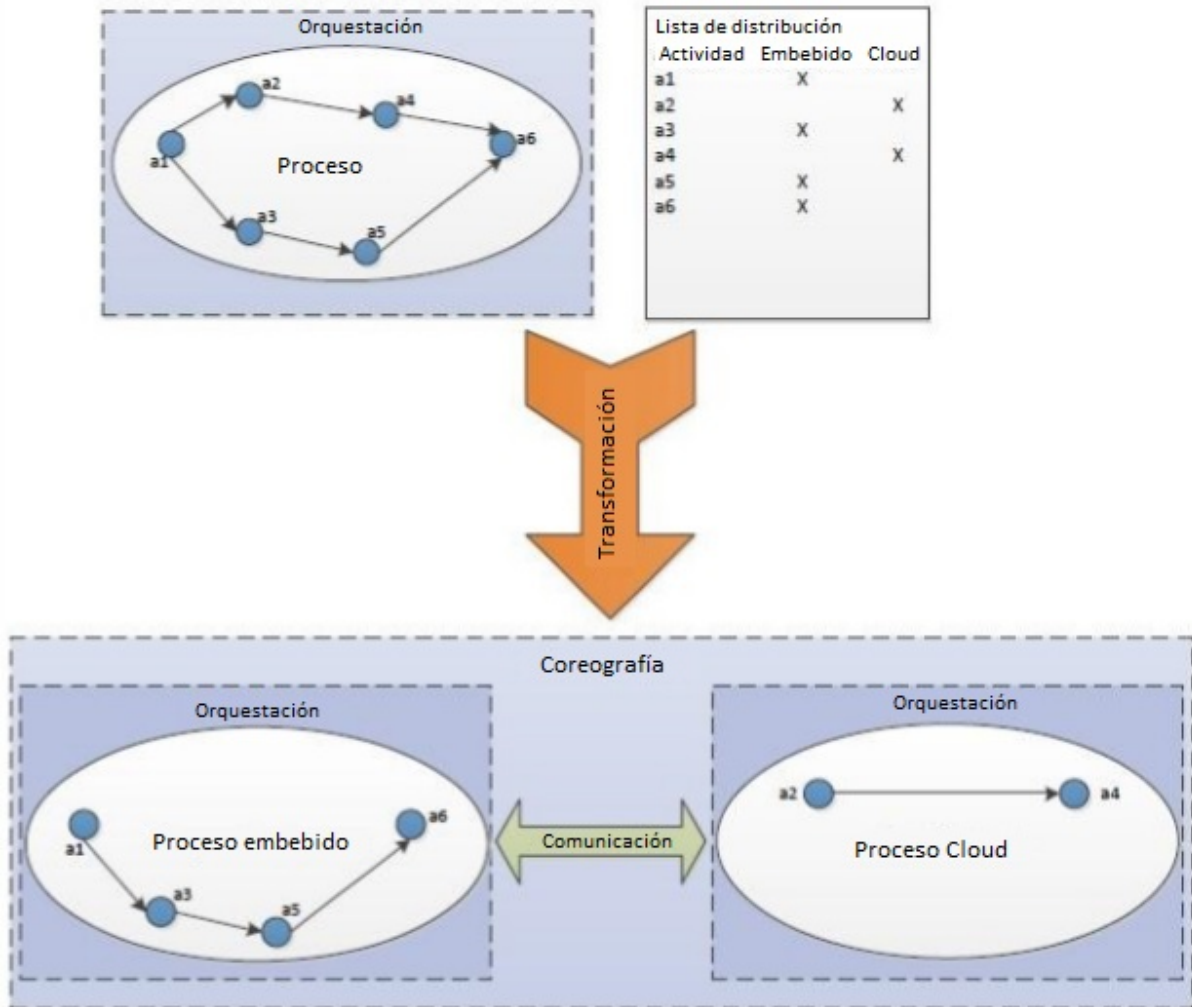


Figura 4.6. Descomposición de procesos[15]

Cuando se conocen las consecuencias de la distribución de actividades, se puede crear una transformación de modelo en la cual un proceso de negocio y una lista con marcas se usan para crear dos procesos individuales, uno para el cloud y otro para el usuario final. Además, se puede generar una descripción de la coreografía para describir la comunicación entre ambos procesos de negocio utilizando algún lenguaje estándar.

4.8 - Distribución óptima de las actividades

Evitar que los datos sensibles se transfieran entre el sistema embebido y sistema de cloud es un método directo y eficaz para proteger la privacidad de los datos de los usuarios. Sin embargo, a veces, los usuarios quieren utilizar los recursos en ambos sitios para poder aprovechar el alto procesamiento y la capacidad de almacenamiento masivo de la nube, sobre todo si los datos que intervienen en el proceso no requieren un alto nivel de privacidad.

Existen varios esquemas acerca de la distribución de una actividad y los datos que maneja. En la Figura 3.5 (revisar si vale la pena poner) se pueden ver ocho esquemas básicos de distribución de una actividad, y sus entradas y salidas de datos en ambos entornos. La parte superior de cada rectángulo representa la distribución en el cloud, y la inferior representa la

distribución en el sistema embebido. El símbolo A representa a la actividad, mientras que los símbolos I y O representan los datos de entrada y salida respectivamente. Si bien las actividades pueden tener varias entradas y salidas de datos de diferentes lugares, se presentan los casos básicos de ejemplo para lograr una mejor comprensión de la distribución.

Para poder obtener el mayor beneficio correspondiente al despliegue de los subprocesos de negocios en ambos sistemas, es necesario determinar el costo que supondrá situar las diferentes actividades que componen al proceso.

Por lo general, un sistema embebido tiene un poder de cómputo relativamente menor aunque un alto nivel de privacidad. Un sistema de cloud tiene mejor rendimiento computacional pero un bajo nivel de privacidad. Mientras que el sistema embebido tiene un costo fijo, los usuarios deben pagar el servicio de cloud por los datos transferidos y las horas de CPU consumidas. Por esto, para determinar la distribución óptima de las actividades es necesario calcular los costos del sistema BPM en el cloud de acuerdo los costos de tiempo, los costos monetarios y los costos por el riesgo de privacidad.

- Costo de Tiempo: el tiempo de ejecución de una actividad en el cloud suele ser menor que en el sistema embebido, especialmente cuando se realizan tareas de procesamiento intensivo ante requerimientos concurrentes de gran tamaño. Sin embargo, en algunas circunstancias, cuando la ejecución en el cloud no requiere demasiado procesamiento, la transmisión de los datos podría ser mayor que el procesamiento en el cloud. Por esto, para lograr el mayor beneficio de costo de tiempo, se debe buscar un equilibrio entre los datos enviados y el procesamiento en el cloud, tomando como referencia los resultados que se lograrían al procesar la misma actividad en el sistema embebido.

- Costo Monetario: dado que el procesamiento, tanto como la transferencia desde y hacia el cloud son cobrados por los proveedores en término de horas de CPU utilizadas y GB (GigaBytes) de datos transferidos, hay que estar seguros de no perder dinero ejecutando las actividades en el cloud cuando se podría ejecutar en el sistema embebido sin mucha diferencia en el procesamiento.

- Costo de Privacidad: la privacidad de los datos es otro tema a considerar. Las organizaciones, al situar datos sensibles o documentos privados en el cloud, deben estudiar el riesgo de impacto que supondrán la fuga de tal información, y aceptar o rechazar tomar dichos riesgos.

Mediante el uso de estas fórmulas los usuarios pueden hacer una estimación de los costos de desplegar partes de sus aplicaciones en un sistema embebido y en la nube. Debemos recordar que un criterio clave desde el punto de vista económico en un sistema de cloud es la tasa de transferencia de datos desde y hacia el servidor. [21]

4.9 - Monitoreo de procesos de negocio distribuidos

En las soluciones orientadas a procesos de negocios existen dos objetivos principales. Primero, la capacidad de proveer flexibilidad en los procesos de negocio, permitiendo una

rápida reacción a los cambios del mercado. Y segundo, proveer una buena visualización de los procesos y permitir una optimización eficiente de estos.

Para poder optimizar los procesos de negocios tenemos que saber qué es lo que está funcionando mal o es ineficiente, por lo tanto, debemos comenzar a medir los procesos. Es aquí donde se puede utilizar una herramienta de monitoreo llamada Monitoreo de Actividades Empresariales, o en inglés, Business Activity Monitoring (BAM). Esta herramienta que se encuentra dentro de muchos BPMS nos permite medir varios aspectos de los procesos de negocio tales como tiempo, costos, rendimiento, estados y resultados de operaciones, procesos y transacciones.

La optimización de los procesos de negocio es una tarea difícil y requiere de un buen conocimiento de los procesos de negocio dentro de la organización, por esto, aunque se cuente con la mejor herramienta de monitoreo, la optimización de los procesos de negocio dependen de las habilidades de la persona responsable de estos. Las personas encargadas de optimizar los procesos de negocio tienen que tener en cuenta varias tareas tales como identificar aquellas actividades que están teniendo poca eficiencia (ya sea en tiempo, costo o calidad); determinar los factores internos y externos que afectan la decisión de optimizar un proceso y cómo éstos influyen con el dueño del proceso, los dueños de las tareas y las áreas específicas de la organización; aplicar reingeniería, buenas prácticas o rediseño del proceso; definir nuevas medidas de rendimiento del nuevo proceso (KPIs – Key Performance Indicator), realizar simulaciones y monitorearlos para confirmar que ha sido mejorado.

BAM provee acceso a información de los procesos en tiempo de ejecución, permite un análisis en tiempo real de los procesos de negocio, muestra los cuellos de botella en las tareas, mide el tiempo de cada tarea y provee herramientas para visualizar toda esa información [17]. Además, se utiliza para asegurar que los procesos de negocio funcionan como es esperado, simplificar la información compleja relacionada a los procesos y mostrarla oportunamente.

Como se ha dicho anteriormente, muchos BPMS contienen módulos BAM para realizar el monitoreo de los procesos de negocio que se están ejecutando en el motor de procesos. Por ejemplo, Bonita Open Solution [27], un BPMS libre y open source, en la cual observamos los distintos módulos que componen a la suite, entre ellos el módulo BAM de monitoreo. Sin embargo, estos módulos de monitoreo funcionan sólo dentro del BPMS donde se están ejecutando los procesos. Por esto, en el caso de la descomposición de procesos, el mayor de los problemas de poseer un esquema de procesos particionados, es la recuperación y monitoreo de las distintas instancias distribuidas (ya sea en un sistema embebido o dentro del cloud), y a su vez lograr dar un esquema integrador de las mismas bajo la óptica del “proceso original” al cual pertenecen.

En el marco de investigación en [19] se desarrolló una aplicación web PHP que permite la ejecución y monitoreo de procesos distribuidos en BonitaOS. Como continuación, este trabajo propone la extensión de dicha herramienta para poder obtener los logs de eventos de dichos procesos distribuidos y aplicarles técnicas de minería de procesos. En el siguiente capítulo se detallarán las herramientas a utilizar para lograr este objetivo.

Capítulo 5 - Software a utilizar y mecanismos de integración

La generación de log de eventos y la ejecución y monitoreo de procesos distribuidos ya han sido abordados en [18] y [19] respectivamente, pero de manera aislada uno del otro. De ellos se heredan las herramientas de software que este trabajo toma como base para lograr aplicar técnicas de minería de procesos en ambientes distribuidos. En este capítulo se describirán dichas herramientas.

5.1 - Bonita Open Solution

Bonita Open Solution (BonitaOS) es una herramienta libre y open source que ayuda a los usuarios a crear aplicaciones basadas en procesos de negocio. Ha sido creada por BonitaSoft [27], una de las empresas que son líderes en el mercado de la gestión de procesos de negocio. Al estar completamente desarrollada en Java y ser open source, se puede hacer uso de esta herramienta para realizar nuestros propios desarrollos sin la necesidad de realizar contratos aparte con la empresa BonitaSoft.

Entre sus características principales podemos destacar las siguientes:

- Es una aplicación potente y ligera. Se destaca sobre todo por la facilidad en su utilización debido al diseño intuitivo de los diferentes elementos que lo componen.
- Tiene soporte para Business Process Model and Notation (BPMN) versión 2.0, lo cual ofrece a los usuarios la posibilidad de diseñar sus proyectos de una manera transparente, ágil y con la capacidad de expresar un proceso de negocio de una forma que sea comprensible para el público comercial y el público técnico de igual manera.
- Soporta varios formatos de archivos (proc, bar, BPMN, XPDL y jBPM) para la importación de procesos de negocio. Además permite realizar la exportación de los diseños de los procesos a formatos de archivo de documentos o imágenes tales como pdf, jpeg, png, bmp, gif y svg, para poder visualizar los procesos sin necesidad del uso de la suite.

Un BPMS como Bonita, además de ofrecer las ventajas propias de su orientación a procesos, presenta otras ventajas en cuanto al desarrollo de aplicaciones basadas en procesos. El hecho de poder modificar el modelo en BPMN y automáticamente modificar toda la aplicación con los nuevos cambios, de forma que cualquier cambio en la lógica de negocio no suponga modificar toda la aplicación, es una cualidad muy beneficiosa, ya que esto reduce enormemente los tiempos de desarrollo y testeado para el mantenimiento y adaptación de las aplicaciones. Con esta capacidad de los BPMS abandonamos la visión monolítica de las aplicaciones basadas en plataformas y abrimos el camino para trabajar en la mejora continua de los procesos. Además, es posible gestionar versiones de los procesos, pudiendo reutilizar procesos ya existentes para la generación de nuevos procesos adaptados o mejorados. [27]

5.1.1 - Módulos de Bonita OS

La aplicación Bonita Open Solution en su versión 5 está compuesta de varios módulos o componentes, como puede verse en la Figura 5.1.

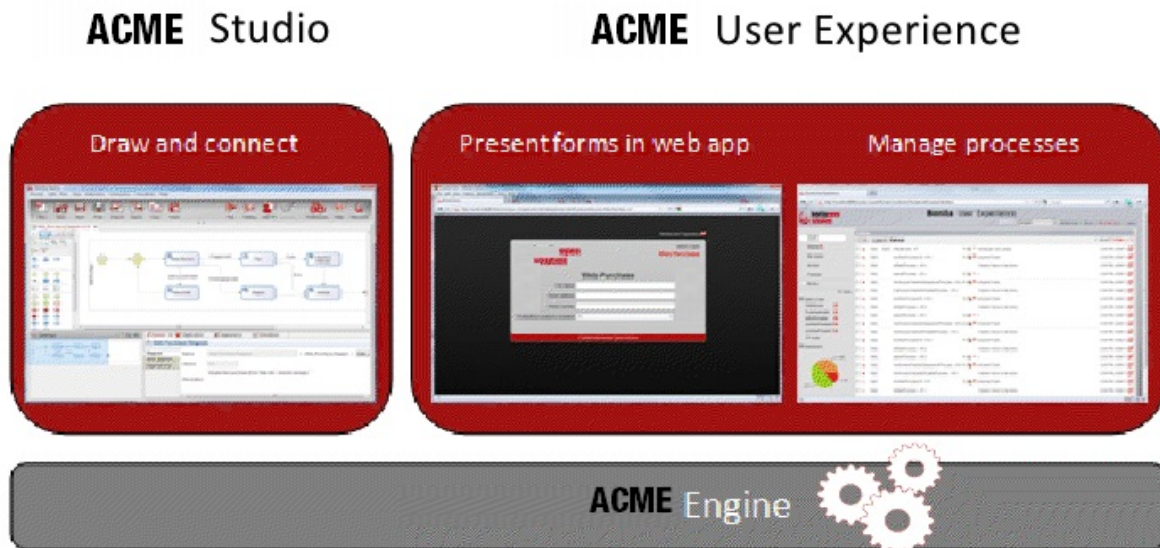


Figura 5.1. Módulos que componen a Bonita Open Solution [27]

Los módulos que se destacan son:

- Bonita Studio: interfaz gráfica cuya función es diseñar los procesos BPM usando la notación BPMN.
- Bonita Form Builder: módulo en el que se definen los formularios que habrán de ser rellenados por los usuarios.
- Bonita Execution Engine (BEE): motor de procesos que se encarga del despliegue y ejecución.
- Bonita User Experience (User XP): interfaz similar a un cliente de correo electrónico para gestionar los pasos, actividades y procesos

Veamos más en detalle las características del motor que será con el que va a interactuar nuestra aplicación.

Bonita Execution Engine (BEE)

Es el motor de procesos de Bonita y se encarga de la conexión de los procesos que existen en el sistema, así también como el despliegue y ejecución de los procesos. El módulo de Bonita Studio está conectado directamente a este otro módulo para funcionar. Este motor es genérico y extensible por lo que siempre seremos capaces de añadir con mayor o menor dificultad nuevos estándares o bien servicios que puedan aparecer en el mundo de BPM con posterioridad.

El módulo Bonita Execution Engine (BEE) es uno de los tres principales módulos de Bonita OS. Es un módulo no intrusivo, ya que solo requiere una Java Virtual Machine (JVM) para ejecutarlo. Es la base de procesamiento, la cual se ejecuta en tiempo real en background y

conecta a Bonita Studio, los formularios creados en el Form Builder, al módulo Bonita User Experience y las aplicaciones externas que utiliza la organización.

Bonita Open Solution se basa en la BEE para crear, acceder y procesar los datos a través del uso de las diferentes interfaces de programación de las aplicaciones (APIs), además de gestionar y ejecutar los procesos creados en Bonita Studio.

La BEE, también conocida como Runtime, fue diseñada para proveer la máxima flexibilidad a través de la inyección de servicios. La BEE es completamente configurable utilizando un archivo XML llamado bonita-server.xml, que puede encontrarse en la carpeta de configuración del servidor. Este archivo de configuración describe todos los servicios utilizados por defecto, aunque es posible cambiar cualquiera de ellos o reemplazarlos por los que se ajusten con la implementación de la aplicación.

Este es el módulo del cual hacemos uso en este trabajo. Contamos con por lo menos dos motores distribuidos (ya sean en la red local o en la nube) que son accesibles mediante sus API REST correspondientes. Con solo desplegar los procesos en el motor, nos comunicamos con él a través de la red para que cree instancias y nos devuelva información sobre el estado y valor de sus variables. Esto nos permite tanto la ejecución como el monitoreo remoto.

5.1.2 - Capacidad de extensión de Bonita

Como ya se ha dicho antes, Bonita Open Solution es open source y libre, por lo que es posible crear un repositorio y realizar una nueva rama (branch) para modificar el código y crear nuestra propia aplicación Bonita, para incluir funcionalidad que quizás tengamos en una aplicación que ya disponemos. Sin embargo, esta es una tarea que requiere mucho trabajo, además del conocimiento necesario que se debe tener de la arquitectura de Bonita.

Para evitar tener que modificar el código del motor de procesos, Bonita nos permite utilizar conectores predefinidos que se encuentran incluidos en la suite, o crear los propios de acuerdo a las necesidades del proceso de negocio. Los conectores son componentes básicos utilizados para modelar las interacciones entre dos o más componentes de software. Los conectores funcionan en tiempo de ejecución y por lo general se comunican entre dos objetos, o entre un cliente y un servidor, mediante mensajes asincrónicos, multicast o streams de datos. Principalmente se utilizan para la gestión de información y para la ejecución de funcionalidades específicas que contienen los sistemas de información.

El motor de Bonita es extensible, esto significa que puede ser usado con la plataforma que tienen Bonita por defecto, es decir con Bonita User Experience, o ser consumido como un EJB (Enterprise JavaBeans) o por HTTP a través aplicaciones externas haciendo uso de la API REST que nos otorga. De esta manera, es posible integrar los procesos de negocio que gestiona Bonita hacia un sistema de información o a una aplicación propia, y ejecutar los métodos que se encuentran en las diferentes APIs que contiene el motor de procesos de Bonita (BEE).

5.1.3 - Conectores de Bonita OS

Bonita Open Solution provee conectores para conectar una tarea (o actividad) o un proceso (pool) hacia sistemas de información externos. Los conectores toman una entrada específica (directamente como un valor que el usuario ingresó en la actividad o construidos en una expresión) y ejecuta código Java. Algunos conectores también reciben el resultado de la llamada realizada, que es útil para informar a la actividad si el proceso de efectuar dicha llamada resultó exitoso, o por ejemplo, si lo que se desea es obtener datos de una base de datos externa, el conector se encarga de devolver los datos a la actividad que ejecutó el conector y realizó la consulta. (Figura 5.2)

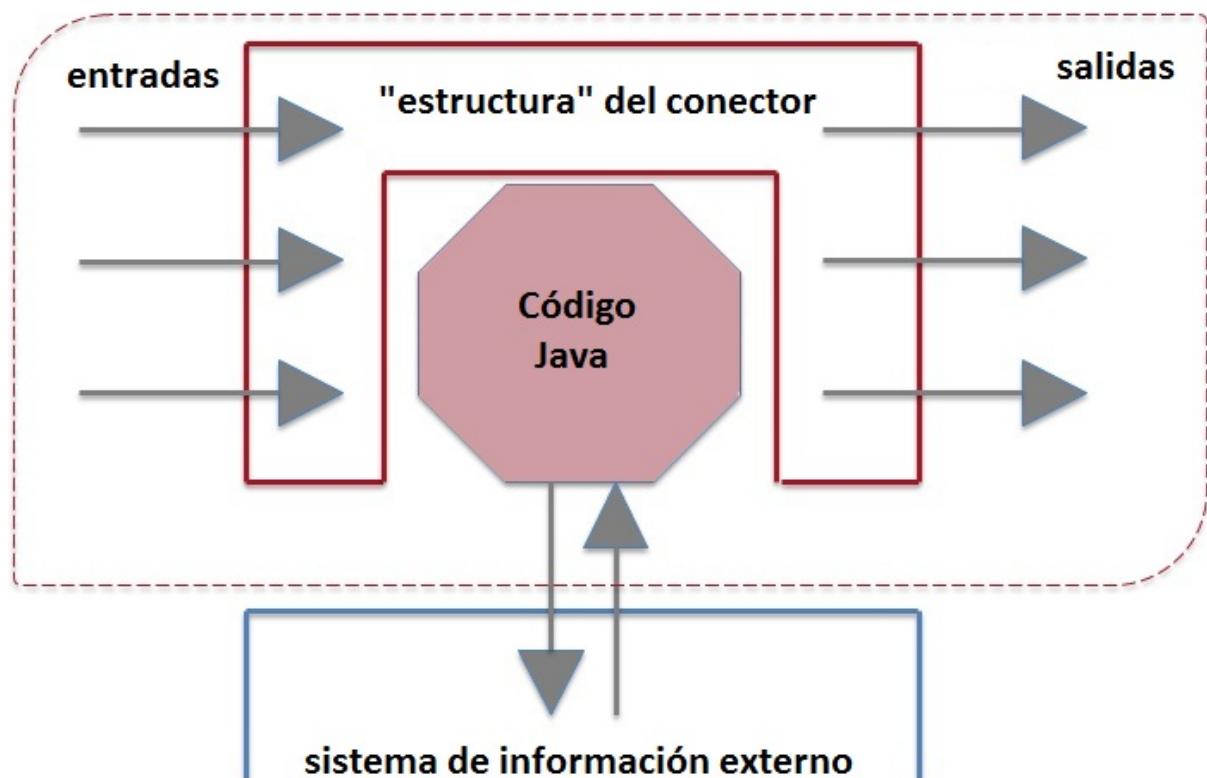


Figura 5.2. Estructura y ejecución de un conector en Bonita OS. [27]

Una gran biblioteca de conectores también es un elemento esencial para la viabilidad a largo plazo de una solución de BPM. Al agregar conectores a las actividades se permite que los procesos de negocio se integren con aplicaciones o herramientas de la organización. Los conectores permiten separar el proceso de definición del proceso de implementación, tal como se verá más adelante. Una de las ventajas que los caracterizan es que permiten reutilizar código ya sea, utilizando conectores que ya se encuentren en la biblioteca, creando conectores propios, o bien descargando e instalando conectores creados por otros desarrolladores de la comunidad.

Estos conectores se utilizan para operar dentro del flujo de trabajo de BPM con sistemas de terceros que tienen distintos tipos de servicios y aplicaciones (como bases de datos, mensajería, ERP's, ECM's, data warehouse, CRM's, entre otros). De esta manera podemos

encontrar conectores para MySQL, Oracle, MSSQL Server, Jasper, SAP SalesForce, Alfresco, Sugar CRM, etc.

Los proveedores de BPM de código abierto como Bonita OS complementan las ofertas con las contribuciones de su base de usuarios y de la comunidad más grande para mejorar la versatilidad del producto. Los productos de BPM de código abierto como el conjunto de aplicaciones de BonitaSoft ya cuentan con una gran biblioteca de más de 100 conectores listos para elegir y usar, entre los que podemos encontrar para:

- Base de datos como PostgreSQL, DB2, MySQL, MS SQL Server;
- Crear o eliminar ítems de un calendario de Google;
- Enviar correos a través de un servidor de correos como Gmail o Hotmail;
- Medios de comunicación social como Twitter y Facebook;
- Generación de reportes con la aplicación Jasper;
- Gestionar reglas de negocio con la aplicación Drools, entre muchos más.

Además esta selección está aún más enriquecida por las contribuciones que realizan los usuarios a la comunidad. Bastante a menudo, un usuario puede interactuar con la fuente de datos de la comunidad para la que no exista un conector actual. En lugar de solicitarle al proveedor del software que cree un conector para esa fuente de datos en una versión futura, el usuario tiene la opción de crear un conector y compartirlo con otros. [27]

Existen dos formas de incluir un conector en una actividad:

- Elegir un conector de los que ofrece Bonita OS, seleccionando de una lista que se encuentra separada por categorías.
- Crear un conector propio, para lo cual es necesario indicarle a Bonita datos como, a qué categoría (nueva o existente) va a pertenecer el conector, el nombre que va a tener, la descripción del conector, el icono que va a tener, las entradas y salidas que se va a tener, e implementar el conector utilizando Java como lenguaje de programación.

Un conector es implementado en dos partes:

- Definición: controla las interfaces externas del conector, es decir, las visibles a los usuarios y las visibles al motor de Bonita (las entradas y salidas).
- Implementación. consiste en un archivo XML y una clase Java que codifica su comportamiento.

El separar la definición de la implementación, permite que si el sistema externo cambia la interfaz a la que se está conectando el conector, lo único necesario para enlazar nuevamente el conector al sistema externo es cambiar la implementación del conector, sin tener que modificar los procesos de negocio, lo que ahorra muchos problemas y tiempo de redefinición de diseño.

Una vez creado el conector es posible agregarlo como si se tratase de un conector ya existente. Ambas acciones, agregar un conector como crearlo, se realiza a través de un asistente de configuración que dispone Bonita Studio.

Utilizando conectores, el proceso de negocio ya no tiene que adaptarse al sistema de información, sino que, el sistema de información se debe adaptar al proceso de negocio. Con frecuencia la mayoría de procesos BPM enfrentan este problema de tener que forzar al sistema de información para evolucionar con respecto a los objetivos del proceso. Este es el resultado de la mejora continua de los procesos.

Con los distintos conectores que se disponen y con la facilidad para manejar los distintos módulos del BPMS de Bonita a la hora de diseñar e implementar los procesos, el desarrollar nuevas aplicaciones o adaptar los procesos actuales de una organización son proyectos totalmente realizables. Su adopción como herramienta de desarrollo, sólo dependerá de la implicación que la gente de TI tenga con los objetivos y metas de negocio. [27]

5.1.4 - API REST de Bonita Open Solution

El motor de procesos de Bonita, es decir Bonita Execution Engine (BEE), es el responsable de la ejecución de los procesos. Todas las interfaces de usuario (User Experience, Formularios de las aplicaciones web, aplicaciones web personalizadas y las herramientas de administración) interactúan con el motor de procesos a través de sus interfaces de programación de aplicación (API) para gestionar los procesos de negocio.

REST (REpresentational State Transfer) es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. Este tipo de arquitectura permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que utilice HTTP. Por esto es que REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet.

Sin importar si la aplicación externa utiliza Java o cualquier otro lenguaje de programación, es posible integrarla con Bonita OS utilizando la API REST. Esta API provee acceso a todos los objetos de Bonita (como procesos, tareas, usuarios, conectores, etc.), para ejecutar operaciones sobre ellos (crearlos, consultarlo, actualizarlos o eliminarlos). A partir de esta API, es posible utilizar estas operaciones para crear un workflow con el motor de procesos de Bonita ya que sigue siendo el responsable de ejecutar la lógica del workflow, mientras que los usuarios siguen siendo los que llevan a cabo las tareas y realizan las actividades administrativas del BPMS.

Existen 3 pasos básicos para que una aplicación externa pueda integrarse y operar con Bonita a través de la API REST.

- Autenticación: para acceder a los métodos de las API, es necesario autenticarse con un módulo de acceso (LoginModule) como un usuario registrado en Bonita.
- Ejecutar llamadas REST: una vez que el acceso al motor de procesos de Bonita es exitoso, es posible ejecutar métodos de la API para, por ejemplo, obtener una lista de los procesos desplegados, obtener una lista de las tareas pendientes de un usuario en particular, comenzar una instancia de un proceso, ejecutar tareas, etc.
- Cierre de Sesión: cuando ya se han realizado las tareas deseadas, es necesario cerrar la sesión para evitar errores en el motor de procesos.

Cabe aclarar que para que se pueda utilizar la API REST de Bonita en un ambiente de producción, es necesario contar con un servidor Tomcat o JBoss que fácilmente se puede descargar desde el sitio de Bonitasoft, que contiene al motor de procesos de Bonita junto con uno de estos servidores. [27]

Para este trabajo en particular utilizaremos varios motores de Bonita OS distribuidos con los que nos comunicaremos vía API REST para monitorearlos y usar sus direcciones para ubicar sus registros de eventos.

5.2 - Proceso generador de log de eventos

Como mencionamos anteriormente contamos con un proceso en BonitaOS capaz de generar log de eventos para un determinado proceso en formato XES [18]. A continuación detallaremos su funcionamiento.

El proceso está compuesto por 4 actividades secuenciales “Seleccionar proceso”, “Seleccionar variables de caso”, “Generar log de eventos” y “Mostrar Estado final” como muestra la Figura 5.3. Las dos primeras son actividades manuales, la tercera es una tarea automática que contiene al conector generador del log de eventos y la última simplemente muestra el resultado final del proceso.

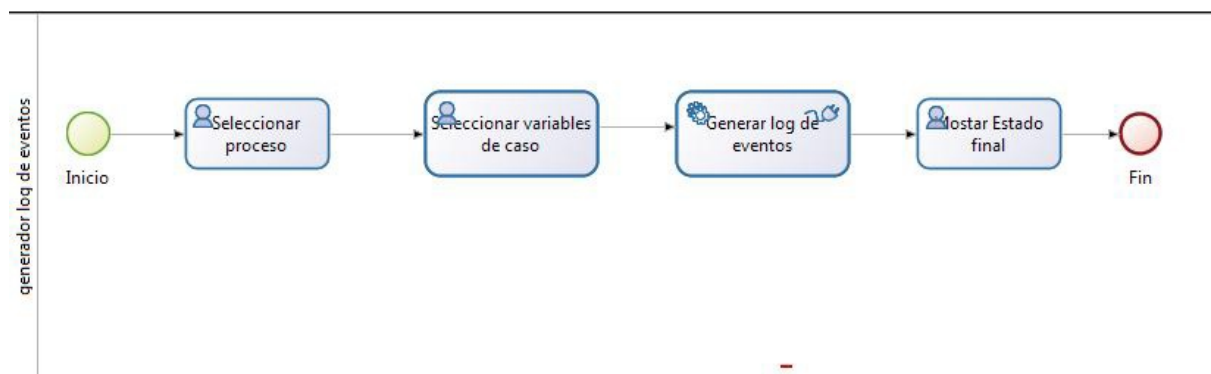


Figura 5.3. Diagrama del proceso generador del log de eventos

En la tarea “Seleccionar proceso” se despliega un formulario donde se despliega un listbox con todas las definiciones de proceso disponibles en el motor de procesos. Se debe seleccionar el proceso del que se desea generar el log de eventos.

En la tarea “Seleccionar variables de caso” se despliega un formulario que muestra en un checkboxlist las variables de proceso disponibles, y el usuario debe seleccionar cuales desea almacenar en el log de eventos. También se debe indicar la ruta de almacenamiento donde se guardará el log de eventos, y el nombre que tendrá el log.

La tercer tarea “Generar log de eventos” es una tarea automática que contiene únicamente un conector que se encarga de generar el log de eventos con las variables seteadas en las tareas previas. El conector insertado en esta tarea es un conector de tipo Script, en particular un conector de script groovy que es código Java. Este conector es un código que se ejecuta al

principio o al inicio de la tarea y mediante el cual se permite agregar funcionalidad a la herramienta. En la implementación el conector se configuró para que se ejecute al inicio de la tarea automática. El script principal invoca un método que crea el documento XML, luego invoca a un método dado un documento XML agrega el elemento raíz y lo devuelve, luego invoca a un método que agrega los atributos del caso y finalmente invoca a un método que escribe el documento XML. [18]

En nuestro enfoque utilizaremos este proceso generador como herramienta para obtener los registros de eventos de los procesos que están distribuidos.

5.3 - Sistema de ejecución y monitoreo de procesos de Bonita OS

A continuación detallaremos las herramientas que nos permiten instanciar y monitorear procesos de BonitaOS de manera remota.

Por un lado contamos con un conector que instancia procesos de manera remota usando la API REST de Bonita para Java [19]. El conector está desarrollado para realizar dos tareas:

- Instanciar procesos en otros motores de procesos de Bonita
- Guardar en una base de datos el identificador de las instancias que han instanciado procesos en otro motor, como así también el identificador de la nueva instancia en el servidor remoto.

Tan solo debemos en el proceso invocar a este conector (definiendo dirección de la API REST destino, credenciales de acceso e identificador de proceso que deseamos instanciar, entre otros) en nuestro proceso local para que se logre la ejecución remota.

Por otro lado, contamos con una aplicación web capaz de, en base a la información que va guardando el conector antes mencionado, monitorear dichos procesos distribuidos (Figura 5.4) [19]. Este sistema contiene 3 componentes fundamentales:

- El motor de procesos de Bonita Open Solution: que se ejecuta sobre un servidor Tomcat y está situado en cada uno de los nodos de la arquitectura. Es el encargado de la ejecución normal de los procesos de negocio y la ejecución del conector de instanciación remota.
- La aplicación de monitoreo distribuido: desarrollada en PHP es la encargada de recolectar la información de los subprocesos situados en los diferentes nodos para unificar el proceso original particionado y mostrar el proceso como si se tratase de uno solo. Esta se comunica con las aplicaciones de monitoreo que residen en los otros nodos a través del uso de servicios web (WebServices), donde las aplicaciones de estos nodos serán las encargadas de recolectar la información de los subprocesos que son ejecutados en el motor de procesos que residen en ese sitio y que conocen si el subproceso contiene alguna referencia a otro nodo.
- Un servidor de base de datos (MySQL): que almacena la información de las instancias de los procesos de negocio. Guarda por cada instanciación remota: el identificador de

la instancia local que realizó la ejecución del conector, la localización del servidor remoto y el identificador de la instancia en el motor remoto que se ha creado .

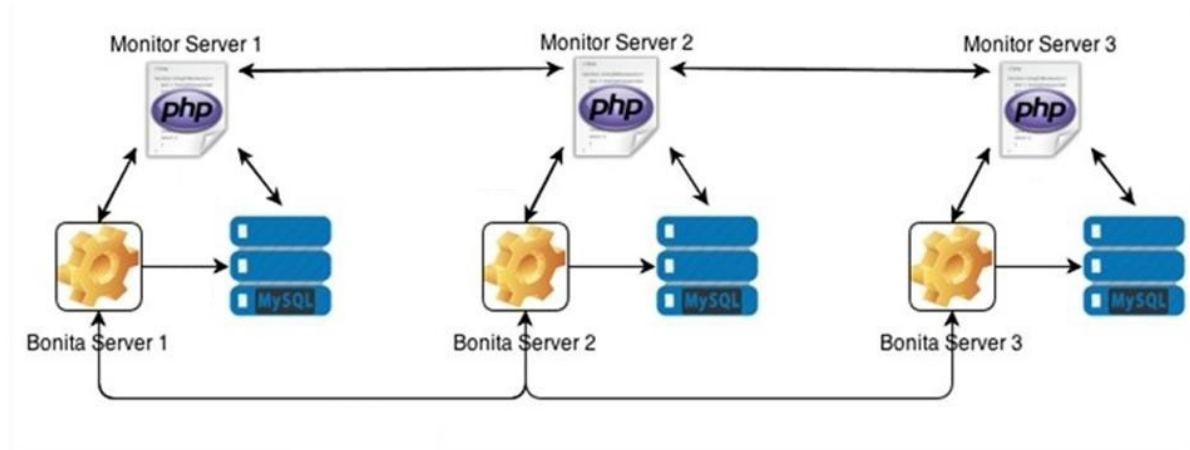


Figura 5.4. Arquitectura del Sistema de Ejecución y Monitoreo [19]

En este caso utilizaremos la aplicación de monitoreo para extender su funcionalidad para permitir obtener los logs de los procesos que monitorea. Además, utilizaremos la información que va dejando es la base de datos que nos permitirá realizar las conexiones correspondientes que se explicarán en detalle en el próximo capítulo.

5.4 - ProM

ProM es un framework open source genérico para implementar técnicas de Process Mining en un ambiente estándar. El framework ProM recibe como entrada logs de eventos en formato XES (eXtensible Event Stream) o MXML (mining eXtensible Markup Language). Actualmente el framework tiene plug-ins para Process Mining, análisis, monitoreo y conversión.

ProM está disponible en archivos de distribución binaria para plataformas Windows, Mac, y Unix. La última versión de ProM tiene licencia GPL.

Los productos de Business Intelligence se centran en los datos y son limitados cuando se requiere formas más avanzadas de análisis. Las herramientas de data mining ofrecen más "inteligencia", pero también se centran en los datos. Los sistemas que se centran en el proceso como sistemas BPM, herramientas de simulación, y herramientas de modelado, se focalizan más en Play-out. Las plataformas de software tradicional no permitían la utilización de técnicas de Process Mining. Esto disparó el desarrollo de varias herramientas de Process Mining que se dedicaban a una técnica especial de Process Mining. Claramente no tenía sentido dedicar una herramienta de Process Mining para cada técnica de descubrimiento de proceso nueva. Esta observación provocó el desarrollo de Prom framework, un entorno "plug-able" (conectable) para Process Mining que utilizaba MXML como formato de entrada.

A lo largo de las distintas versiones de ProM, más plugins para soportar distintas técnicas de data mining fueron incorporados. De esta forma ProM se convirtió el estándar de facto para Process Mining. Grupos de investigación de todo el mundo contribuyen en el desarrollo de ProM y cientos de organizaciones han descargado ProM.

La última versión de ProM (ProM 6) está basada en XES más que en MXML. XES es el nuevo estándar de Process Mining adoptado por la IEEE Task Force for Process Mining.

Para generar los logs en formato XES, ProM cuenta con una aplicación llamada XESame que permite la extracción de un log de evento de fuentes de datos. El principal motivo de selección de la herramienta ProM es su carácter open source, gratuito, sumado a la cantidad de documentación disponible gracias a los distintos aportes realizados por distintos grupos de investigadores de todo el mundo. [5][29]

Esta herramienta nos servirá para mostrar la aplicación de técnicas de minería de procesos a los registros de los procesos distribuidos que hayamos obtenido previamente.

Hasta aquí hemos detallado las herramientas y el escenario con el que contamos hasta el momento. En el próximo capítulo introduciremos la solución propuesta al problema de aplicar técnicas de minería de procesos distribuidos en BonitaOS y luego un caso de estudio que valide dicha solución.

Capítulo 6 - Solución propuesta

Repasemos con que contamos hasta el momento en materia de minería y procesos distribuidos antes de presentar la solución propuesta:

- Tenemos motores de BonitaOS en diferentes servidores distribuidos a través de la red con los cuales podemos interactuar mediante su API REST. Ellos son los encargados de ejecutar los procesos que en ellos están desplegados, con la particularidad de que mediante su interfaz de protocolo HTTP, podemos instanciar procesos y obtener información de sus estados de manera remota.
- Contamos con una aplicación web PHP que mediante web Services recolecta información de dichos procesos distribuidos y los presenta de forma tal que permite realizar un monitoreo activo de los procesos y sus instancias.
Esto es posible ya que los procesos ejecutan un conector personalizado que realiza básicamente dos cosas:
 - Instanciar un proceso remoto mediante la invocación a la API REST de Bonita desde Java. Se configura al proceso “local” para que en el momento que quiere continuar la ejecución con la creación de una instancia remota, en ese lugar de su flujo ejecute este conector. Esto es lo que permite la ejecución distribuida.
 - Guardar la información de dicha operación en una base de datos local (MySQL). Para luego poder realizar el monitoreo de las instancias, se recolecta, entre otras cosas, la dirección de la API REST asociada al motor remoto y los identificadores de instancia tanto local como remoto involucrados en las ejecuciones.
- A su vez, contamos con un proceso generador de log de eventos que al ejecutarlo nos devuelve el registro de los eventos ocurridos para un determinado proceso en el formato estándar XES. Este es un formato soportado por la herramienta ProM que utilizaremos para aplicar las técnicas de minería de procesos. Este proceso nos va asistiendo en la creación de dicho archivo. Nos permite elegir para un determinado proceso, que variables queremos incluir en el resultado final. Esto lo logra con la ejecución de un conector personalizado que mediante script Groovy, va creando este archivo XES (que es una forma particular de XML) en base a nuestras preferencias y usando como interfaz de acceso a los datos del proceso a la API de Bonita.

Dado este escenario se quiere extender la aplicación de monitoreo para que provea la funcionalidad de obtener el registro de eventos de un proceso distribuido. En este contexto se pueden observar dos posibles enfoques:

- Análisis por partes: donde se obtendrían los registros de cada parte del proceso distribuido por separado (en archivos independientes) y cada uno de ellos sería una entrada diferente en la herramienta de minería de procesos,

- Análisis en conjunto: en el cual, mediante el procesamiento de cada registro, se produciría un solo archivo de eventos que representa el comportamiento del proceso completo, siendo este la única entrada a la que se le aplicaría la minería.

La primer alternativa brinda un análisis de alcance a nivel de servidor, es decir, sacar conclusiones sobre el rendimiento de esa parte del proceso en ese servidor específico en el que se encuentra (por ejemplo, si una tarea tarda mucho más de lo esperado en completarse puede deberse a que las prestaciones que da dicho servidor no son suficientes y por lo tanto considerar el aumento de sus capacidades o mover esa tarea a otro servidor capaz de mejorar dicho rendimiento). También da lugar a evaluar aspectos del proceso como por ejemplo si la forma en que se decidió descomponerlo es la óptima para ese caso en particular o pueden hacerse algunos reajustes. Su dinámica se muestra en la Figura 6.1

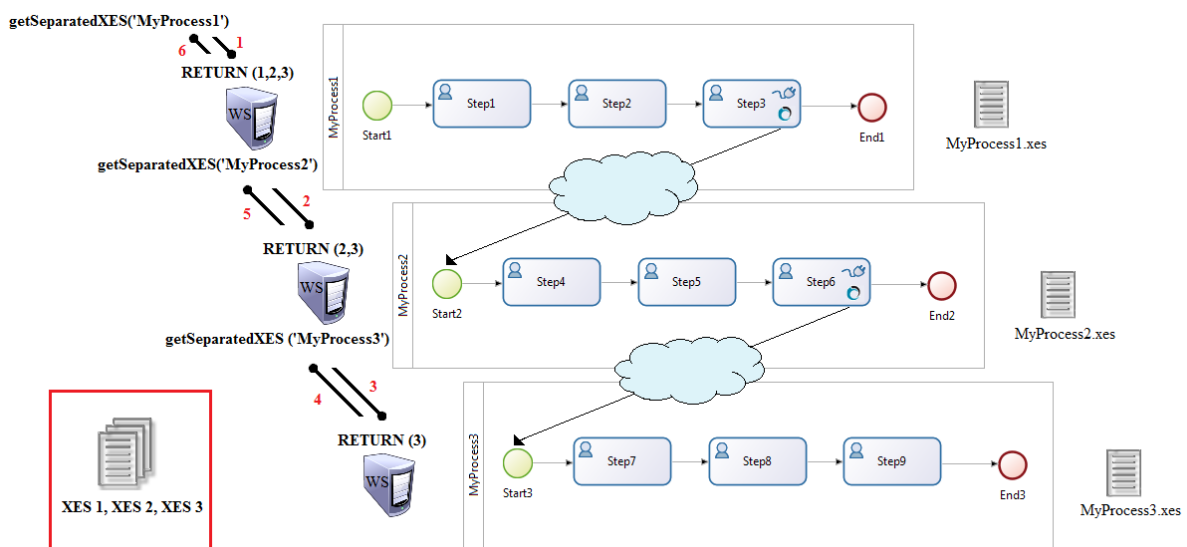


Figura 6.1. Cadena de llamadas de web service para obtener registros separados.

El segundo análisis, representado por la Figura 6.2, es el más tradicional, dado que su resultado es el equivalente al del proceso unificado. En este caso se pueden observar aspectos tales como la detección de posibles cuellos de botella, rendimiento total, predicciones, etc.

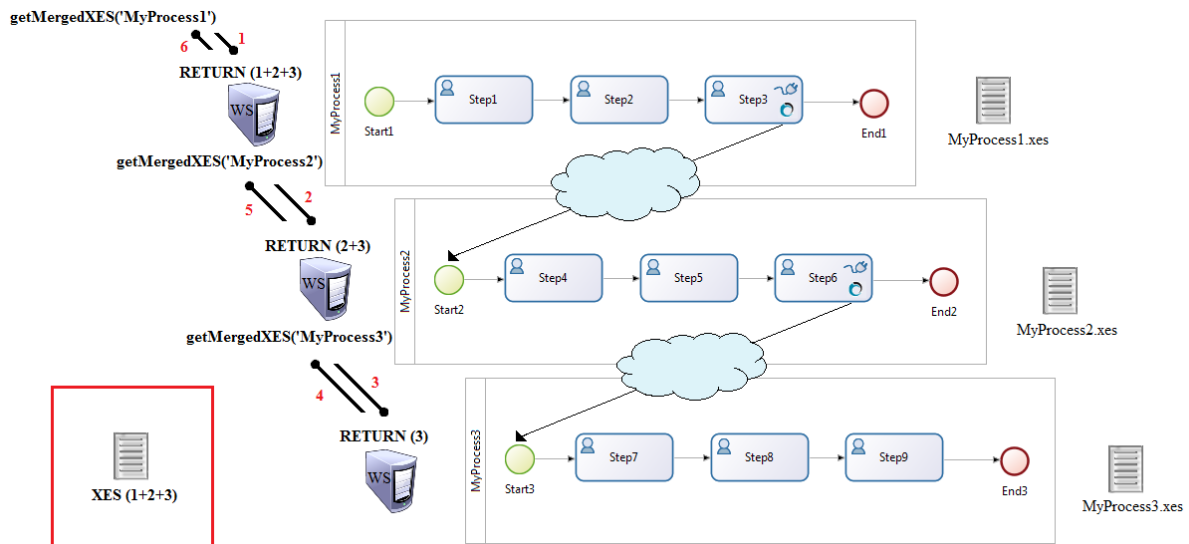


Figura 6.2. Cadena de llamadas de web service para obtener un registro unificado.

Esta nueva funcionalidad se presenta como un servicio web. Utilizando el kit de herramientas NuSOAP [30] para desarrollar Web Services bajo PHP se ponen a disposición dos servicios: uno para obtener los registros de cada una de las partes en que está dividido el proceso y otro para obtener el registro del proceso unificado. En el primer caso basta con ir por cada servidor en donde se encuentra cada parte y recuperar el registro que genera el conector antes mencionado. En el otro caso, como XES tiene una estructura de XML, se utiliza la clase SimpleXML [31] de PHP para el procesamiento de los registros existentes y generación del log unificado.

Antes de ver en profundidad de qué manera se implementaron estos servicios, señalemos que precondiciones damos por aceptadas para que la ejecución de éstos sea satisfactoria. Primero damos por sentado que todas las instancias de un proceso se encuentran en un mismo servidor. Esto es importante ya que cuando obtenemos el log de eventos tiene que reflejar la actividad de todas las instancias. Si el mismo proceso estuviese implantando en varios servidores no tendríamos la visión completa de su registro. Otro punto es que previamente se tiene que haber ejecutado el proceso generador de logs para todos los procesos de los cuales queremos obtener información, de manera que estén disponibles para que el servicio web pueda hacer uso de ellos. Las últimas condiciones previas son más del orden de la configuración ya que necesitaremos que los registros se guarden en ciertas rutas específicas: C:/log/BonitaLogsXes/ en Windows o /var/log/BonitaLogsXes/ en Linux y que sus nombres cumplan la siguiente regla: {nombre de proceso} + "--" + {versión} + ".xes"

Ahora sí veamos de qué forma se implementaron estos servicios web para entender cómo los llama la aplicación de monitoreo y cómo integran la información que van dejando tanto por este último, como por el procesos generador de log de eventos.

Comenzamos por crear una pestaña nueva en la aplicación de monitoreo que sirva de interfaz para obtener los logs de los procesos distribuidos en cuestión.

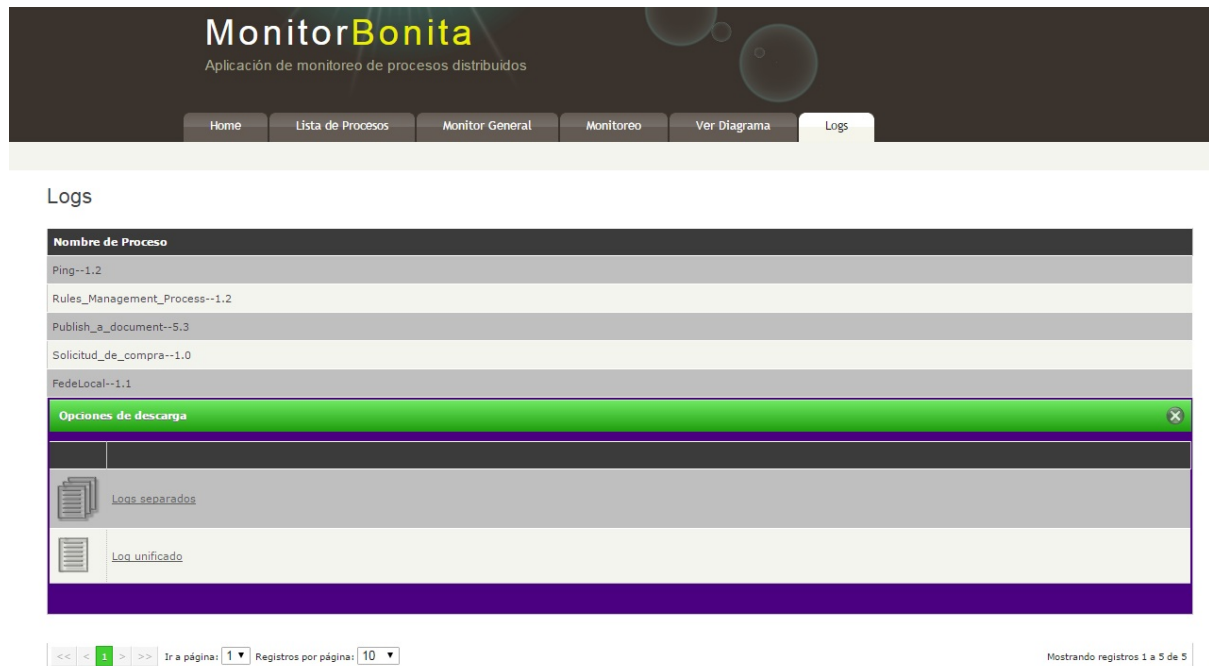


Figura 6.3. Menú para obtener registros desde la aplicación de monitoreo

Como se observa en la Figura 6.3, la aplicación nos trae la lista de los procesos que al menos han ejecutado una vez el conector que instancia procesos de manera remota. Recordamos que esto lo podemos hacer ya que el conector desarrollado [19] mantiene una base de datos la cual registra, por cada instanciación remota, que dos procesos están involucrados (local y remoto), sus correspondientes identificadores de instancias y la dirección de su servidor REST (entre otros).

Ahora por cada entrada le damos la opción de obtener el log de eventos de cualquiera de las dos maneras antes mencionadas: separado en un archivo por proceso o unificado en uno solo. Esto lo logramos mediante la llamada a dos servicios web que reciben sólo por parámetro el nombre del proceso. A continuación mostramos en la Figura 6.4 la secuencialidad de eventos y luego describiremos el comportamiento del método que devuelve los registros separados.

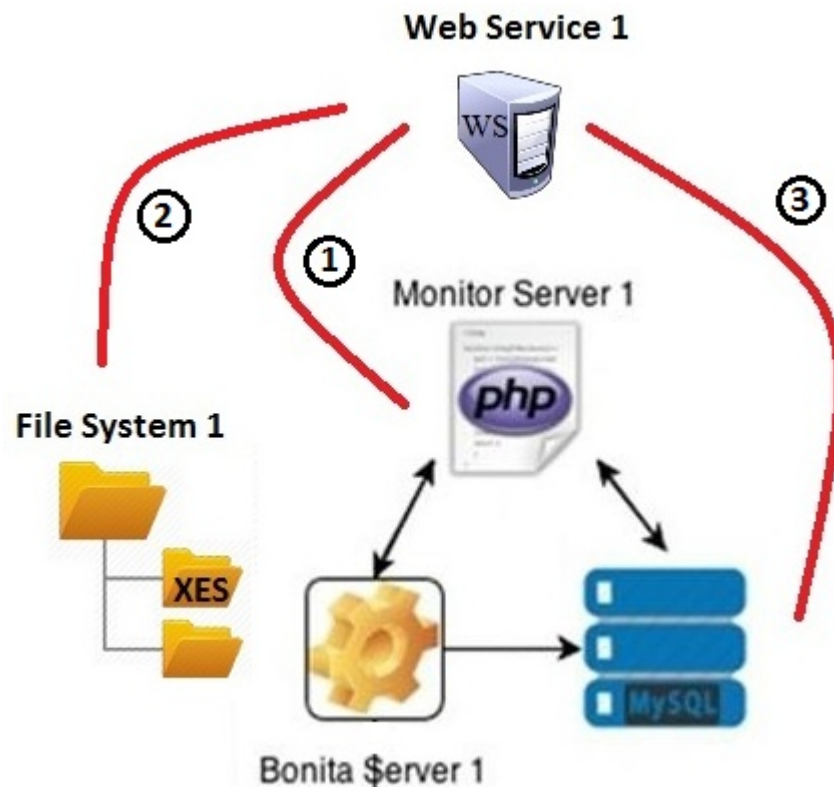


Figura 6.4. Dinámica de llamados para obtener registro de eventos en un servidor

Lo primero que hace el método es, luego de incluir la librería NuSOAP, es crear un cliente para poder invocar los métodos que define el web service. Una vez establecida la conexión entre cliente y servidor, hace un llamado al método específico para obtener los registros de eventos (pasándole como parámetro el identificador del proceso en cuestión). Esto lo podemos observar en 1 en la Figura 6.4.

Una vez en el web service, lo primero que realiza este método es determinar sobre qué sistema operativo está corriendo el servidor (Windows o Linux) para decidir a qué dirección local del árbol de directorios ir a buscar el archivo de registro. Una vez resuelto esto, obtiene el contenido de dicho archivo XES (recordemos que es un XML) y lo inserta en un arreglo que será el que devuelva como resultado. Esta secuencia se puede observar en el apartado número 2 de la Figura 6.4. Se estableció una regla interna que lo primero que se inserta en el arreglo siempre es el nombre del proceso, luego un caracter de corte de control, en este caso “|” (tubería) y a continuación el contenido plano de dicho archivo de registro. Luego lo que hacemos es consultar en la base de datos si hay o no al menos una referencia a una instancia remota de la instancia local. Se representa en la Figura 6.4 como 3. Esto va a ser siempre verdadero en el primer llamado al método, ya que, como mencionamos anteriormente, solo habrá registros en la base de datos si se registró una ejecución remota. Sin embargo esto se pregunta ya que, luego del primer “enganche”, se puede dar una situación recursiva en la que el proceso remoto en esta instancia, a su vez instancie otro proceso de manera remota, y así sucesivamente. Cuando sabemos que la ejecución continuó en otro servidor, debemos ir a buscar el registro de ese proceso para empaquetarlo y retornarlo junto con el actual. Esto lo conseguimos mediante el mismo web service que está corriendo en dicho servidor también.

Obtenemos de la base de datos su IP y el identificador de proceso destino. Con estos datos nos es suficiente para armar una nueva petición (del mismo método) al web service para que nos retorne el log de eventos del proceso remoto. En la Figura 6.5 se ilustra cómo se da esta comunicación entre los Web Services en los distintos servidores.

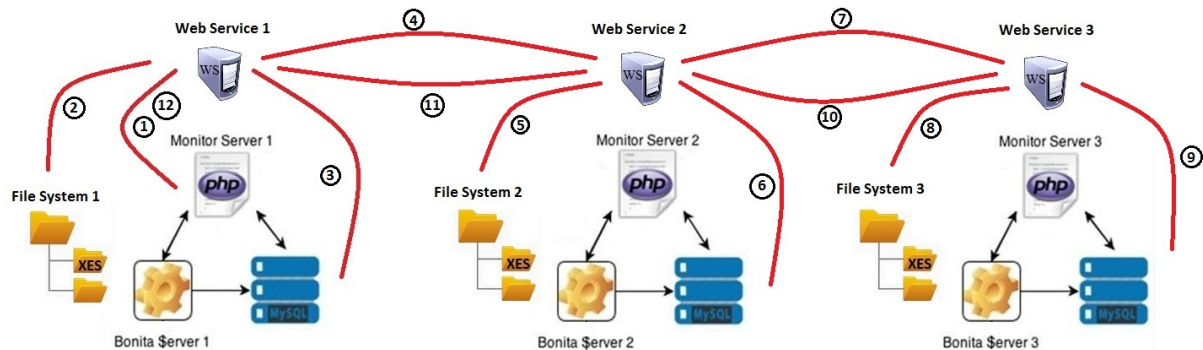


Figura 6.5. Dinámica de llamados para obtener registro de eventos en varios servidores

Como vimos en el identificador 3, la base de datos le dice que hay un proceso remoto que está relacionado con el proceso local en el servidor 1, por lo que se comunica con el Web Service 2 para pedirle el registro separado de dicho proceso (esto se muestra en el identificador 4). El Web Service 2 recupera el registro de este proceso desde su sistema de archivos local y lo adjunta a la respuesta, como observamos en 5. No sin antes chequear que la cadena de llamados remotos termine aquí. Como se señala en 6, chequea en la base de datos y nota que para este proceso local hubo una correspondiente ejecución remota. Por lo tanto, antes de retornar el registro del proceso local que ya encoló para responder, arma una nueva petición del mismo método pero esta vez al Web Service 3, representado en 7. Ya en el Web Service 3, como podemos notar, termina la cadena de llamados distribuidos. Lo primero que hace es obtener el archivo de registros para ese proceso local del servidor (8) y, como no registra en la base de datos una ejecución remota en otro servidor para dicho proceso (9), simplemente devuelve el proceso antes encolado (10). Recordemos que el Web Service 2 ya tenía en su arreglo de respuesta su registro local, entonces lo que hace es agregar a éste lo que recibió del Web Service 3 como respuesta y de manera recursiva retornar esto al Web Service 1, como se ve en 11. Ya en esta instancia, de vuelta en el Web Service 1, como el método es el mismo, toma la respuesta que recibió y la une a su registro local previamente reservado.

Siguiendo esta lógica retorna el resultado (un arreglo con los 3 registros de eventos separados), pero solo que esta vez al sistema de monitoreo, que instancia un objeto de la clase ZipArchive para crear un archivo comprimido que estará compuesto por los archivos de registro separados. Lo que hace es recorrer la colección de registros que fue devuelta por el servicio web y por cada elemento crear un archivo con el nombre del proceso (que viene primero hasta el carácter de corte "|") con lo que le sigue como su contenido (el XML). Finalmente cierra el archivo comprimido y lo devuelve vía descarga directa al navegador.

Antes de pasar a la otra opción, veamos que forma tiene el formato XES ya que nos va a servir para entender una decisión que tomaremos más adelante con respecto al armado de un

log unificado. Como sabemos XES es un estándar basado en XML para logs. Como tal podemos definir la siguiente estructura de etiquetas principales:

```
<log> // Proceso
  <trace> // Instancia
    <event>...</event> // Tarea
    <event>...</event>
  </trace>
  <trace>
    <event>...</event>
    <event>...</event>
  </trace>
</log>
```

Vemos que cuenta a gran escala con 3 tipos de etiquetas:

- log: aquí va toda la información relativa al proceso, como su nombre, versión, etc.
- trace: donde se definen los datos propios de una instancia, como identificador, valor de variables de instancia, etc.
- event: etiqueta que delimita cada una de las tareas ejecutadas para la instancia definida en el nivel superior, junto a algún valor de variables locales a la tarea (si es que contiene), actor involucrado, fecha y hora de inicio y finalización, etc.

Ahora sí veamos cómo se hace para obtener esa misma información pero en un solo archivo que será la única entrada para la aplicación de minería.

En lo que respecta a la preparación antes de llamar al servicio web, se comporta de forma similar a su par descrito anteriormente, incluyendo la librería NuSOAP y llamando esta vez a otro método del web service con el identificador de proceso como parámetro. Luego determina sobre qué sistema operativo está para determinar la ruta del archivo de registro local a tratar y agrega su contenido al arreglo que devolverá al final. Una vez hecho esto consulta en la base de datos si hay referencias remotas, en caso de ser así aquí se produce el tratamiento que difiere del método anterior.

Como vamos a trabajar con solo un archivo de retorno tenemos que ir “insertando” los registros de eventos pero no en cualquier lugar del archivo XES. Aquí hacemos uso de la tabla relacional que nos deja la base de datos que además nos indica qué instancia específica de un proceso local se corresponde con una instancia específica de otro proceso remoto. Mediante esta relación y la librería de manejo de XML SimpleXML, vamos buscando en ambos archivos (recordemos que todos los XES tienen a nivel de la etiqueta <trace> su identificador de instancia) y cuando encontramos dicha relación insertamos dentro de la etiqueta <trace> de la instancia local el contenido correspondiente de su par remoto (todas las etiquetas <event>). El resultado será que el proceso local tendrá dentro de su lista de tareas, aparte de las propias, las correspondientes a su proceso remoto. Este procedimiento lo realizamos por cada tupla que tengamos en la base que relaciona instancias locales y remotas del proceso en cuestión.

Al igual que el ejemplo anterior, en el primer llamado entre servidores de la cadena, siempre va a dar verdadero que hay relación local-remoto, pero también así se cubre la situación que el remoto en esta fase sea el local en la próxima y tenga su par remoto a quien pedirle su información. Así se crea una capa transparente en la cual si tenemos una cadena del tipo A-->B-->C, cuando A le pida a B, éste antes de retornarle a A tendrá que hacer lo suyo con C y recién ahí darle su información. Podríamos notar entonces que C se retornaría solo ya que es el fin de la cadena, B recibiría a C y lo incorpora a su estructura en base a la información de instancias de su base de datos local y generaría una versión ampliada de sí misma que podríamos denominar B+C, que es lo que retornaría a A y éste haría lo mismo creando el log completo A+B+C. Podemos señalar que en este caso tenemos el panorama completo de la ejecución de un proceso distribuido, a diferencia de la opción anterior donde tenemos un archivo y por lo tanto una visión por cada sub-proceso en el que el proceso completo fue particionado.

Una vez concluido el código del web service retorna su resultado al controlador de la aplicación de monitoreo (al igual que la opción anterior) y lo renombra con la extensión “.xes” para que sea descargado por el navegador listo para su posterior análisis en la herramienta de minería.

Ahora que ya vimos en profundidad cómo funcionan éstos web services, detallaremos un caso de estudio que muestre para un proceso distribuido, la obtención de su registro de eventos y su posterior análisis con ProM.

Capítulo 7 - Validación con un caso de estudio

En este capítulo mostraremos un ejemplo práctico para mostrar las herramientas y funcionalidades que materializan los conceptos planteados previamente. Entre ellas podemos enumerar las siguientes:

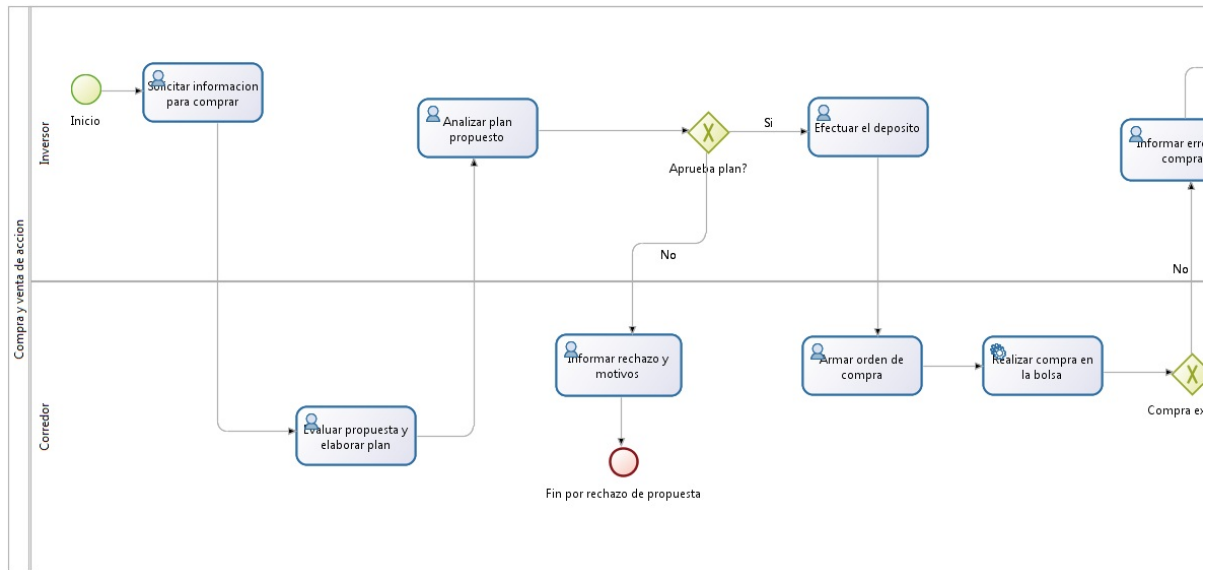
- Motores de procesos Bonita OS distribuidos que son instanciados por un conector mediante su API REST (que resuelve el tópico de ejecución de procesos distribuidos).
- Conector que permite obtener el registro de eventos para un proceso local en formato XES (referente a la información necesaria para poder aplicar técnicas de minería de procesos).
- Aplicación web que permite el monitoreo y, ahora, agrega la funcionalidad de obtener el registro de eventos para procesos distribuidos (funcionalidad que permitirá el posterior análisis de minería de procesos en entornos distribuidos).
- ProM, un framework open source genérico para implementar técnicas de Process Mining (herramienta que nos permitiera realizar minería de procesos del log de eventos de un proceso distribuido).

Se mostrará un ejemplo de proceso que simula la compra y venta de una acción en el mercado de valores. Identificamos dos grupos de actores:

- Inversor: es quien inicia el proceso y quien va a comprar y luego vender la acción elegida
- Corredor: es quien recomienda una inversión concreta basada en las peticiones del inversor

Comprende todo lo relativo a la solicitud del inversor, sugerencia del corredor y si esta es aceptada, su compra y posterior venta cuando sea el inversor lo considere rentable.

Veamos el diagrama del proceso completo (Figura 7.1) para explicar cada una de sus tareas. Luego mostraremos cómo serán sus divisiones en subprocesos distribuidos y los motivos que nos llevaron a dicha división.



continuación...

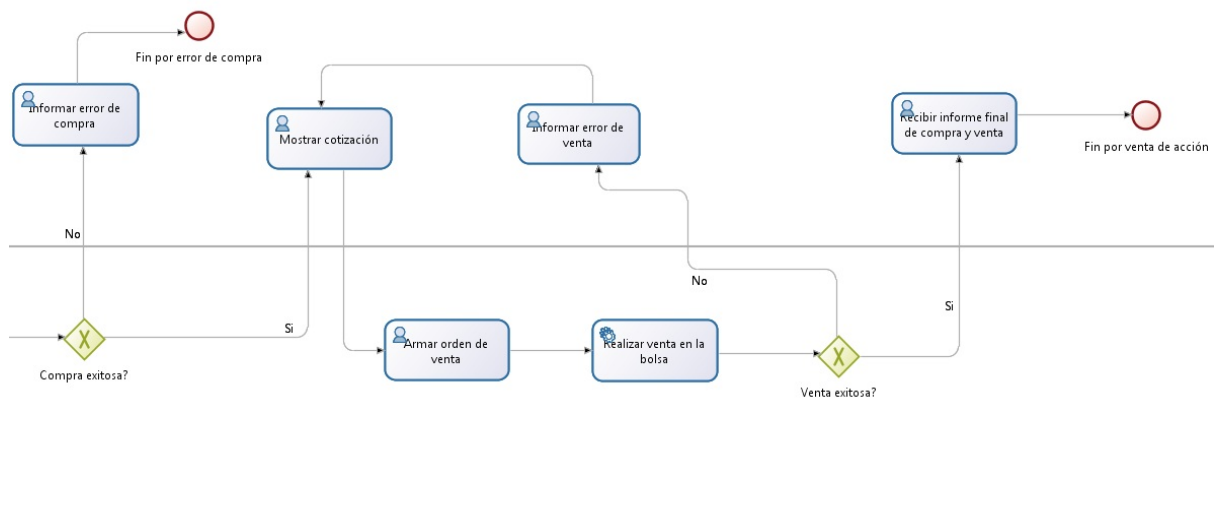
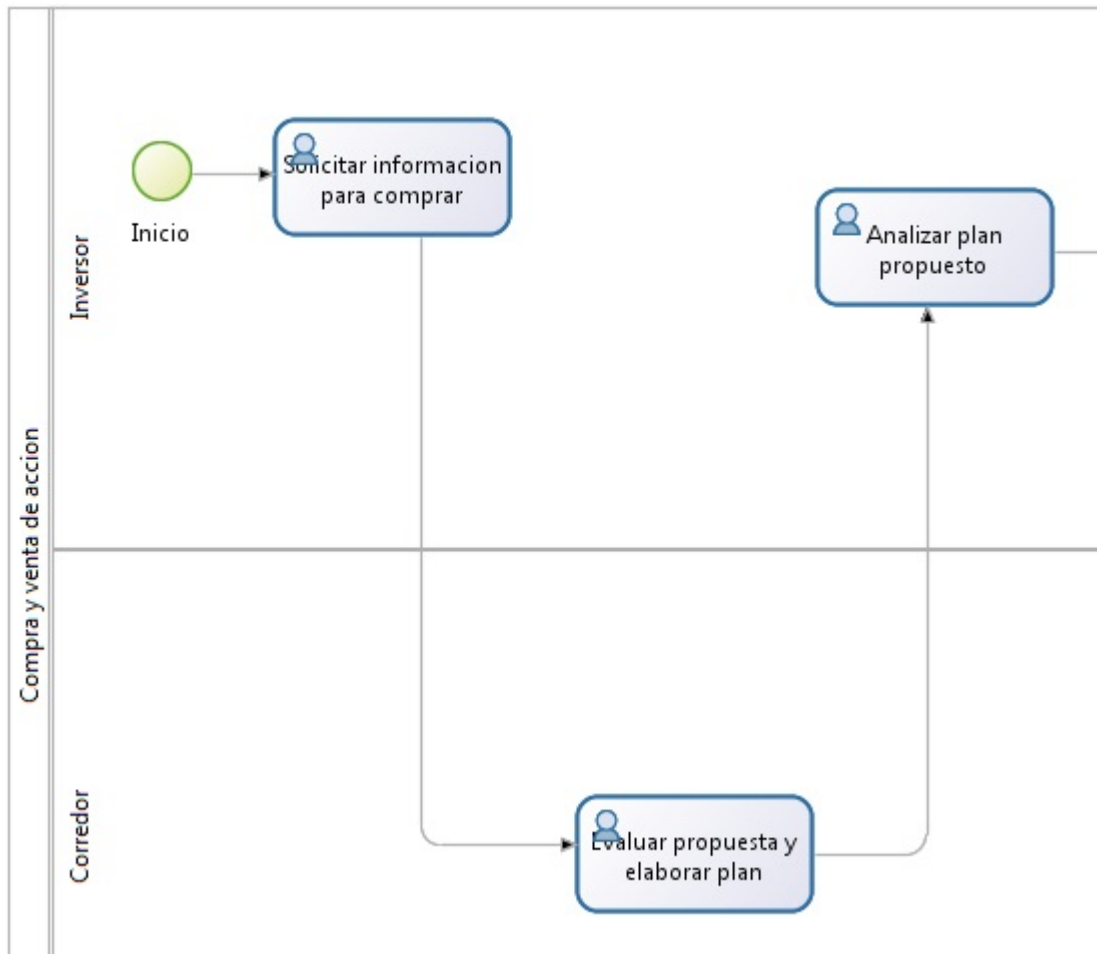


Figura 7.1. Diagrama de proceso "Compra y venta de acción".

Simulemos una ejecución para ver de qué trata cada tarea. Empecemos por las primeras tres tareas "Solicitar información para comprar", "Evaluar propuesta y elaborar plan" y "Analizar plan propuesto"



Comenzamos con el inversor completando la solicitud donde especifica que capital tiene pensado invertir y las características de esta inversión (plazos y riesgos estimados). Luego un campo de comentarios, si lo considera necesario, para orientar un poco más al corredor que la procesará.

Solicitar informacion para comprar

Capital a invertir (\$)

Plazo

Riesgo

Comentarios

Created with Bonita Open Solution

Una vez completada esta tarea, el corredor pasa a evaluar la solicitud. Se le muestran los datos que ingresó el inversor y luego un link que muestra las cotizaciones de las acciones para que elija la que considere que cumple con lo que pidió el inversor. Una vez elegida, la selecciona de la lista que aparece más abajo, junto con el precio actual y la cantidad que recomienda comprar. Una vez completado esto, presiona en el botón "Calcular" que calcula el total de la inversión sugerida y agrega algún comentario que considere necesario para el inversor.

Evaluar propuesta y elaborar plan

Capital a invertir 15000

Plazo Mediano (Meses)

Riesgo Bajo

Comentarios del inversor Quisiera tener un rendimiento mejor que el de un plazo fijo

[Ver acciones -->](#)

Elegir acción YPF S.A. (YPFD)

Precio actual (\$) 239.35

Cantidad de acciones 60

Calcular

Total a invertir (\$) 14361

Comentarios Esta es una empresa estable y suele dar interés anual de 40%

Enviar

Created with Bonita Open Solution

Luego, el control vuelve al inversor que evalúa la propuesta hecha por el corredor. Aparte, se le muestra mediante un iframe, la cotización del momento de la acción que se le recomendó. Mediante un checkbox determina si aprueba o no (en este último caso tiene la posibilidad de dejar un comentario que justifique su rechazo)

Analizar plan propuesto

[Iniciar Sesión](#)



0810-666-4717

PUENTE



Home / Cotizaciones

Ypf S.A. (YPFD)

 Exportar precios históricos Opciones Calculadora de lanzamiento cubierto Valuación de Opciones Lanzamiento Cubierto Acciones

\$ 240,00

 + 0,1000 (0,03%)

Fecha/Hora: 12:59

Volumen: 7.061.264

Mínimo: 238,90

Últ. cierre: 239,50

Máximo: 240,50

Apertura 239,00

239,35

Vender

239,00

Comprar

[Ver más -->](#)

Acción YPF S.A. (YPFD)

Precio actual (\$) 239.35

Cantidad de acciones 60

Total a invertir (\$) 14361

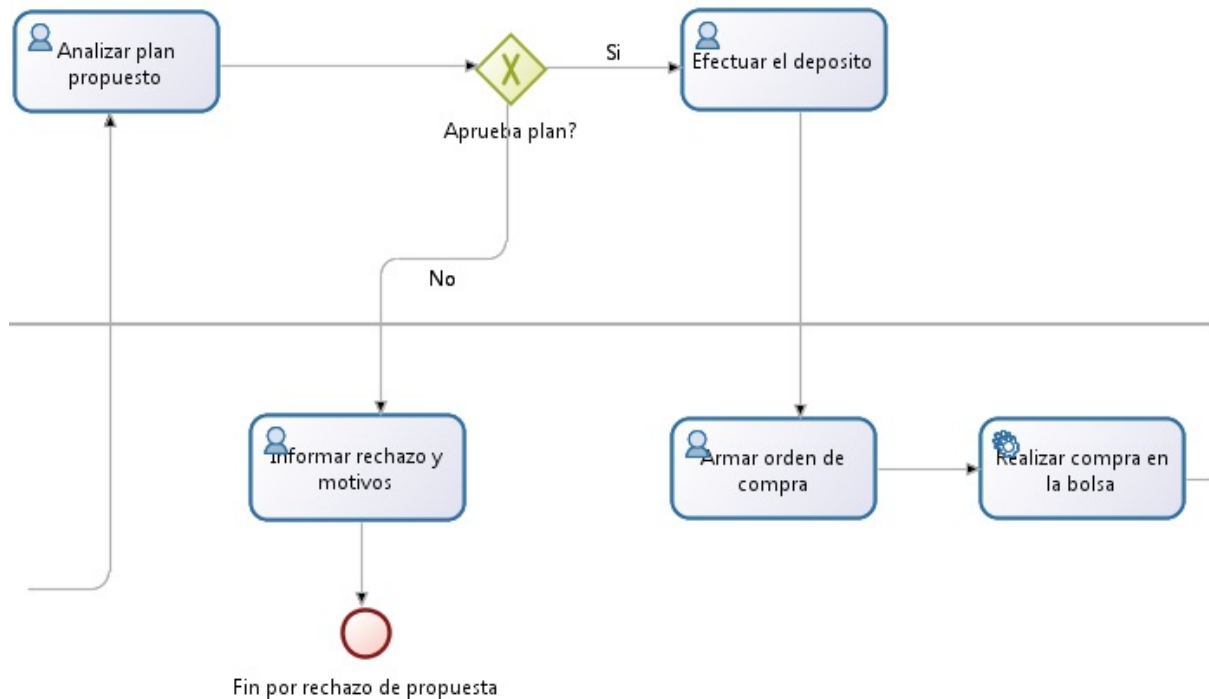
Comentarios del corredor Esta es una empresa estable y suele dar interés anual de 40%

Aprueba el plan?


Comentarios En caso de no aprobar, detalle cuales fueron los motivos

Enviar

Pasemos a la siguiente sección del proceso donde se decide si aprobar o no la solicitud. En caso de aprobar, se arma la orden de compra y se efectúa dicha transacción.



Si no aprueba el corredor recibe la notificación junto con el mensaje que dejó previamente el inversor. Al completar esta tarea el proceso finaliza.


| Salir

Compra y venta de accion

Informar rechazo y motivos

Acción

Precio actual (\$)

Cantidad de acciones

Total a invertir (\$)

El inversor rechazó el plan propuesto por los siguientes motivos:

Comentarios

Created with Bonita Open Solution

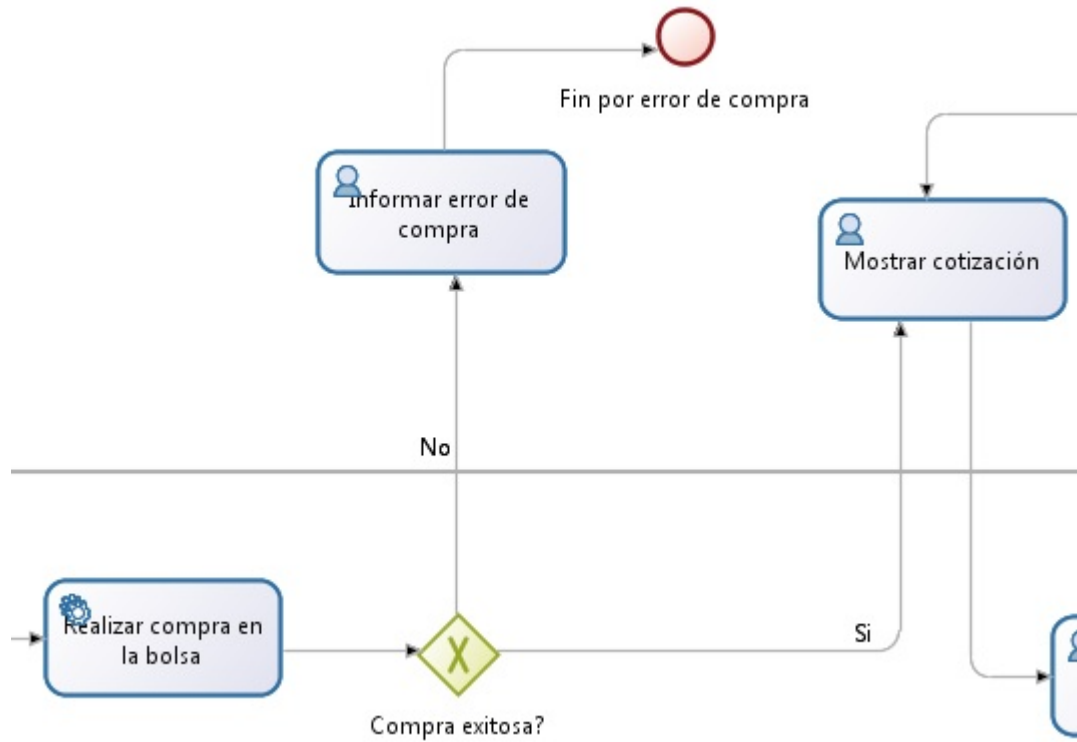
En caso de aprobar el control se mantiene en el inversor, que ingresará el CBU de la cuenta de donde serán extraídos los fondos para comprar las acciones pautadas.

En este punto, con ya todo acordado, el corredor arma y confirma la orden de compra. Esta orden es enviada al sistema de la bolsa de valores que procesa dicha orden (simulada en este caso por un script groovy) que devuelve si aprueba o no. Esto lo decide si la operación se está realizando durante algún día hábil y en horario bancario (de 10 a 15hs).



The screenshot shows the 'bonitaopen solution' logo on the left and the text 'Compra y venta de accion' on the right. Below the logo, the title 'Armar orden de compra' is centered. The form contains three input fields: 'Acción' with a dropdown menu showing 'YPF S.A. (YFPD)', 'Monto (\$)' with the value '14361', and 'CBU' with the value '28505909 40090418135201'. A 'Comprar' button is located below the CBU field. At the bottom of the interface, a red banner contains the text 'Created with Bonita Open Solution'.

Aquí nos encontramos con otra compuerta condicional que, en base a la respuesta de la bolsa, toma diferentes caminos. Si rechaza la compra por estar fuera de día u horario, se lo informa al inversor y da por finalizado el proceso.



Si aprueba, se le muestra al inversor la cotización mediante un iframe. Esta tarea quedará en espera hasta que éste decida vender sus acciones. Dicho iframe se actualiza cada vez que entra a la tarea, por lo que puede consultarlo diariamente que reflejará siempre la cotización actual. Aparte de mostrarle a qué precio compró, se le da la posibilidad de calcular cuál sería su saldo final e interés obtenido si vendiese con la cotización actual. Una vez que decida vender, completa la tarea que notifica al corredor para que arme la correspondiente orden de venta.

Mostrar cotización

[Iniciar Sesión](#)

0810-666-4717

PUENTE



Home / Cotizaciones

Ypf S.A. (YPFD)

[Exportar precios históricos](#)[Opciones](#)[Calculadora de lanzamiento cubierto](#)[Valuación de Opciones](#)[Lanzamiento Cubierto](#)[Acciones](#)

\$ 335,00

 + 1,1000 (0,33%)

Fecha/Hora: 13:06

Volumen: 7.707.514

Mínimo: 333,50

Últ. cierre: 333,90

Máximo: 339,50

Apertura 335,05

335,00

Vender

334,00

Comprar

Acción YPF S.A. (YPFD)

Precio de compra (\$) 239.35

Cantidad de acciones compradas 60

Total invertido (\$) 14361

Precio actual (\$) 334

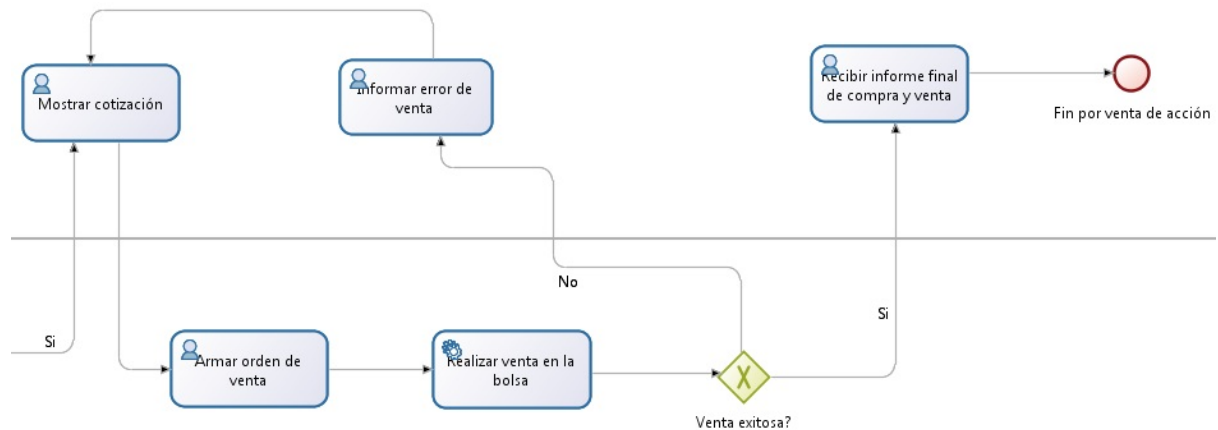
Calcular

Saldo final (\$) 20040

Interés (%) 39.54

Vender

Ya nos encontramos en la parte final del proceso, donde se vende la acción.



En este punto, al igual que cuando se le dio la orden para comprar, el corredor arma la orden de venta para las acciones para luego enviársela al sistema de la bolsa de valores.

bonitaopen solution | Salir
Compra y venta de accion

Armar orden de venta

Acción YPF S.A. (YPFD)

Monto (\$) 20040

CBU 28505909 40090418135201

Vender

Created with Bonita Open Solution

Al igual que en momento de la compra, el sistema de la bolsa devolverá si pudo o no realizar la operación (por las restricciones de días y horarios mencionadas anteriormente). En base a esta respuesta se procederá de distinta forma.

Si no se pudo concretar la venta, el sistema informará al inversor y volverá a la tarea de mostrar cotización, esperando que éste vuelva a dar la orden de vender. Aquí a diferencia de la operación de compra, no finaliza el proceso sino que se forma un bucle.

En caso que se pueda concretar la venta, se le informa al inversor y se le muestra un resumen de todo el proceso de compra y venta, con las cotizaciones y montos finales de ambas operaciones junto con el interés obtenido. Luego, finaliza el proceso.

Recibir informe final de compra

Detalle final de la compra y venta

Acción

Compra:

Precio de compra (\$) Cantidad de acciones compradas Total invertido (\$)

Venta:

Precio de venta (\$) Saldo final (\$) Interés (%)

Created with Bonita Open Solution

7.1 - Criterios de descomposición

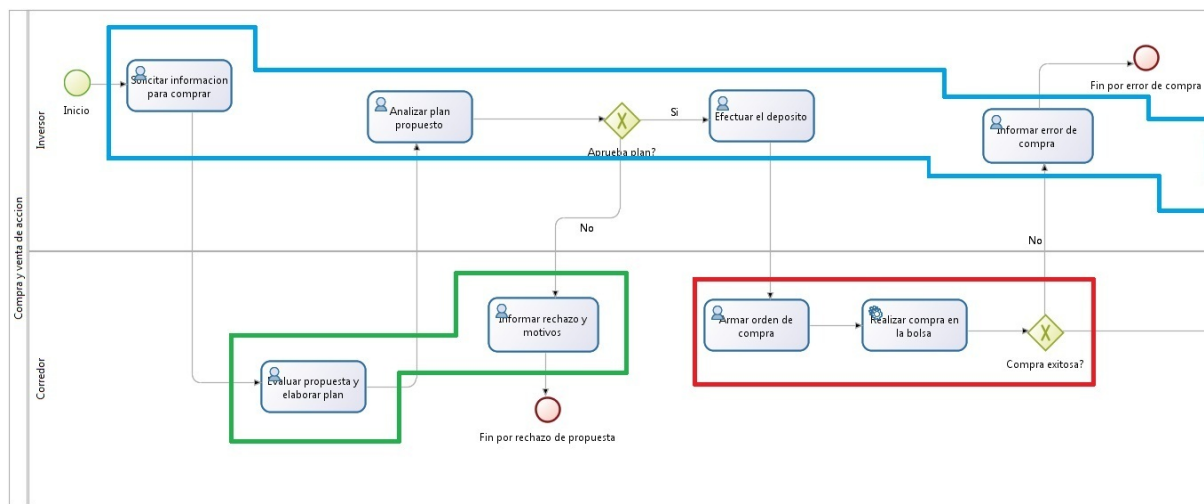
Habiendo ya explicado el proceso de compra y venta de acciones en su totalidad, enumeremos ahora las características del ambiente donde se ejecuta dicho proceso para determinar la forma en que procederemos a descomponerlo.

- Respecto a la disponibilidad: sobre este tópico podemos observar dos grupos que se diferencian en torno a la disponibilidad y formas de acceder a sus tareas, y por ende, al motor de procesos. Esto se da a nivel de tipo de actores, inversores y corredores. Notamos en este punto que los corredores estarán en un lugar fijo, probablemente todos trabajando en un mismo edificio, cumpliendo el mismo horario y, por consiguiente, sobre un mismo servidor al que accederán a través de una red local de alta velocidad y estabilidad. Este sería un enfoque de BPM más tradicional ya que su ejecución es local y corporativa. Por otro lado, los inversores no están en un lugar fijo, ya que aparte de representar a personas que tienen realidades, rutinas y horarios diferentes, el proceso está orientado a que ellos puedan en cualquier momento de su día, interactuar con los corredores para la compra o venta de acciones, sin importar donde se encuentren físicamente y en cualquier momento. Dada esta particularidad de

actores, sus tareas deben residir en un servidor cloud (BPM as a Service) que permita ejecutar sus tareas siendo necesario para esto solo un navegador web (smartphone, laptop, PC de escritorio) y acceso a Internet.

- **Respecto a la seguridad:** en materia de seguridad supondremos que el sistema de la bolsa de valores tiene como condición necesaria para su ejecución, interactuar con entornos y enlaces que estén cifrados. Esto ocurre ya que manejan información sensible y que de filtrarse comprometería a las partes. Dado este escenario contaremos con un servidor cuyo contenido está cifrado y sus peticiones se realizan bajo el protocolo de red HTTPS. En dicho servidor se ejecutarán las tareas que invocan al sistema de la bolsa.

Dadas estas características, el proceso distribuido estará dividido de la siguiente manera:



continuación...

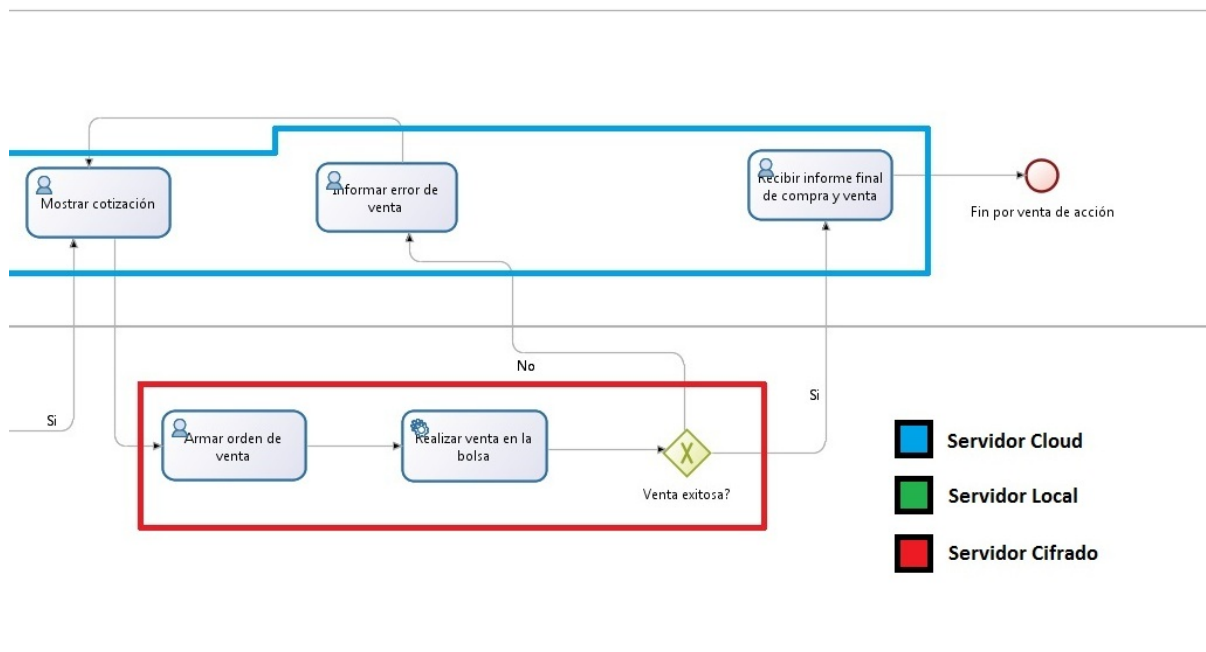
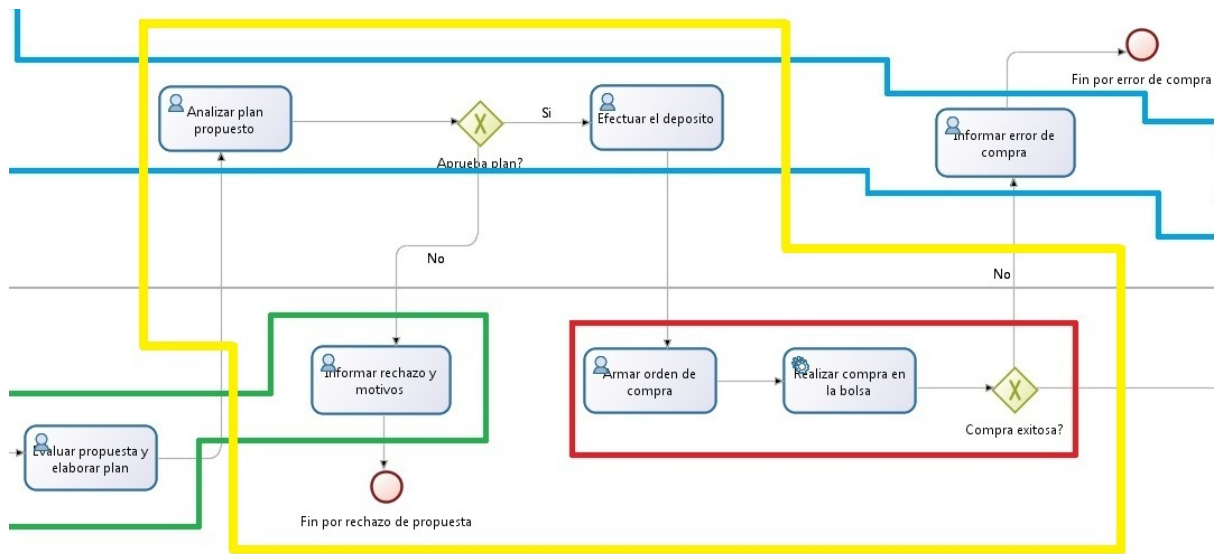


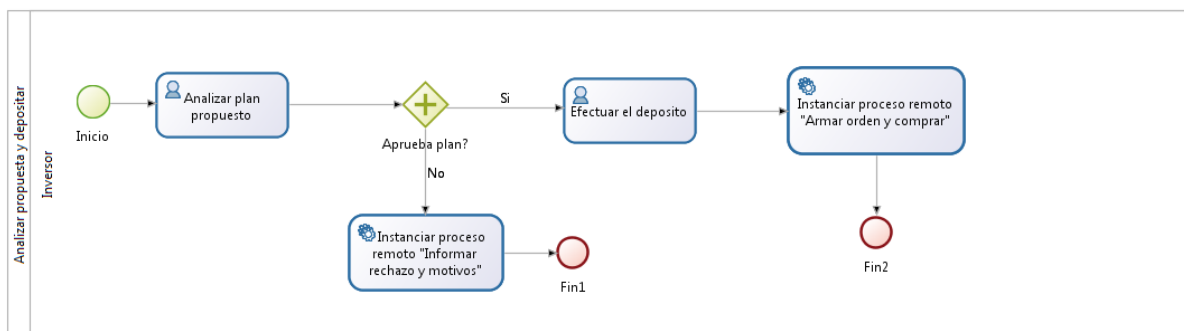
Figura 7.2 . Diagrama de proceso “Compra y venta de acción” distribuido.

Como vemos en la Figura 7.2, las tareas correspondientes al inversor irán al motor de procesos en el servidor cloud, que proporcionará accesibilidad a través de Internet, dado que estos actores no se van a encontrar en una locación fija. Caso contrario, las tareas referentes a los corredores estarán en un servidor local ya que estos sí tienen horarios y locación fija. Por último, un subconjunto de las tareas de los corredores, específicamente las que interactúan con el sistema de la bolsa, residirán en otro servidor local salvo que éste tendrá la particularidad de estar cifrado y que se comunicará mediante protocolos seguros.

Veamos de qué forma quedarán estos subprocesos, con respecto a su estructura y funcionalidad, para que permitan su ejecución distribuida. Tomemos el caso de esta porción del proceso señalada a continuación:



Aquí tenemos el subproceso perteneciente al inversor, que viene de recibir la propuesta del corredor, y se dispone a analizarla. En caso de aprobar, procederá a efectuar el depósito y luego pasará el control al servidor cifrado para efectivizar la compra. Caso contrario, delegará el control al servidor local informando al corredor el motivo de su rechazo. Veamos cómo estaría diseñado este subproceso:



Como observamos, cuando el flujo requiere un paso del control a otro subproceso en otro servidor, previamente tenemos una tarea de sistema que ejecuta el conector instanciador de procesos remotos. Necesitamos definir, entre otros, datos como el identificador de proceso

remoto y la URL de la API REST de dicho servidor para que se logre la instanciación remota y quede el registro de dicho enlace en la base de datos local que nos servirá luego de guía para obtener los registros de eventos de todo el proceso disturbios completo.

7.2 - Análisis de minería del proceso con ProM

Una vez que tenemos a los procesos desplegados cada uno en sus servidores correspondientes, simulamos la ejecución de sus instancias. Luego de un mes, ejecutándose una solicitud por día, contamos con 20 instancias del proceso de compra y venta de acciones. Cada uno fue dejando su información en sus registros de eventos locales. Lo que nos resta es mediante los web services desarrollados, obtener los registros del proceso completo.

Los nombres de los actores identificados en el proceso son:

- Inversores: inversor1, inversor2, inversor3, inversor4, inversor5.
- Corredores: corredor1, corredor2.

Se tomó esta decisión para que el análisis de las capturas de ProM sea más explicativo.

En este ejemplo en particular, como el criterio de descomposición principal fue orientado a actores (inversores y corredores) y dada su naturaleza de “ir y venir” entre los participantes prácticamente de a una o dos tareas consecutivas máximo, los subprocesos obtenidos no poseen la suficiente información y complejidad para realizar el análisis por partes (es decir que cada uno de ellos sea el recurso de entrada en ProM). Esta opción es viable cuando la descomposición tiene una orientación funcional, y se encapsula y delega todo un comportamiento complejo lo suficientemente grande (relativo al proceso completo en cuestión) como para que la herramienta de minería nos de resultados que nos sirvan para tomar decisiones estratégicas. Funciona cuando el subproceso tiene un concepto de modularización de comportamiento, donde éste resuelve un tema que no es motivación específica del proceso que lo contiene, o posee variables o actores ajenos a su proceso contenedor.

Luego de esta observación, nos disponemos a obtener el log unificado de todos los subprocesos distribuidos en los tres servidores en un solo archivo de log en formato XES y así tener una visión global del proceso completo.

Esto lo realizamos desde la aplicación de monitoreo, en su menú de logs, seleccionamos el proceso de la lista y seleccionamos la opción “Log unificado” como muestra la Figura 7.3:

Logs

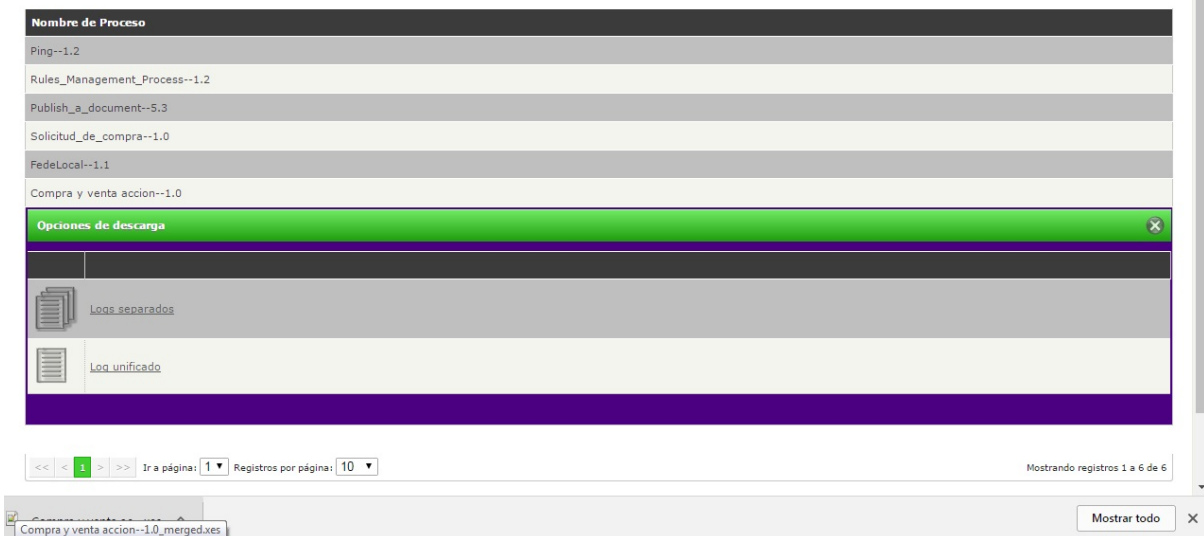


Figura 7.3. Obtener log unificado del proceso desde la aplicación de monitoreo.

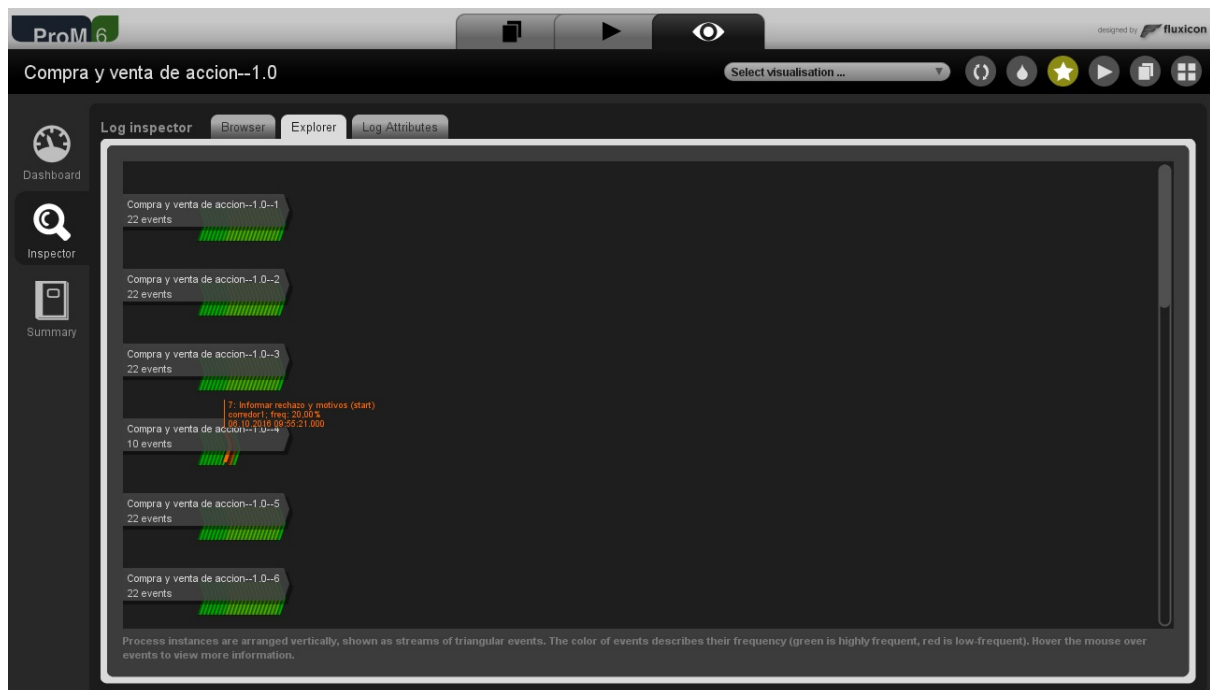
Ahora este log es el que importamos desde ProM. Luego, podemos visualizarlo con el View Inspector para obtener una visión general del registro. En esta sección se pueden ver los diagramas de los eventos por caso, y debajo las clases de eventos por casos también. Como datos clave nos muestra la cantidad de procesos analizados (uno en nuestro caso), la cantidad de casos (20), la cantidad total de eventos que se observan, la cantidad de clases de eventos y la cantidad de tipos de eventos y finalmente la cantidad de originadores. A la derecha vemos la fecha y hora del primer y último evento del registro. La presentación es la siguiente:



La próxima es la pestaña de inspección. En ella podemos ver cada una de las instancias por separado, con sus valores de sus variables, eventos, fechas de cada uno y actores involucrados.

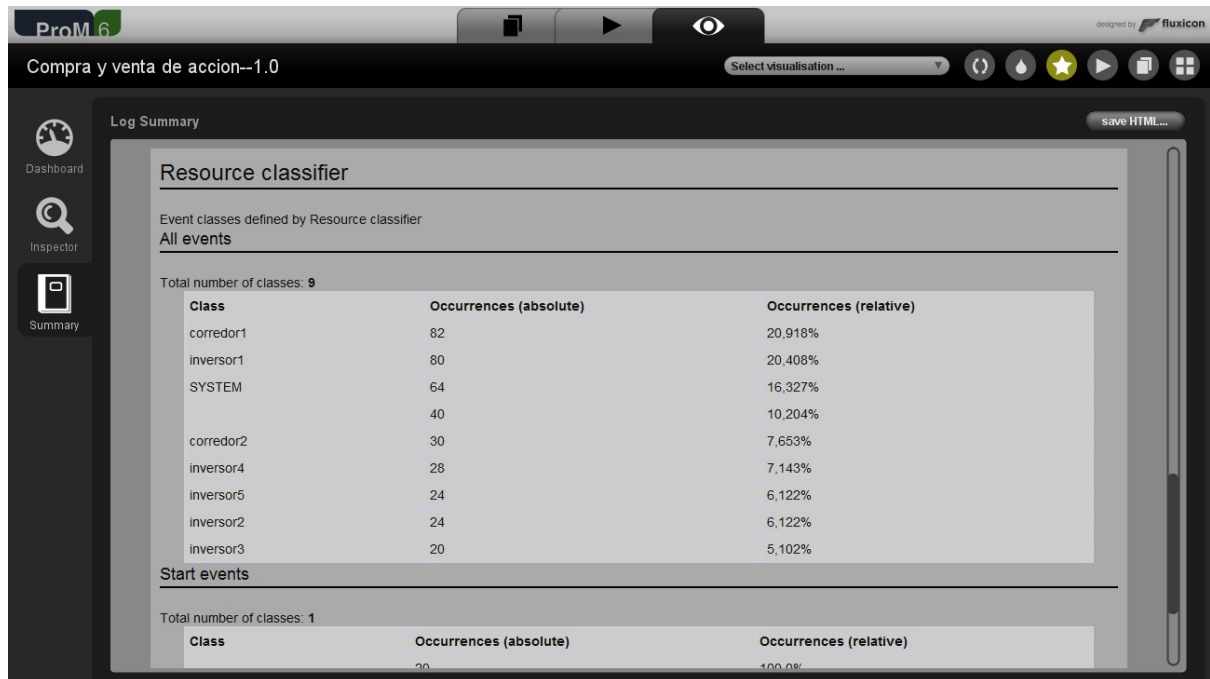


Podemos observar que la instancia numero 2 fue iniciada por el inversor1, compró acciones de Comercial del Plata y obtuvo 70% de interés. Dentro de esta pestaña tenemos una vista de exploración donde vemos las tareas por instancia con 3 diferentes colores, representando la frecuencia que tiene cada una en el espectro global (verde es alta frecuencia, amarillo media y rojo baja)



En el explorador vemos que la instancia 4 tiene actividades marcadas en rojo (frecuencia del 20%), que son las relativas al rechazo de la compra de la acción, por lo que ya podemos inferir que se aceptaron más propuestas de las que se rechazaron.

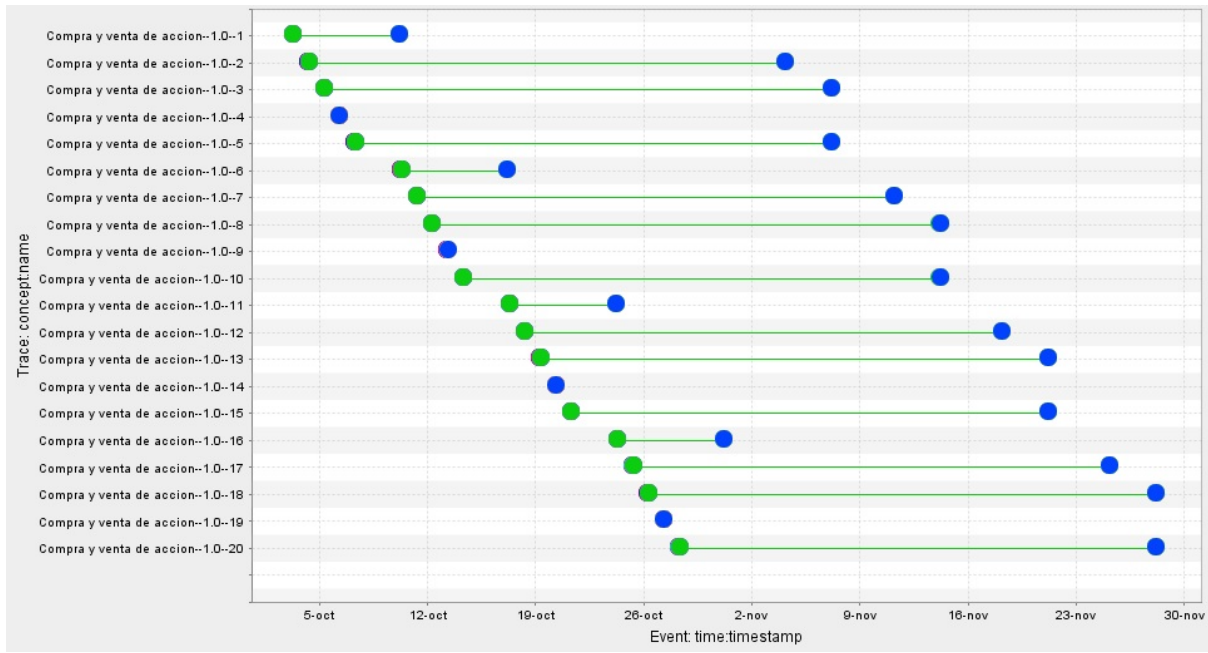
En la última pestaña tenemos un resumen donde se nos muestran la cantidad de instancias y eventos. Aparte tenemos un resumen de la cantidad de veces que se ejecuta cada tarea y su frecuencia. Así también para los actores como se muestra a continuación:



En este resumen de actores podemos ver que los que más interactuaron con el procesos fueron inversor1 y corredor1, casi 3 veces más que sus seguidores corredor2 e inversor4. Hay una gran brecha entre estos mientras que entre los demás inversores es casi la misma frecuencia.

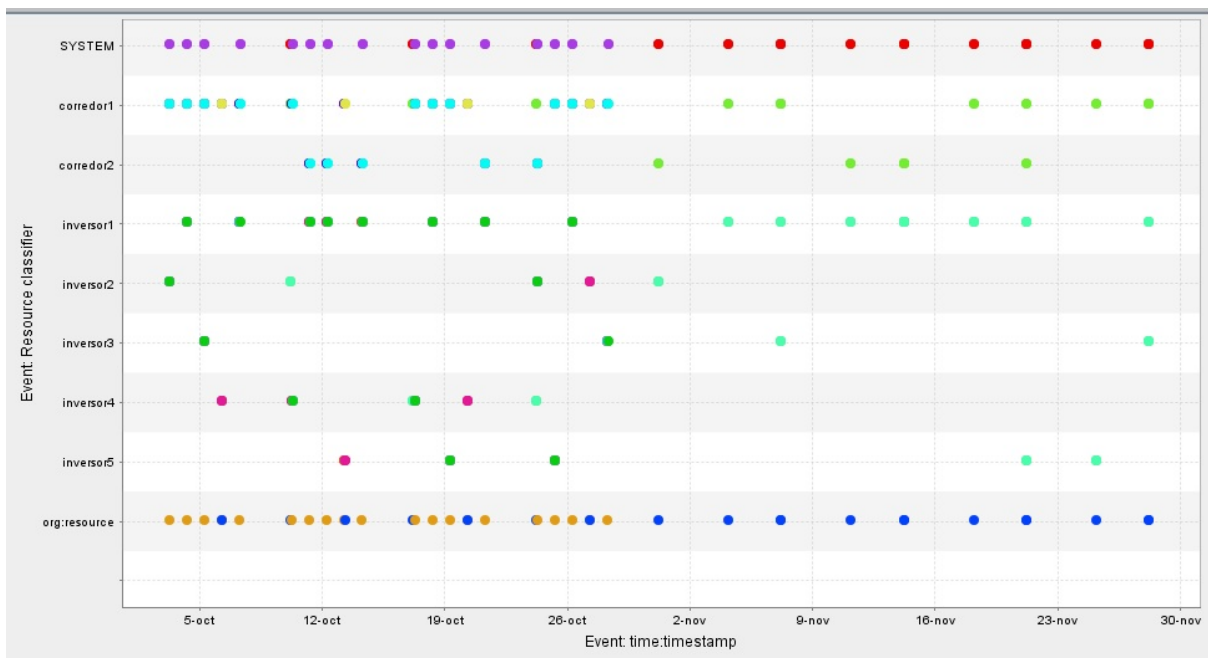
7.2.1 - Dotted Chart

Esta técnica permite visualizar información del proceso organizada sobre un gráfico de ejes x e y, siendo éstos configurables para mostrar y proyectar puntos que coincidan con la información contenida en el log. Dado que sus ejes son configurables, podemos proyectar distintos gráficos con los datos que contamos. Veamos primero la versión clásica que es instancias sobre tiempo:



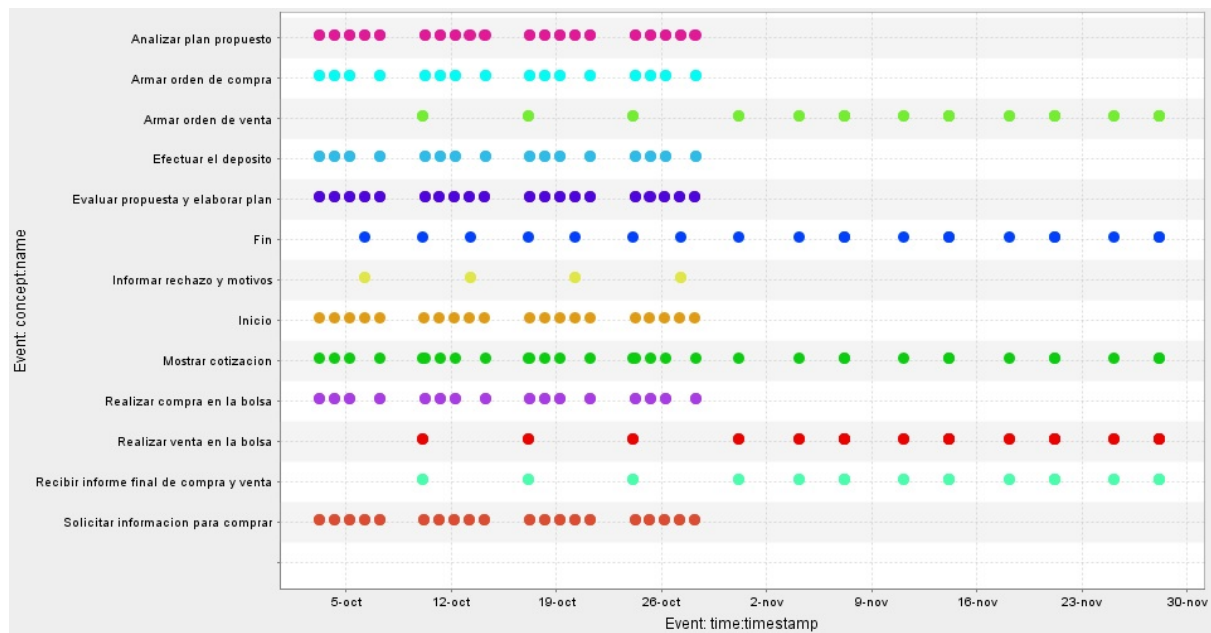
Aquí podemos observar que, como era de esperar, el proceso está dividido en 2 grandes partes: la primera que está relacionado con la compra de la acción y la segunda con la venta. Vemos entonces que hay instancias que compran y venden (las que tienen la union de linea verde) con distintos tiempos y otras que no compran, rechazando la propuesta y finalizando la instancia (las que sólo tienen un punto azul sin linea). Corroboramos lo que señalamos anteriormente en el inspector que la mayor parte de las instancias lograron comprar y, por lo tanto, vender. Lo que aquí podemos ver con mayor claridad es que tenemos dos grupos dentro de lo que compraron: los que vendieron pasados aproximadamente 7 días y los que lo hicieron alrededor de 30 días.

Veamos otro posible gráfico, esta vez la actividad de los actores a través del tiempo:



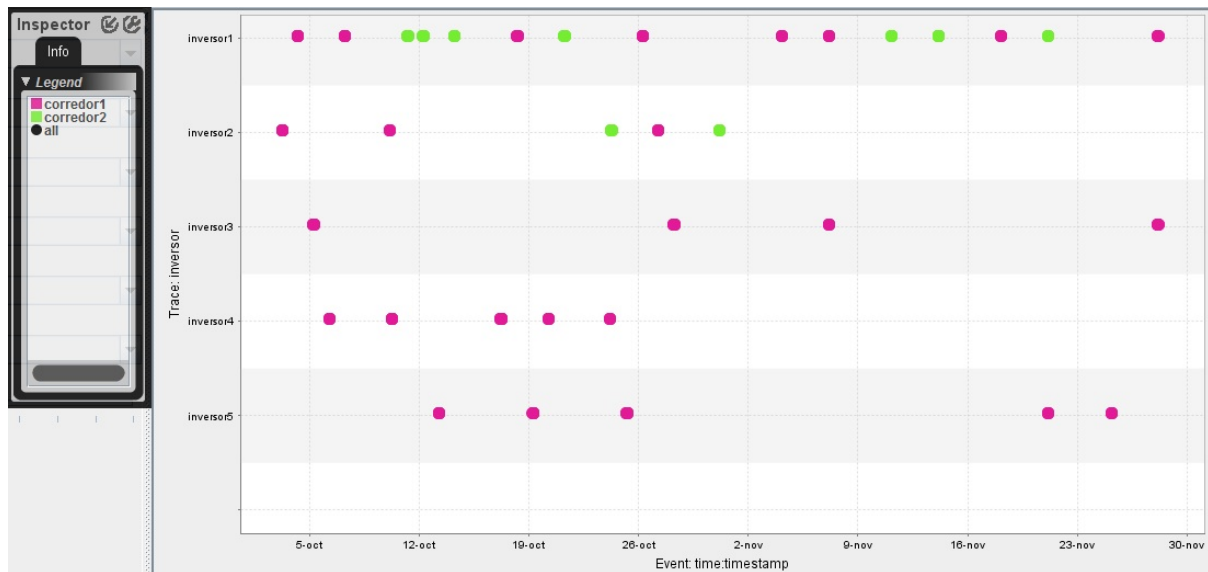
Aquí notamos que, aparte de los actores que veníamos mencionando, aparece SYSTEM que representa a las actividades que se realizan en comunicación con el sistema de la bolsa de valores. El último “org:resource” hace referencia a las actividades de inicio (amarillo) y fin (azul) del proceso. Podemos ver aparte que el inversor3 es el que más esporádicamente interactúa con el proceso y que el corredor1 trabajo casi todos los días de Octubre.

Observemos otro gráfico orientado a las fechas, tareas en el tiempo:



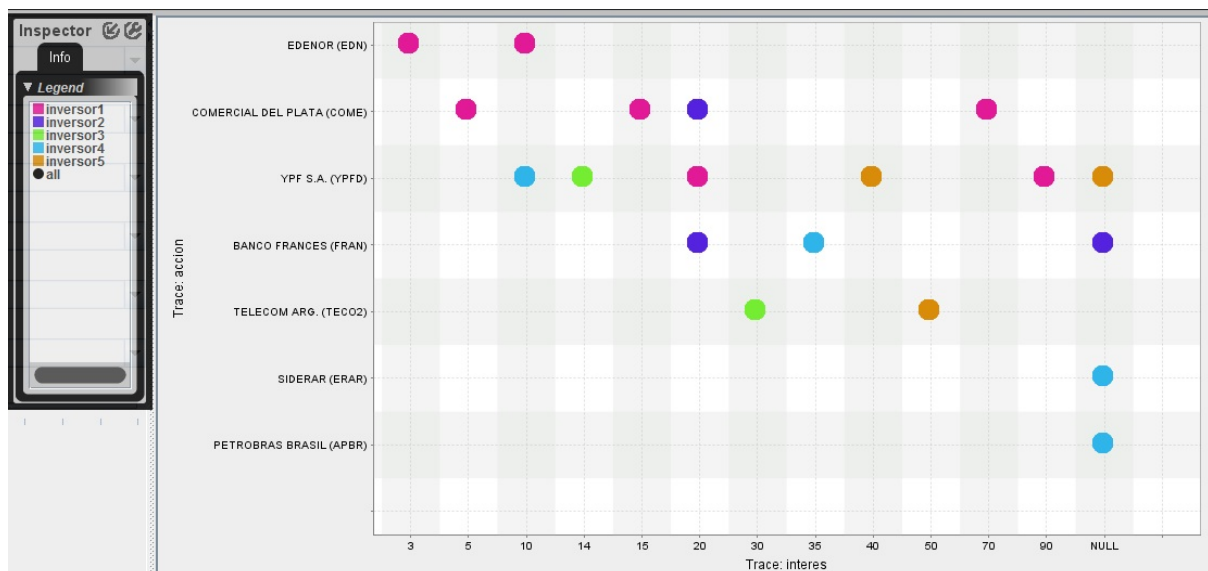
Podemos ver que tenemos un patrón de comportamiento dividido por meses. En Octubre conviven todas las tareas relativas a la compra y algunas a la venta, mientras que en Noviembre, actividades exclusivas de venta. Podemos ver también que en Octubre se trabajó todos los días hábiles (esos espacio entre cada grupo de 5 puntos representan los fines de semana).

El último en relación al tiempo tiene otro nivel de señalamiento: inversores en el tiempo pero destacando con cual de los 2 corredores interactuó:



Como vemos, aparte de la actividad en el tiempo de los inversores, destacamos con rosa cuando este estaba realizando una compra o venta la cual estaba administrada por el corredor1 y con verde cuando era con el corredor2. Esto nos permite ver claramente que los inversores que solo interactuaron con ambos corredores fueron los inversores 1 y 2 y que los demás fueron exclusivos del corredor 1.

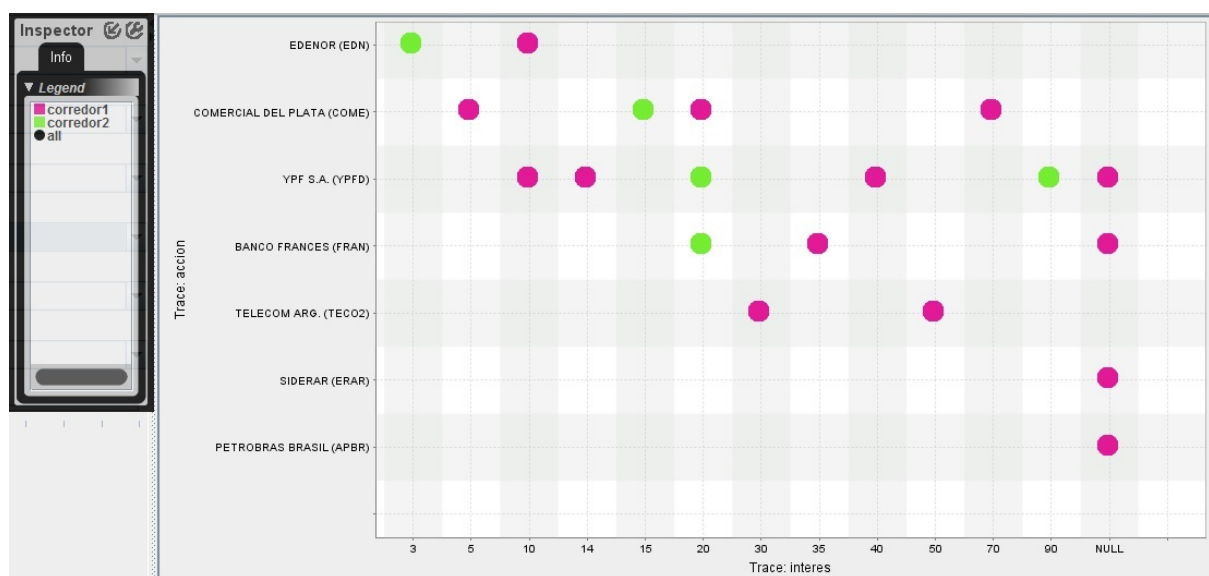
En otro plano de análisis, más en sintonía con el dominio del proceso, podemos proyectar las acciones elegidas sobre el rendimiento de sus inversiones, es decir su interés obtenido. Siguiendo al ejemplo anterior, vamos a destacar en este gráfico primero a los inversores y luego a los corredores involucrados para tener una visión más integradora de la situación.



Podemos notar que las inversiones que mejor interés dieron fueron YPF y Comercial del Plata (ambas vendidas por el inversor1) y que las que menos dieron, Edenor y nuevamente Comercial del Plata, también fueron inversión del inversor1. Aparte de notar que el rendimiento medio estuvo entre el 10% y el 35%, se puede destacar que Petrobras y Siderar

nunca fueron aceptadas como propuesta de inversión. Esto nos puede servir para detectar que sus proyecciones no eran buenas, o que simplemente los inversores no confiaban en estas compañías. También este gráfico nos permite crear perfiles de inversores, por ejemplo, podemos señalar que el inversor1 es el que más veces invirtió pero su promedio está dentro de la media. Nunca rechazó una propuesta y tuvo rendimientos muy variados. Lo podríamos catalogar como impulsivo y vertiginoso, por ejemplo, y esto nos servirá para armarle propuestas personalizadas en un futuro. Mientras que el inversor5 fue el más criterioso ya que sus 2 inversiones superan el rendimiento medio y registra un rechazo (quizás porque no le pareció el momento adecuado de comprar).

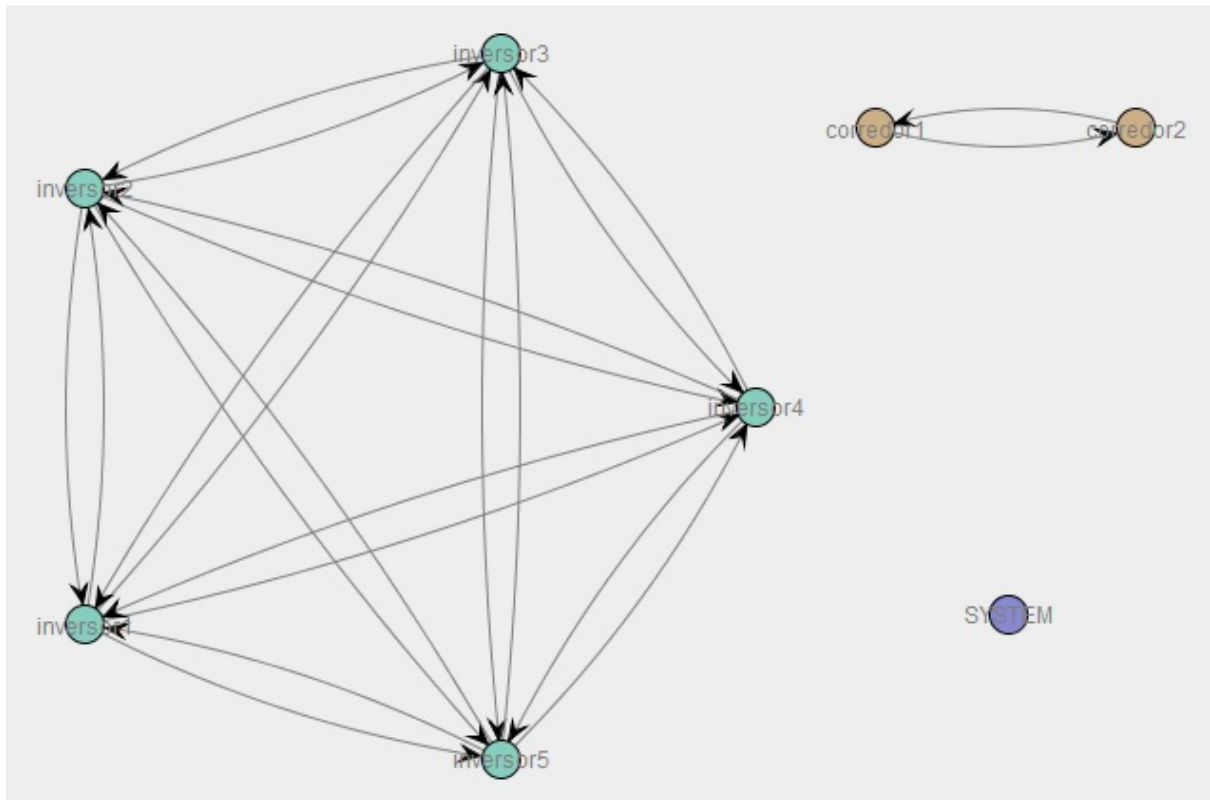
Veamos el mismo gráfico pero para ver en relación a los corredores para ver qué tan acertadas y rentables fueron sus propuestas y quien fue el más rechazado:



A primera vista vemos que el corredor1 es al único que le rechazaron sus propuestas, esto puede ser hasta lógico porque su nivel de interacciones mucho más alta que la del corredor2). Pero después notamos que en los extremos del gráfico (menor y mayor interés) aparece el corredor2. Luego sus propuestas se encuentran en la media general. Podemos concluir que si bien el que más trabajó fue el corredor1, el más eficiente fue el 2.

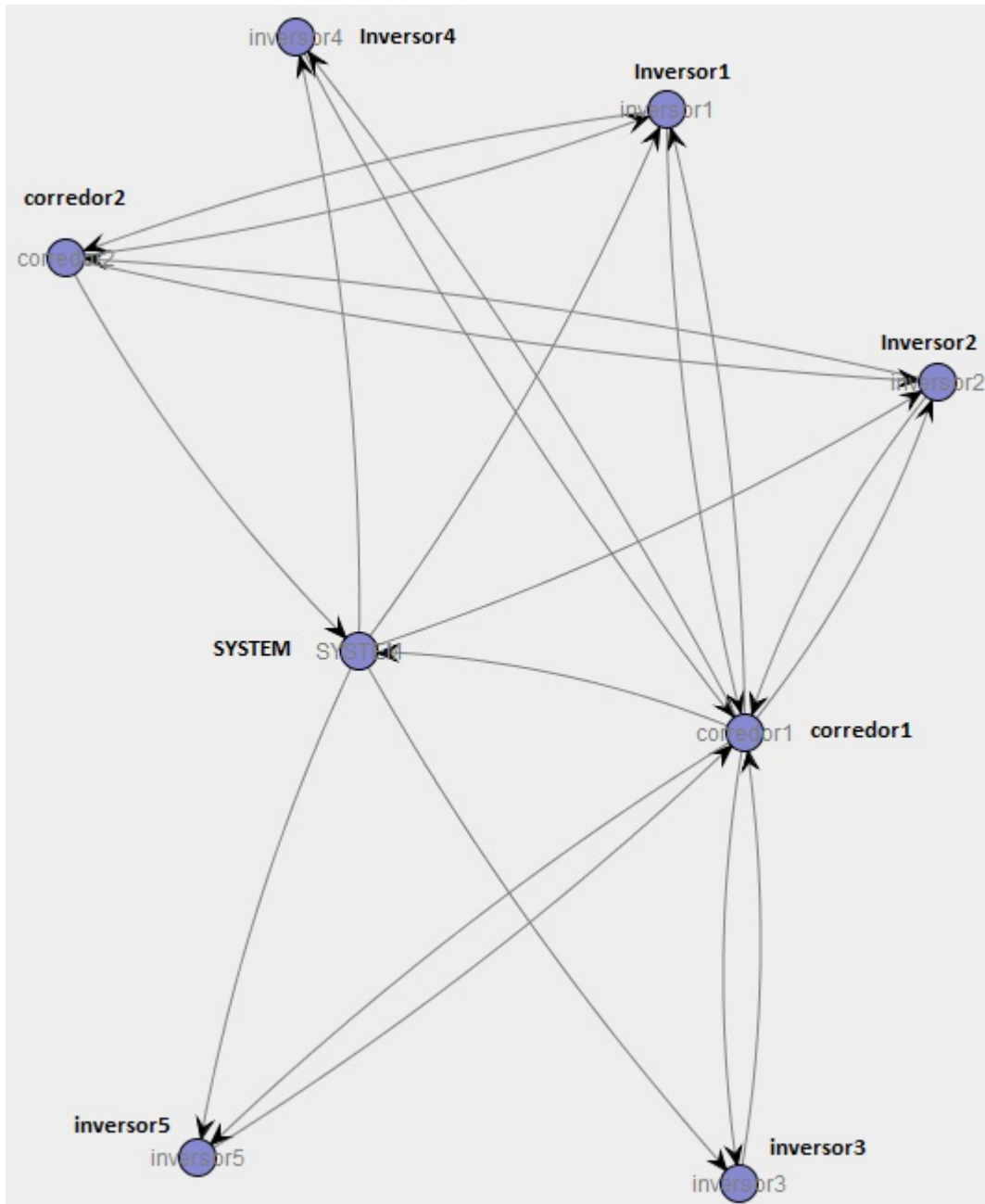
7.2.2 - Minería de actores

En esta sección veremos 2 gráficos que tienen como objeto de análisis central a los actores. Primero los veremos agrupados en base a su comportamiento (es decir tareas realizadas) y luego un grafo que representará sus interacciones



Aquí podemos corroborar que tenemos 2 grupos y un actor que representa al sistema, que para este caso particular no es novedad ya que decidimos nombrar a los actores en base a su rol (por motivos de representación y legibilidad de los gráficos), pero esto no suele ser así, cada actor suele tener su usuario por lo general conformado por la primera letra de su nombre, seguida por su apellido. En ese caso este gráfico sería menos predecible en cuanto a los nodos que componen cada grupo, pero podemos observar que todos tienen conexión entre ellos, lo que quiere decir que todos los del grupo realizan prácticamente las mismas tareas. Con esto podríamos diagnosticar algún desfase de comportamiento, por ejemplo un actor que está cumpliendo tareas de otro o que hace tareas de varios grupos cuando no le corresponde, dando lugar a una advertencia o que el propio comportamiento requiere la creación de un nuevo rol que tiene tareas mixtas.

Veamos ahora el grafo que representa la interacción entre los actores:



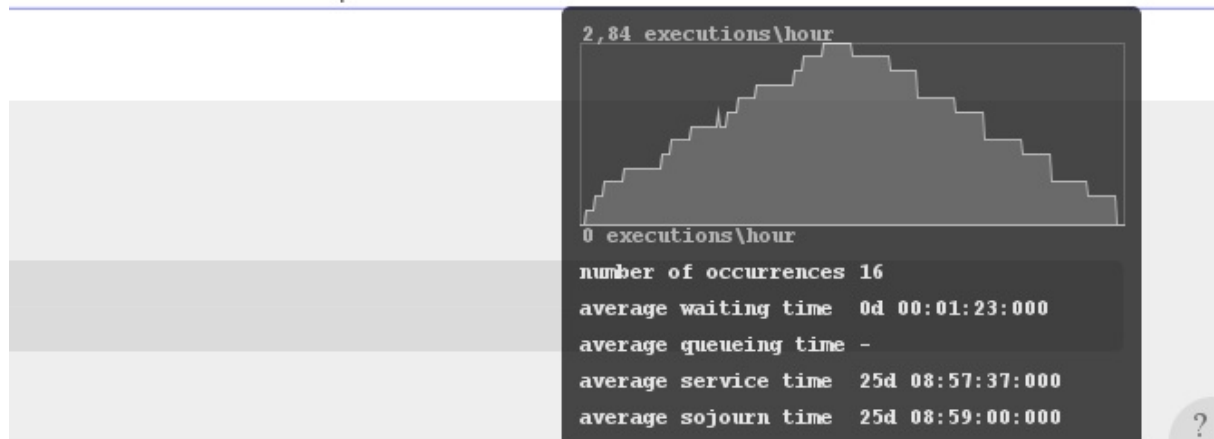
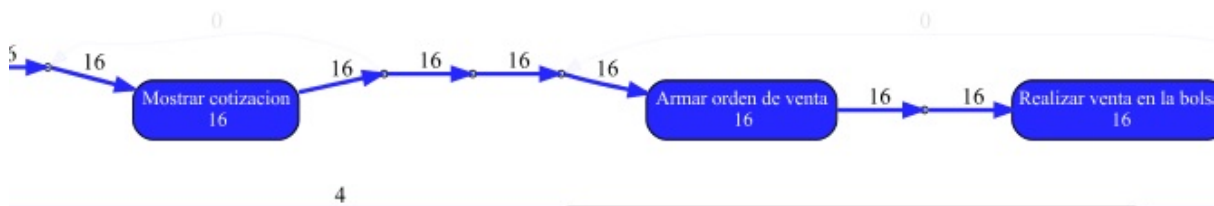
Notamos que el corredor1 tiene mucha más interacción que el corredor2, que los inversores no interactúan entre sí y que el sistema recibe peticiones de los corredores y envía respuestas solo con los inversores.

7.2.3 - Diagramas de transición

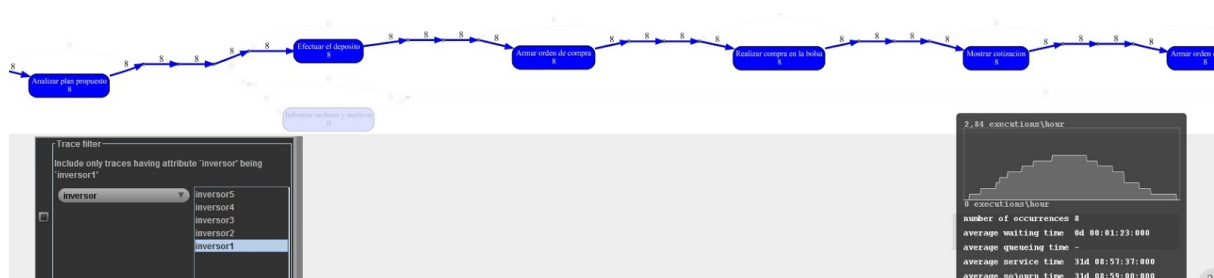
En esta sección veremos todos los gráficos que se enfocan en la transición de una tarea a otra, frecuencia, tiempos, demoras, caminos más tomados, etc. Empecemos por la cantidad de veces que se toma, por ejemplo, el camino de aceptar una propuesta contra su rechazo:



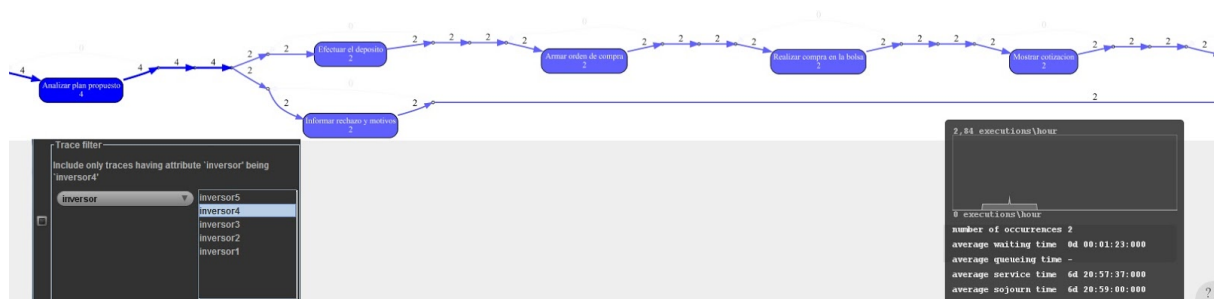
Aquí podemos observar que de las 20 instancias que registramos, 16 son aprobadas y 4 rechazadas. Podemos notarlo por los numeros que estan en los vértices, como en la intensidad de los colores (más intenso el azul, más frecuente el camino con respecto al otro). Si nos posicionamos sobre una tarea podemos obtener los siguientes datos:



Por ejemplo, podemos ver que del total de las 16 instancias que deciden comprar, el plazo medio de la inversión es de 25 días. Esta vista también nos permite filtrar por algún criterio, veamos a continuación para un inversor determinado cómo varía éste gráfico y éste último indicador.

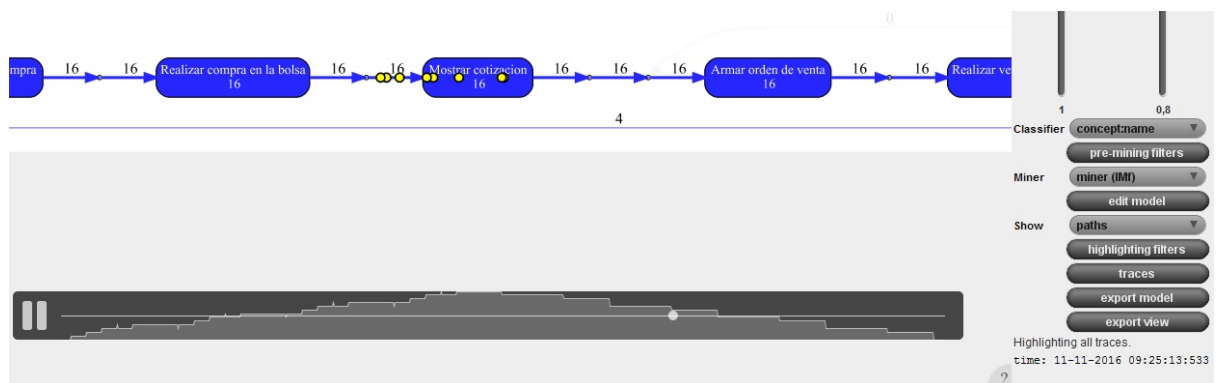


Como señala el gráfico, el inversor1 aceptó todas las propuestas que le hicieron (ocho para ser exactos) y su plazo promedio de espera para vender su acción está un poco por encima de la media, 31 días. Ahora pasemos a ver la misma porción pero sólo del inversor4.



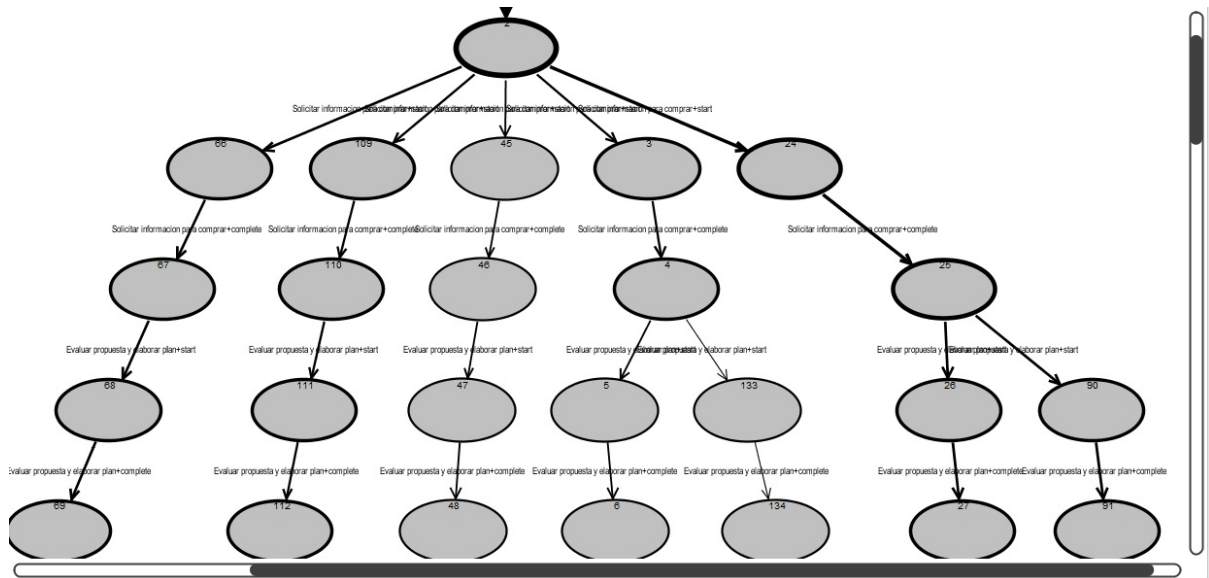
Destacamos aquí que tiene un mecanismo de decisión de 50/50. De las 4 propuestas recibidas sólo aceptó la mitad y tiene otro perfil en cuanto al plazo de sus inversiones. Este está bastante por debajo de la media y ronda los 6 días.

Por último, esta vista nos permite observar una simulación de la ejecución de las instancias a lo largo del tiempo. Permittiéndonos detectar más visualmente algún cuello de botella.



Aquí nos muestra como hay un acumulamiento de instancias en la tarea “Mostrar cotización”, pero esto es esperable ya que esta es la tarea que queda pendiente hasta que el inversor decide completarla para dar la orden de vender.

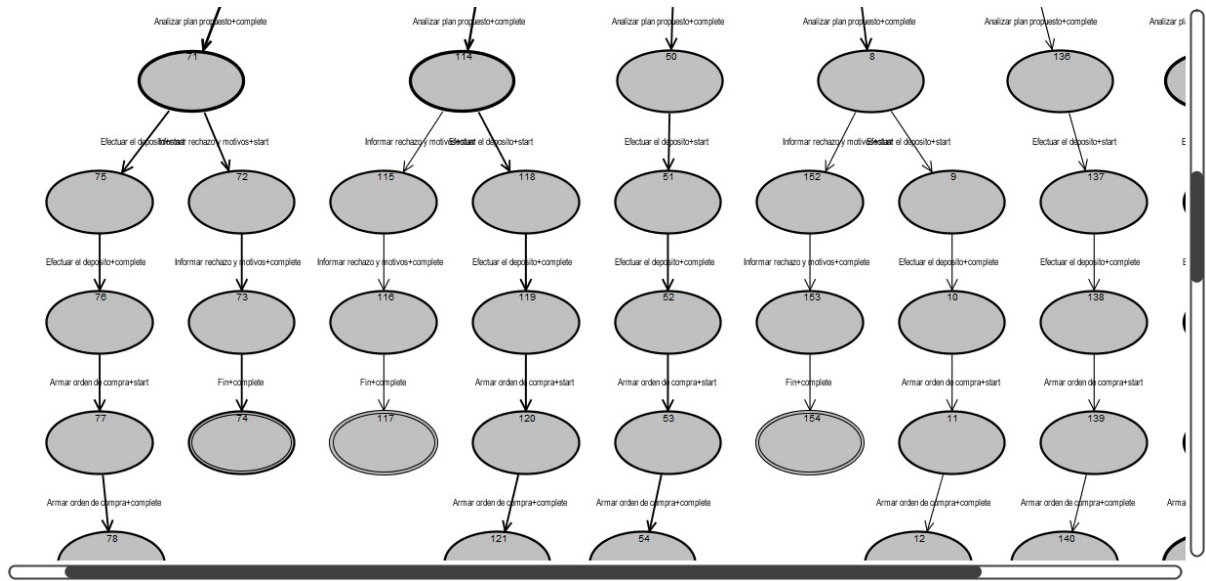
Dentro de los diagramas de transición, tenemos otro método que nos permite hacer otro tipo de análisis ya que tiene otro enfoque para armar las conexiones. Este se basa en nodos de transición, dividiendo caminos cada vez que registra un cambio de actor y/o de tarea. Veamos cómo se representa el inicio del proceso y qué conclusiones podemos sacar:



Vemos que el inicio se ejecuta indistintamente y que no tiene actor asignado. De él se desprenden las transiciones referentes a la primer tarea del proceso que es la de solicitar inversión. Podemos notar que hay una bifurcación por actor que ejecuta esta tarea (una por cada uno de los cinco inversores) y que la intensidad de la arista es proporcional a la frecuencia de ejecución de dicho camino por sobre el total. Podemos observar así que el inversor1 (el de la derecha) es el que más instancias inició por su grosor de arista. Recordemos que cada tarea está mapeada con su evento de inicio y de fin, por lo tanto, el primer nodo luego del inicio centralizado, representa el evento de inicio de la primer tarea y el segundo su finalización. Si continuamos el flujo, vemos que luego de completar la tarea de solicitud, sólo los flujos que representan a los inversores 1 y 2 (el siguiente a su izquierda) se vuelven a dividir. Por la altura del diagrama que ocupan vemos que, al igual que sus vecinos, representan la próxima actividad de propuesta de inversión. Lo que está representando esta división es que estos inversores interactuaron al nivel de la misma tarea tanto con el corredor1 como con el corredor2 por lo menos alguna vez (y con una frecuencia también dada por la intensidad de su arista que podemos decir a primer observación que fue equitativa para cada inversor).

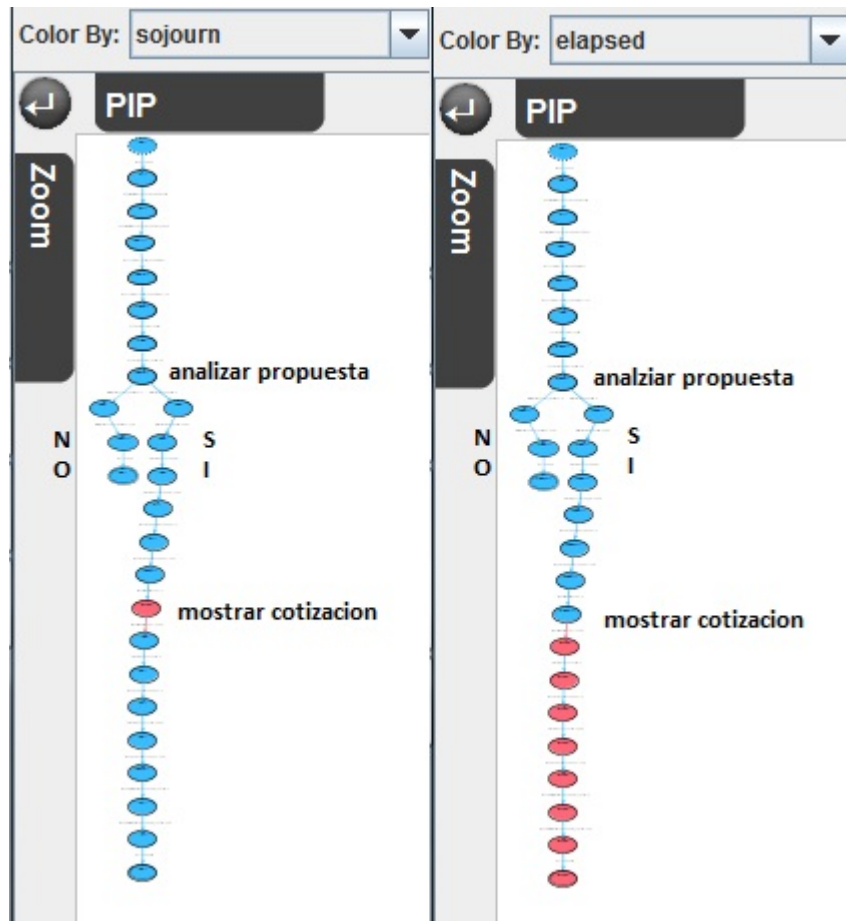
Aquí vimos entonces como cuando se incorpora un actor distinto al que venía o del que estaba, se produce una separación y marca caminos paralelos. Permite ver hasta qué punto las instancias tienen atributos (actores o tareas) similares y cuando se produce un cambio (y como repercute en los valores de instancia o su camino futuro)

Analicemos el otro caso de separación que puede ser por cambio de tarea (lo que se da comúnmente después de la ejecución de una compuerta exclusiva):



En una observación rápida vemos que cuando se dividen los caminos, hay uno que después de un par de nodos, termina. Esto representa el camino cuando se rechaza la propuesta, que luego notifica los motivos y termina. Podemos ver entonces que son 3 de los 5 inversores que rechazaron alguna vez alguna propuesta y por el grosor de sus nodos vemos que el inversor4 fue el que más lo hizo con respecto a los otros (nodo 74).

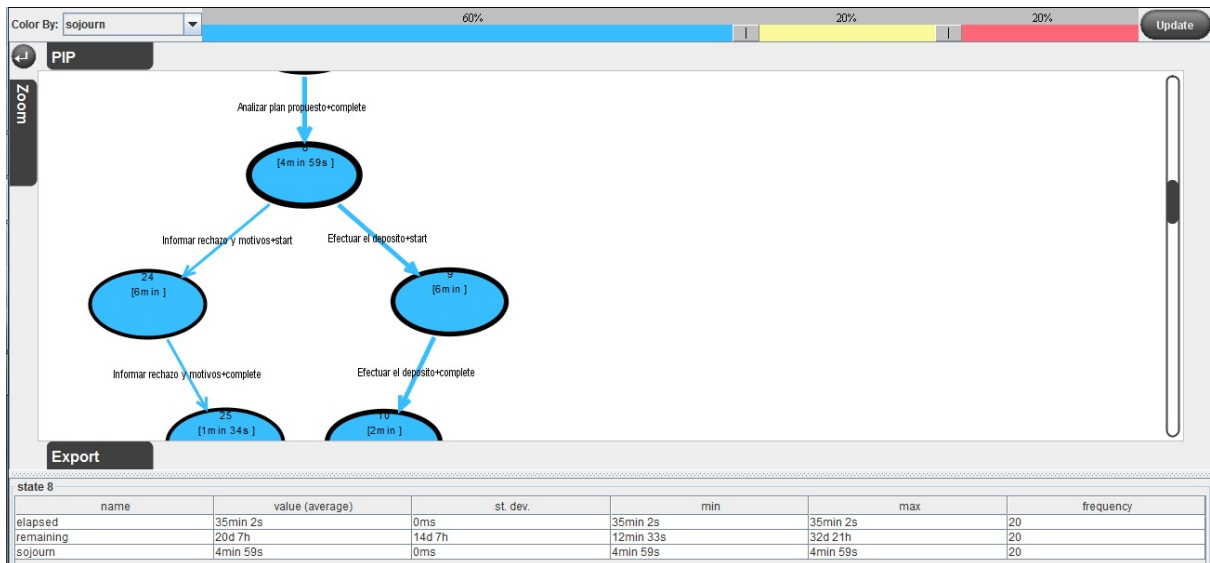
ProM tiene otra técnica que destaca con colores las tareas con respecto al tiempo que consumieron. Permite marcar el tiempo tanto para la tarea aisladamente, como mostrar tiempo acumulado.



(a) Transición tiempo por tarea (b) Transición tiempo transcurrido

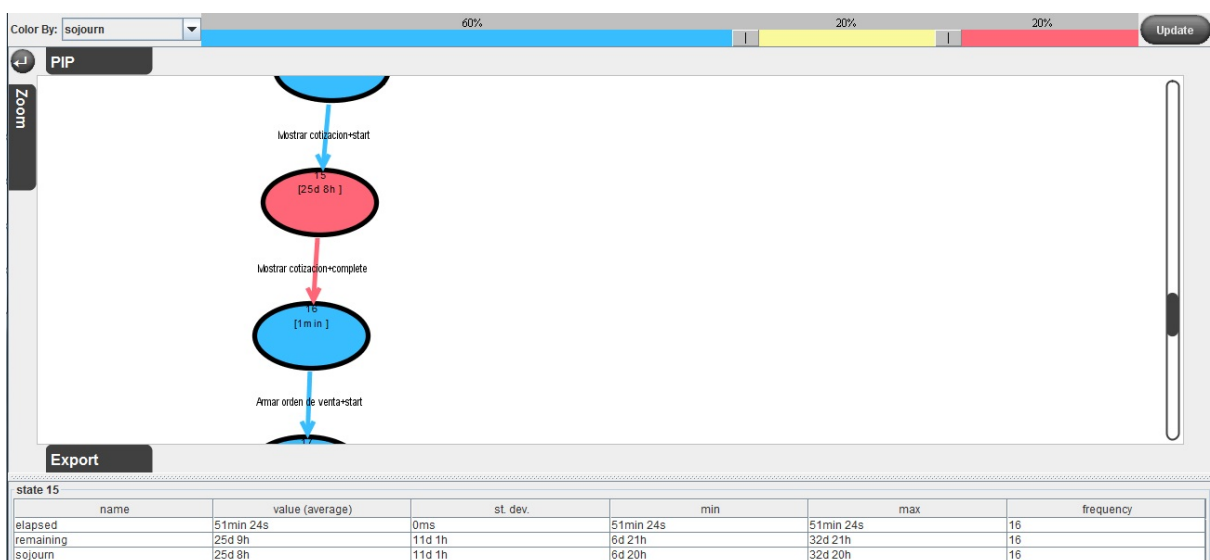
Como vemos en los gráficos en (a) tenemos el gráfico destacando la duración de cada tarea en particular. Es claro que todas las tareas tienen una resolución rápida salvo la que corresponde a esperar a vender la acción que se compró previamente (minutos contra días). Esta representación nos ayuda para detectar cuellos de botella, o tareas que están tardando más de lo que se espera. Una proyección de esa espera y retraso se puede ver en (b) donde se destaca con colores el tiempo transcurrido. Aquí vemos la implicancia de ese “efecto arrastre” de la demora de una tarea. Esto nos permite ver con más claridad que partes del proceso se ven retrasadas aunque su resolución individual sea rápida. Vemos como eso no lo marca el gráfico en (a) ya que las tareas posteriores a “Mostrar cotización” se realizan rápido y al mismo ritmo que las relativas a la compra. Pero en realidad tardarán más en el tiempo ya que esperan a que se complete la tarea de venta que es la que más tiempo consume. Esa característica la muestra de manera muy clara el gráfico (b). Es decir sabemos que se resuelven rápido pero que van a demorar en llegar a ejecutarse y que la demora en su resolución está ligada a cuando se decide vender la acción (que suele durar varios días en promedio). Este tipo de análisis nos puede interesar en ambientes distribuidos, donde tenemos al proceso en varios servidores, y cuando vemos la performance aislada nos da bien, pero no sabemos que su retraso (que no sale en los registros) es debido a una tarea que quizás demora y se encuentra en otro servidor, produciendo el mencionado arrastre.

Veamos que nos muestra si nos acercamos a la tarea de decisión que evalúa la propuesta



Nuevamente vemos que se toma más veces el camino de aprobar (el de la derecha) por el grosor de sus bordes. Si bien las duraciones son similares, vemos en la tabla de abajo que en la sección “remaining”, o sea, tiempo que falta para que termine el proceso luego de esta tarea, tenemos datos como el promedio, desviación estándar, mínimo, máximo y frecuencia. Podemos observar que el mínimo hace referencia a la situación de rechazo ya que sabemos que luego de informar, termina. Y es lógico que todo este proceso solo dure 12 minutos como señala la columna mínimo. Luego en máximo podemos observar el caso del proceso que más tiempo tardó, en este caso 32 días. Este tipo de gráficos y tablas nos dan una visión más interactiva del proceso, permitiéndonos parar en cualquier parte y ver que se hizo, que falta y cuánto tiempo transcurrió o necesitamos esperar para que termine.

Por último observemos que nos dice a la altura de la tarea que determina la venta, que es otro punto crítico del proceso.



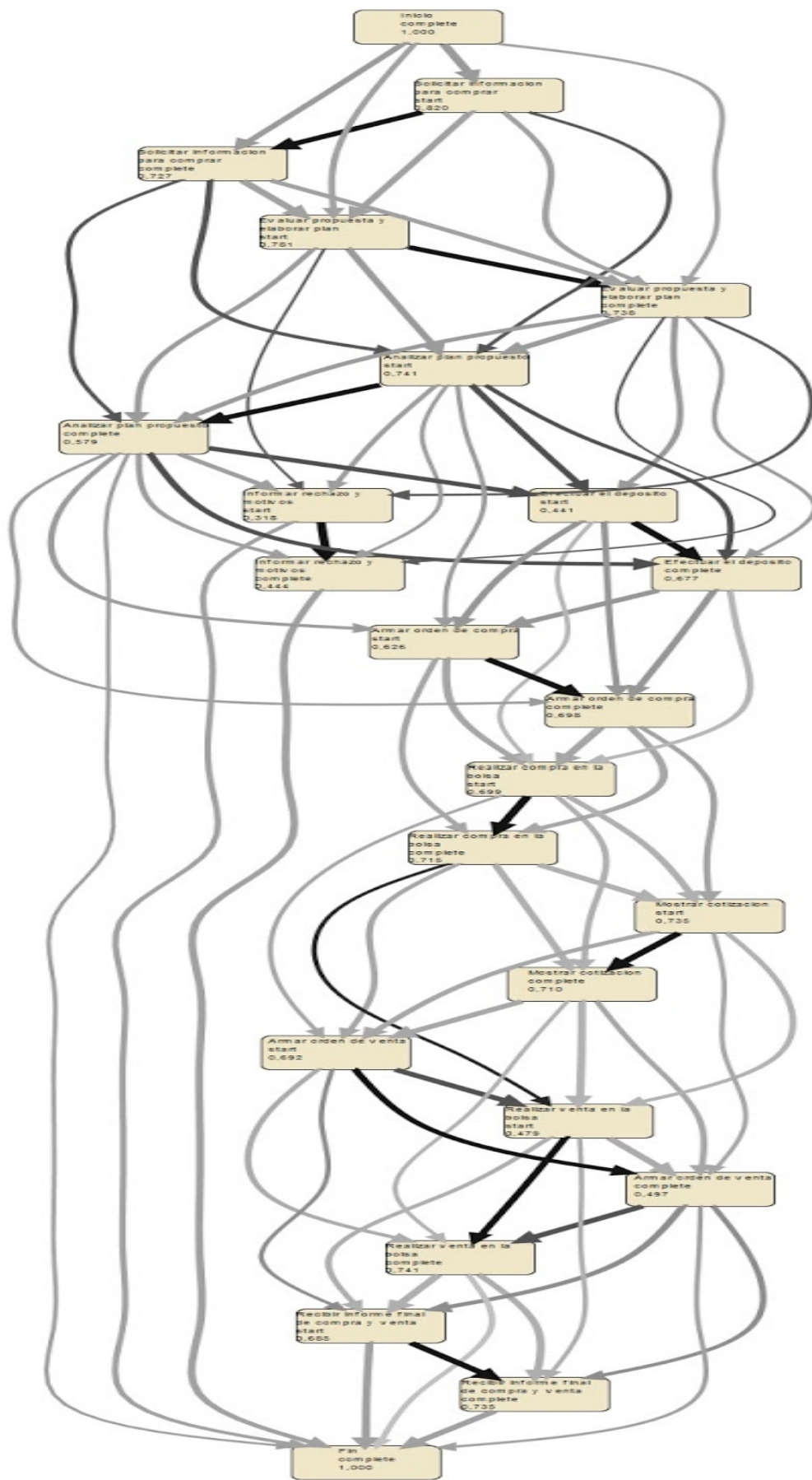
Ya en el extremo final del proceso notamos que tenemos 51 minutos transcurridos desde el inicio del proceso hasta este punto y que lo mínimo que tardó desde aquí en terminar fueron 6 días (y máximo 32 días). Esto nos demuestra y corrobora que el proceso está dividido en dos grandes partes cada una con conceptos de tiempo de espera bastante distintos y esto es correcto y esperable. Este gráfico nos permite validar esta regla de manera rápida e intuitiva.

7.2.4 - Fuzzy Miner

La técnica de Fuzzy Miner es útil en registros grandes, desestructurados y/o ruidosos. Es decir que costaría con otras técnicas poder analizarlos. Si bien esta es una ventaja, tiene como contraparte que el modelo presentado no suele mostrar una semántica real en las relaciones entre los nodos y eventos. Sirve principalmente para explorar o intentar inferir un prototipo de modelo cuando no se cuenta con uno, basado en los eventos registrados.

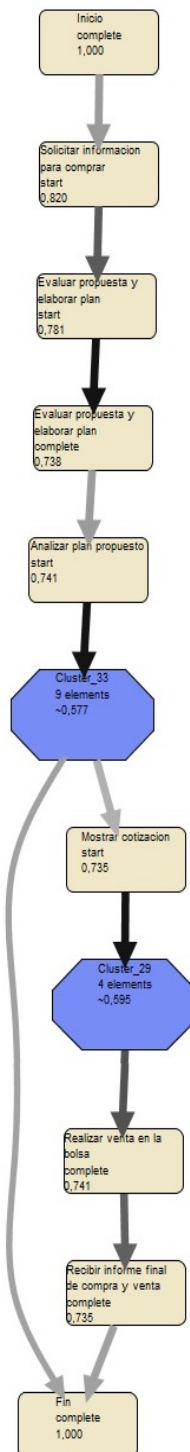
Este no es el caso, ya que contamos con un proceso estructurado y con poca desviación, los caminos y tiempos si bien varían no afectan al modelo de proceso en sí, pero veamos qué resultados arroja porque permite algunas configuraciones que las otras técnicas no.

Si le aplicamos esta técnica nos mostrará una imagen como las que venimos viendo, pero permite, mediante la calibración de la variable "Cutoff" podemos "expandir el filtro" de análisis de relaciones entre las tareas y ver relaciones que en otros casos no veríamos.



En este modelo vemos como tenemos, aparte de las ya conocidas líneas intensas que marcan el flujo más frecuente, otro conjunto de relaciones que se podrían considerar como derivadas. Esto nos sirve si queremos hacer un análisis no tan lineal y ajustado a la ejecución sino a la relación entre dos tareas que no son necesariamente consecutivas

Por otro lado, Fuzzy miner nos permite realizar “clusters” (grupos) de tareas para tener una visión más “corta” y más simple a la vista. Esto también es calibrable, veamos como resulta el proceso bajo este tratamiento:



Como hemos visto a lo largo de este capítulo, las técnicas de minería de procesos nos proveen herramientas para el análisis y optimización de procesos de negocio. Podemos hacer análisis desde varias perspectivas como hemos visto: orientadas al tiempo, a la actividad, a la transición y a los actores. Estimar tiempos medios, mínimos y máximos que nos dan una orientación de como se seguirá comportando el proceso en el futuro. También nos brinda la posibilidad de detectar fallos, anomalías, cuellos de botella, que resultaran en el rediseño y/o redistribución del proceso, dando lugar al reinicio del ciclo de vida de BPM. Todo esto dentro de un ambiente distribuido, donde quizás las configuraciones de un servidor parecen estar bien localmente, pero repercuten en la performance total del proceso completo distribuido.

Capítulo 8 - Conclusiones y trabajos futuros

La madurez de los BPMS (Business Process Management Systems) ha hecho que las organizaciones se vuelquen cada vez con más confianza hacia modelos de gestión por procesos de negocio.

La minería de procesos (Process Mining), es una poderosa tecnología para administrar procesos operacionales no triviales, cuyo objetivo es extraer conocimiento de los registros de eventos obtenidos de distintos sistemas de información. Así, constituye un buen soporte para cerrar el ciclo de mejora continua que propone BPM.

Por otra parte Cloud Computing ofrece almacenamiento y cómputo distribuido bajo demanda, permitiendo descomponer un proceso y ejecutar cada una de sus partes de manera distribuida.

Todos estos conceptos ya fueron abordados pero de manera aislada uno del otro. Teníamos el análisis de minería de procesos por un lado y el estudio de ejecución y monitoreo de procesos distribuidos por otro. Como sucede muy a menudo en el plano tecnológico, nuevas tecnologías van apareciendo y adquiriendo la madurez y estabilidad necesaria para ser incorporadas a un sistema ya en producción y dar solución a un problema que hasta ese momento no se podía resolver, o a optimizar la forma, el tiempo y los recursos que este utilizaba. Esto pasó con el cómputo distribuido y la posibilidad de orientar servicios (en nuestro caso BPM) a la nube que, si bien es un concepto que tiene sus años, recién en estos últimos contamos con el software, hardware e infraestructura suficiente para que sea “rentable” la incorporación de esta tecnología a nuestros sistemas. La minería de procesos es una disciplina que ya se viene aplicando hace tiempo y viene dando soporte a todo lo relativo a la mejora continua de los procesos. Pero este análisis supone que el proceso y, por lo tanto, su información de registro, están centralizados en un solo servidor y monitoreados de forma local. Dado este escenario se produce una relación excluyente entre la minería de procesos y el Cloud Computing. La elección de uno implícitamente hacía inviable la utilización del otro. Podíamos tener minería de procesos pero locales o ejecución y monitoreo de procesos distribuidos pero sin la posibilidad de aplicarle técnicas de minería.

La minería de procesos distribuidos fue el desafío que abordó este trabajo, utilizando BonitaOS como BPMS y ProM como herramienta de minería de procesos.

Al contar con dos tecnologías y además herramientas (procesos generador de logs y sistema de ejecución y monitoreo distribuido) que debíamos integrar, comenzamos por plantear el escenario arquitectónico. Por un lado se estudió de qué manera funcionaba el proceso generador de logs y en qué momento se debía ejecutar para que nos sea útil para nuestro análisis. Llegamos a la conclusión que este proceso debía correrse antes de intentar generar cualquier tipo de integración. Esta fue nuestra precondition.

Luego estudiamos cómo realiza las conexiones, ejecuciones y monitoreo la herramienta web. Notamos que utiliza un conector que mediante la API REST de Bonita instanciaba procesos remotos y guardaba la relación proceso local - proceso remoto en una base de datos local.

Este dato fue clave para nuestro planteo de solución porque aquí contamos con la relación que nos permitiría unir cada uno de las instancias del proceso completo.

Teniendo esto como base agregamos una pestaña a la aplicación web de monitoreo para manejar todo lo relativo al registro de eventos. Se decidió orientar la solución a servicios web mediante la librería de PHP NuSOAP. Esto nos permite encapsular y modularizar nuestros métodos sin intervenir en la lógica de la aplicación de monitoreo. Una vez lograda la interacción entre la aplicación y los servicios web, detectamos que podíamos ofrecer dos tipos de opciones de descarga: el log unificado del proceso completo, como comúnmente se trabaja, y un log por cada subprocesso. Esto surgió ya que detectamos la posibilidad que la minería posterior quisiera hacerse tanto desde una perspectiva integral del proceso como a nivel servidor.

Se tuvo que estudiar la estructura de los logs en formato XES para lograr darle la semántica real y que su comportamiento sea reflejado de manera correcta en ProM. Este fue otro desafío ya que debíamos ir buscando en los registros que queríamos unir, donde estaban los identificadores de las instancias (locales y remotas) relacionadas (información que obtenemos de la base de datos de la aplicación de monitoreo) y hacer el uso correcto de los tags y niveles de profundidad que tiene el formato XES para generar un registro correcto. Esto lo logramos gracias a SimpleXML y el manejo de strings en PHP.

Una vez resuelto esto vimos que ProM es una herramienta muy potente para el análisis de procesos. Pudimos importar los registros de eventos generados anteriormente y generar gráficos y diagramas de transición, entre otros, que nos permitieron sacar conclusiones sobre su comportamiento, posibles cuellos de botella y estimar sus tiempos, entre otros.

La minería de procesos es una disciplina que asiste de manera directa a la evolución de los procesos en una empresa y en este trabajo vimos cómo solucionar el tópico de aplicarla en entornos distribuidos. La presente investigación fue presentada en el marco del XVIII Workshop de Investigadores en Ciencias de la Computación del año 2016. [25]

En cuanto a los trabajos futuros, o de la forma en que podría extenderse esta funcionalidad planteada, podemos enunciar:

- Generar un único log del proceso que contenga tanto la información de sus tareas como de sus servidores y en qué lugar está cada una de ellas. Esto nos permitirá en la herramienta de minería poder ver con claridad dónde y qué partes del proceso están distribuidas y no tener que estar importando diferentes archivos de registro de manera aislada.
- Poder pre-filtrar el registro. Como sabemos los archivos de registro pueden ser muy pesados y complejos. Si sabemos de antemano que sólo queremos analizar un periodo de tiempo o a un actor determinado, permitir al servicio web obtener solo ese subconjunto de eventos para optimizar el flujo de datos y el procesamiento de la herramienta de minería.

Referencias bibliográficas

- [1] "Business Process Management: concepts, Languages, Architectures". Mathias Weske Springer-Verlag Berlin Heidelberg 2007
- [2] "Process BPM Basics For Dummies®, Software AG Special Edition". Kiran Garimella Michael Lees Bruce Williams: Published by Wiley Publishing, Inc., Indianapolis, Indiana. 2008
- [3] " Business Process Management Systems " James. F. Chang. Auerbach Publications. 2006.
- [4] "Process Mining Conformance and Extension ". O. by Anne Rozinat. - Eindhoven: Technische Universiteit Eindhoven, 2010. - Proefschrift.
- [5] "Process Mining Discovery, Conformance and Enhancement of Business Processes". Wil M.P. van der Aalst Department Mathematics & Computer Science Eindhoven University of Technology Den Dolech 2 5612 AZ Eindhoven The Netherlands. 2011.
- [6] "Process Mining Project Methodology: Developing a General Approach to Apply Process Mining in Practice". T.H.C. VAN DER HEIJDEN. BSc Industrial Engineering — TU/e 2011 Student identity number 0611037.
- [7] "Business Process Mining: From Theory to Practice". Turner CJ, Tiwari A, Olaiya R, Xu Y. Business Process Management Journal 2012 Vol. 18 Iss:3, pp.493-512
- [8] "Using Process Mining to Bridge the Gap between BI and BPM". Wil M.P. van der Aalst Department Mathematics & Computer Science Eindhoven University of Technology Den Dolech 2 5612 AZ Eindhoven The Netherlands. 2011.
- [9] T. Anstett, F. Leymann, R. Mietzner, and S. Strauch, "Towards bpm in the cloud: Exploiting different delivery models for the execution of business processes," in Proceedings of the 2009 Congress on Services - I. Washington, DC, USA: IEEE Computer Society, 2009, pp. 670–677.
- [10] Xun Xu. "From cloud computing to cloud manufacturing". Department of Mechanical Engineering, University of Auckland. August 2011.
- [11] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology. September 2011.
- [12] Barrie Sosinsky. "Cloud Computing Bible". Wiley Publishing Inc. ISBN: 978-0-470-90356-8. 2011.
- [13] SANDETEL. "Cloud Computing Aplicado a los sectores de la Agroindustria, Eficiencia Energética, Industrias Culturales y Turismo". Noviembre 2012

- [14] C. Baun, M. Kunze, J. Nimis and S. Tai. "Cloud Computing: Web-Based Dynamic IT Services". Springer. ISBN 978-3-642-20916-1. 2011.
- [15] Evert F. Duipmans. "Business Process Management in the cloud with data and activity distribution". Faculty of electrical engineering, mathematics and computer science software engineering. University of Twente. EWI/SE - 2012-002. November 2012.
- [16] Rafael Accorsi. "Business Process as a Service: Chances for Remote Auditing". Department of Telematics, University of Freiburg, Germany. 2011. 35th IEEE Annual Computer Software and Applications Conference Workshops
- [17] Jiri Kolar. "Business Activity Monitoring". Masaryk University Faculty of Informatics. Brno, spring 2009.
- [18] "Aplicación de técnicas de Process Mining para análisis de procesos de negocios desplegados en un BPMS". Alumna Virginia Magliano, para su graduación como Licenciada en Sistemas de la Facultad de Informática de la UNLP. Octubre 2014.
- [19] "Ejecución y monitoreo de procesos de negocios distribuidos". Alumno Leonardo Karabogolian, para su graduación como Licenciado en Sistemas de la Facultad de Informática de la UNLP. Marzo 2015.
- [20] J. Martinez Garro, P. Bazán. "Monitoreo de procesos distribuidos en el cloud: Una propuesta arquitectónica". Jornadas Chilenas de Computación 2013.
- [21] Han YB, Sun JY, Wang GL. "A cloud-based BPM architecture with user-end distribution of non-compute-intensive activities and sensitive data". JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 25(6): 1157–1167 Nov. 2010. DOI 10.1007/s11390-010-1092-5
- [22] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in Proceedings of the 2008 ACM/IEEE conference on Supercomputing, SC '08, (Piscataway, NJ, USA), pp. 50:1–50:12, IEEE Press, 2008.
- [23] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
- [24] "Técnicas de ejecución y monitoreo de procesos en Cloud BPM" [Martinez Garro J., Bazan P.] publicado en XVI Workshop de Investigadores en Ciencia de la Computación 2014: WICC 2014. Usuahia, Tierra del Fuego, Argentina. Mayo 2014. Con referato. ISBN 978-950-34-1084-4
- [25] "Técnicas de minería de procesos de negocio distribuidos con Bonita OS" Federico Gastón Madrid, Mg. Patricia Bazán, Lic. José Martínez Garro. WICC 2016

[26] Process Mining Manifesto -
http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_manifesto

[27] Documentación BonitaSoft - <http://documentation.bonitasoft.com/5x/index.php>

[28] API REST de BonitaSoft -
<http://documentation.bonitasoft.com/java/doc/rest/5.10/API/index.html>

[29] Process Mining org site: <http://www.processmining.org/>

[30] NuSOAP - <http://nusoap.sourceforge.net/>

[31] SimpleXML - <http://php.net/simplexml>