

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3



Algoritmo para la inferencia de redes sociales a partir de la interacción de los recursos en la ejecución de las actividades de un proceso

Tesis presentada para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Leansi Vega Rouco

Tutor: Ing. Anel Wilfredo Fernández Rodríguez

Co-tutor: Dr.C. Damián Pérez Alfonso

La Habana, junio de 2016

“Año 58 de la Revolución”

Declaración de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Leansi Vega Rouco

Autor

Ing. Anel Wilfredo Fernández Rodríguez

Tutor

Resumen

La minería de procesos es una disciplina de investigación que proporciona técnicas para descubrir, monitorear y mejorar los procesos de negocio en las organizaciones. Las mismas dedican tiempo y esfuerzo para analizar sus procesos con el objetivo de mejorar su desempeño organizacional y operacional. En la minería de procesos la inferencia de redes sociales para un proceso global es un elemento importante en el análisis del comportamiento de las interacciones de los recursos. Este análisis se enriquece cuando se realiza a nivel de subproceso, obteniendo información relevante que facilita la comprensión del proceso global. La presente investigación tiene como objetivo principal, desarrollar un algoritmo de inferencia de redes sociales a partir de eventos con alto número de interacciones entre los recursos para facilitar la comprensión de procesos complejos en su dimensión social. Los principales aportes de esta investigación son: el cálculo de los factores popularidad, eficiencia y sobrecarga y la inferencia de redes sociales a nivel de subproceso para mejorar la comprensión del proceso global, un plugin para la herramienta con capacidad de minería de procesos ProM que realiza el algoritmo diseñado.

Palabras claves: Minería de procesos, análisis de redes sociales, comprensión de procesos.

Índice

Introducción	5
1. INFLUENCIA DE REDES SOCIALES EN LA MINERÍA DE PROCESO: BASES CONCEPTUALES.....	10
1.1. Marco teórico	10
1.1.1. Teoría de grafos.....	10
1.1.2. Conceptos básicos de redes sociales.....	11
1.1.3. Conceptos básicos de minería de proceso	12
1.1.4. Técnicas principales de minería de proceso.....	13
1.2. Métricas para la inferencia de redes sociales	15
1.3. Análisis de redes sociales en la minería de proceso	18
1.4. Comprensión del proceso.....	20
1.5. Conclusiones del capítulo.....	21
2. ALGORITMO PARA LA INFERENCIA DE REDES SOCIALES A PARTIR DE LA INTERACCIÓN DE LOS RECURSOS EN LA EJECUCIÓN DE LAS ACTIVIDADES DE UN PROCESO	23
2.1. Base conceptual	24
2.1.1. Análisis de complejidad.....	26
2.2. Aplicación de las técnicas para inferir redes sociales	27
2.2.1. Análisis de complejidad.....	36
2.3. Cálculo de factores para realizar el ARS	39
2.4. Conclusiones del capítulo.....	41
3. VALIDACIÓN DE LA PROPUESTA.....	43
3.1. Realización computacional del algoritmo propuesto.....	43
3.2. Caso de estudio	48
3.3. Cálculo de los factores popularidad, eficiencia y sobrecarga	49
3.4. Conclusiones del capítulo.....	57
Conclusiones generales	58
Recomendaciones	59
Referencias bibliográficas.....	60

Introducción

En la actualidad el avance de las tecnologías de la información y las comunicaciones TIC han propiciado que las grandes y medianas empresas por gestionen sus recursos mediante su informatización, permitiendo un mayor control y un mejor manejo de la empresa, así como el aprovechamiento eficiente del tiempo de sus procesos (1).

Dichas empresas utilizan sistemas de información con el objetivo de gestionar la ejecución de sus procesos de negocios. Para esto se utilizan herramientas informáticas, que registran en forma de trazas toda la información que se generan al ocurrir instancias o casos de dichos procesos. El hecho de que a partir de la información contenida en las trazas se pueda identificar el proceso, se le denomina Minería de Proceso (MP) o minería de flujos de trabajo (2). Con el modelo de procesos basado en trazas se pueden realizar comparaciones entre el modelo teórico y el modelo descubierto y además, es posible apoyar el rediseño de los procesos en la empresa, así como su utilidad como soporte a la operacionalización de los mismos. En los sistemas de hoy en día, mediante la extracción de la información de las trazas existentes, la MP permite descubrir, monitorear y mejorar los procesos reales a través de la extracción de conocimiento de los registros de eventos ampliamente disponibles (3) .

La MP es una disciplina de investigación relativamente joven que se ubica entre la inteligencia computacional y la minería de datos, por una parte, y la modelación y análisis de procesos, por otra. La MP incluye el descubrimiento automático de procesos (extraer modelos de procesos a partir de un registro de eventos), la verificación de conformidad (monitorear desviaciones al comparar el modelo y el registro de eventos), la minería de redes sociales/organizacionales, la construcción automática de modelos de simulación, la extensión de modelos, la reparación de modelos, la predicción de casos y las recomendaciones basadas en historia (4).

La aplicación de MP permite obtener la relación que existe entre los recursos que intervienen en un proceso determinado. Esta información se modela a través de Redes Sociales (RS) y se utiliza posteriormente para realizar Análisis de Redes Sociales (ARS) dentro de un entorno, viéndose con mayor influencia en las empresas a nivel mundial. El ARS es una actividad que surge con necesidad en los procesos de toma de decisiones a diferentes niveles.

Una RS se define como un conjunto de puntos (actores) vinculados por una serie de relaciones que cumplen determinadas propiedades (5), debiendo poseer o estar vinculadas a ciertas particularidades o rasgos comunes con el fin de poder interactuar entre sí (6). Para la representación de las redes sociales generalmente se utilizan métodos formales matemáticos, ya que permiten representar las descripciones de las redes de forma compacta y sistemática (7). Igualmente, las estructuras más conocidas para la representación de sus relaciones, son los grafos y las matrices (8). La teoría de grafos ha sido aceptada como herramienta de análisis de redes y los algoritmos que se utilizan tienen

varios años de uso y aceptación. Es una forma más comprensible de entender el comportamiento de ciertos recursos a la hora de realizar un ARS.

La inferencia de redes sociales constituye un problema de marcado interés en la comunidad científica durante los últimos años. Estudiar el modo en que las personas construyen sus relaciones y cómo estas interacciones desarrollan diversas potencialidades y patrones puede contribuir a explicar la ocurrencia de ciertos fenómenos sociales. Además, estos pueden ser empleados en favor de detectar, estimular, predecir o controlar la magnitud de los comportamientos humanos en diversos contextos y escenarios. El entorno empresarial es uno en el cual las personas interactúan a través de los procesos que ejecutan, donde la gran mayoría de estos procesos son gestionados por sistemas informáticos. Las ejecuciones de estos procesos se registran en dichos sistemas en trazas de eventos. Por lo que resultaría útil inferir las interacciones de los recursos involucrados en los procesos, a partir de las evidencias que existen en las trazas de los eventos de dichos procesos (1).

El análisis a nivel global de las interacciones existentes entre los recursos de un proceso complejo resulta engorroso. De esta forma se dificulta obtener información provechosa que ayude a tener una mejor comprensión del proceso que se esté analizando (9). Conocer los recursos con más probabilidad de extender el proceso en un futuro es uno de los factores que se ve afectado en el análisis de un proceso global, así como determinar los recursos indispensables en el proceso. Debido a la alta complejidad de los procesos tratados se hace necesario tener en cuenta los recursos y los flujos de información que se pueden optimizar en la ejecución del proceso. Además, en el ARS a nivel global se dificulta tener información relevante de los recursos del proceso que están sobrecargados. La inferencia de redes sociales a partir de registros de eventos constituye un criterio que aporta información relevante para la comprensión de procesos complejos (9).

A partir de la siguiente situación descrita se define el siguiente **problema a resolver**:

- El alto número de interacciones entre los recursos que ocurre en procesos complejos dificulta la comprensión de los mismos en su dimensión social.

El problema definido se enmarca en el siguiente **objeto de estudio**: Minería de Proceso.

El **objetivo general** de la investigación es desarrollar un algoritmo de inferencia de redes sociales a partir de eventos con alto número de interacciones entre los recursos para facilitar la comprensión de procesos complejos en su dimensión social.

El **campo de acción** es la inferencia de redes sociales.

El objetivo general se desglosa en los siguientes **objetivos específicos**:

- Construir el marco teórico referencial asociado a la minería de proceso para la inferencia de redes sociales.

- Diseñar un algoritmo que posibilite la inferencia de redes sociales a partir de las interacciones de los recursos en la ejecución de las actividades de un proceso.
- Desarrollar un plugin a partir del algoritmo diseñado para la inferencia de redes sociales teniendo en cuenta las interacciones de los recursos en la ejecución de las actividades de un proceso.
- Validar el algoritmo implementado a partir de un caso de estudio.

A partir de un análisis preliminar se enuncia la siguiente **hipótesis**: si se desarrolla un algoritmo de inferencia de redes sociales a partir de eventos con alto número de interacciones entre los recursos, entonces se facilitará la comprensión de procesos complejos en su dimensión social.

Para dar cumplimiento a los objetivos específicos trazados se hizo uso de los siguientes **métodos de investigación**:

- Métodos teóricos:
 - Análisis-síntesis: para descomponer el problema de investigación en elementos por separado, de esta forma se puede profundizar en el estudio de cada elemento, para luego sintetizarlos en la solución de la propuesta.
 - Hipotético-deductivo: para elaborar la hipótesis de investigación y proponer líneas de trabajo a partir de resultados parciales.
 - Histórico-lógico: para llevar a cabo el estudio crítico de las principales tendencias del análisis de interacciones e influencia en las redes sociales, patrones de comportamiento en el ámbito universitario, así como de los algoritmos para calcular el valor de dichos patrones.
 - Sistémico: para la implementación de la herramienta computacional y de los algoritmos propuestos.
- Métodos empíricos:
 - Medición: para la obtención de información numérica que aporten valores probabilísticos para la información.

Con la realización de la investigación se **espera** el desarrollo de un algoritmo para la inferencia de redes sociales a partir de las interacciones de los recursos en la ejecución de las actividades de un proceso.

La **estructura de la tesis** se expone haciendo uso de una perspectiva en forma de anillo como se muestra en la Figura 1 enfocándose a su objeto de estudio: la inferencia de redes sociales. Esta perspectiva garantiza la consistencia estructural de la investigación.

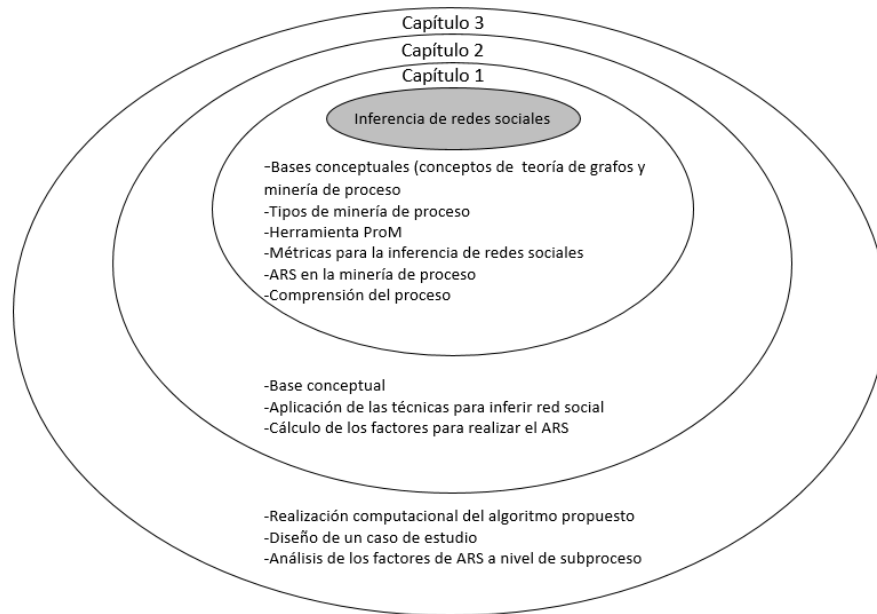


Figura 1: Estructura de la tesis.

Capítulo 1

Influencia de redes sociales en la minería de proceso: Bases conceptuales

1. INFLUENCIA DE REDES SOCIALES EN LA MINERÍA DE PROCESO: BASES CONCEPTUALES

En este capítulo se expone el marco conceptual además de realizar un estudio enfocado a la influencia de redes sociales en la minería de procesos y su estrecha relación.

1.1. Marco teórico

En la presente sección se exponen conceptos relacionados con la teoría de grafos, teoría de redes sociales y minería de proceso.

1.1.1. Teoría de grafos

En este epígrafe se presentan definiciones y notaciones utilizadas en los algoritmos que se proponen. Las definiciones que siguen son tomadas de (10).

DEFINICIÓN 1.1.1.1 *Un grafo, o grafo no dirigido, $G = (V, E)$ se define como un conjunto V finito y no vacío de vértices y un multiconjunto E de aristas, donde cada arista $(v_i, v_j) \in E$, $v_i, v_j \in V$ es un par no ordenado de vértices. Opcionalmente una arista puede tener un valor que la identifique y una lista de atributos. Cuando los elementos de E tienen multiplicidad uno, el grafo se denomina grafo simple.*

DEFINICIÓN 1.1.1.2 *Un vértice v_j es adyacente (vecino) a otro vértice v_i si $\exists (v_i, v_j) \in E$. Además, se dice que la arista (v_i, v_j) incide en el vértice v_j .*

DEFINICIÓN 1.1.1.3 *Se denomina grado de un vértice v a la cantidad de aristas que inciden en v y se representa como $g(v)$.*

DEFINICIÓN 1.1.1.4 *En un grafo dirigido, se denomina grado de entrada de un vértice v a la cantidad de aristas que tienen como vértice final a v y se representa como $g^-(v)$. De forma similar, se denomina grado de salida a la cantidad de aristas que tienen como vértice inicial a v y se representa como $g^+(v)$.*

DEFINICIÓN 1.1.1.5 *Se define un grafo ponderado como una estructura $G = (V, E, fc)$, donde:*

- V representa el conjunto de vértices del grafo.
- E representa un multiconjunto de aristas del grafo.
- La función $fc: E \rightarrow R^+$ le hace corresponder a cada arista un valor real positivo denominado costo.

DEFINICIÓN 1.1.1.6 *Se denomina camino desde el vértice v_i al vértice v_j en un grafo $G = (V, E)$ a la secuencia de vértices $CA = v_{k1}, v_{k2}, \dots, v_{kt}$ si $\exists (v_{k1}, v_{k2}), (v_{k2}, v_{k3}) \dots, (v_{kt-1}, v_{kt}) \in E$, $v_i = v_{k1}$, $v_j = v_{kt}$. Además, una arista solo puede aparecer una vez.*

DEFINICIÓN 1.1.1.7 Se denomina longitud de un camino $CA = v_{k1}, v_{k2}, \dots, v_{kt}$, $t > 1$, a la suma de los costos de todas las aristas presentes en el mismo y se representa como $|CA| = \sum_{h=1}^{t-1} C k_h k_{h+1}$. En el caso de grafos no ponderados, la longitud del camino se puede calcular como la cantidad de aristas presentes en el mismo, o sea $|CA| = t - 1$.

1.1.2. Conceptos básicos de redes sociales

Los conceptos que se enuncian a continuación son tomados de (11).

DEFINICIÓN 1.1.2.1 Una red social se define como un conjunto finito de actores que tienen un determinado número de relaciones definidas entre ellos. Una red social es modelada como un grafo $G = (V, E)$. En la presente investigación, el conjunto de vértices V se corresponde a los actores y el conjunto de aristas E a las relaciones sociales.

DEFINICIÓN 1.1.2.2 Se denomina actor a la entidad social que interactúa con otras entidades. Esta puede ser una persona, un grupo de individuos, una organización, una ciudad, una nación, entre otros.

DEFINICIÓN 1.1.2.3 Un vínculo o conexión corresponde al tipo de relación que conecta un par de actores distintos. Estos vínculos pueden ser de naturaleza diferente: relaciones transnacionales, de comunicación, instrumentales, sentimentales, de poder o autoridad, de parentesco, entre otros.

DEFINICIÓN 1.1.2.4 Una díada se compone de un par de actores y la posible relación existente entre ellos. Varios estudios parten de la díada como unidad fundamental, analizando para cada par de actores, si el vínculo es recíproco o no.

DEFINICIÓN 1.1.2.5 Se denomina tríada al conjunto de tres actores y las posibles relaciones existentes entre ellos. Una propiedad interesante de una tríada es la existencia o no, de transitividad entre sus miembros. Es decir, comprobar que, si el actor i tiene una relación con el actor j y a su vez el actor j tiene una relación con el actor k , entonces el actor i también tiene una relación con el actor k .

La centralidad basada en vértices se define con el fin de medir la importancia de un vértice en la red. Esta ha llamado mucho la atención como una herramienta para el estudio de las redes sociales. Un vértice con alta centralidad suele ser considerado más altamente influenciado que otros vértices de la red (12, 13).

DEFINICIÓN 1.1.2.6 Sea A la matriz de adyacencia de un grafo G , y $g(v_i)$ el grado del vértice v_i , entonces el grado de centralidad c_i^{DEG} del vértice v_i se define:

$$c_i^{DEG} = g(v_i)$$

Una manera de interpretar el grado de centralidad de un vértice v_i es contar el número de caminos de longitud 1 que terminan en dicho vértice. Una generalización natural de este punto de vista es la

centralidad camino – K que es el número de caminos de longitud menor o igual que K que terminan en el vértice v_i .

DEFINICIÓN 1.1.2.7 La medida de cercanía, definida por el matemático Murray Beauchamp en 1965 (14) y luego popularizada por Freeman en 1979 (15), es la más conocida y utilizada de las medidas radiales de longitud. Se basa en calcular el promedio de las distancias más cortas desde un vértice hacia todos los demás. Formalmente, la cercanía se define como:

$$c_i^{CLO} = e_i^T S,$$

donde $e_i^T S$ es la transpuesta de un vector columna cuyo i -ésimo elemento es 1 y el resto son 0, S es la matriz cuyos elementos en la posición (i, j) corresponden a la distancia más corta desde el vértice v_i al vértice v_j .

DEFINICIÓN 1.1.2.8 La intermediación es una medida que cuantifica la frecuencia o el número de veces que un vértice actúa como un puente a lo largo del camino más corto entre otros dos vértices. De manera formal, la intermediación se define como:

$$c_i^{BET} = \sum_{j,k} (b_{jik} / b_{jk}),$$

donde b_{jk} es la cantidad de caminos mínimos desde v_j a v_k y b_{jik} la cantidad de caminos mínimos desde v_j a v_k que pasan por v_i .

1.1.3. Conceptos básicos de minería de proceso

En esta sección se exponen un conjunto de conceptos fundamentales en la investigación que posibilitan el estudio de la teoría de la minería de proceso.

DEFINICIÓN 1.1.3.1 Un proceso es el conjunto de actividades o tareas mutuamente relacionadas entre sí, que admite elementos de entrada durante su desarrollo, ya sea al inicio o a lo largo del mismo, los cuales se administran, regulan o autorregulan bajo modelos de gestión particulares para obtener elementos de salida o resultados esperados (16).

DEFINICIÓN 1.1.3.2 Un proceso de negocio como una colección de actividades que son realizadas coordinadamente en un ambiente técnico y organizacional. La conjunción de estas actividades logra un objetivo del negocio. Cada proceso de negocio es ejecutado por una simple organización, pero con él pueden interactuar procesos de negocios de otras organizaciones (17).

DEFINICIÓN 1.1.3.3 Un subproceso es una encapsulación de las actividades del negocio que representa una unidad de trabajo lógica y coherente. Los subprocesos tienen sus propios atributos y metas, pero contribuyen a alcanzar la meta general del proceso. Un subproceso es también un proceso y la mínima expresión de un subproceso es una actividad (9, 23).

Un proceso puede descomponerse en varios subprocesos mediante algunos de los patrones de flujo de control conocidos como: secuencia, paralelismo, selección exclusiva y lazos (9).

DEFINICIÓN 1.1.3.4 *El modelado de procesos de negocio es el modelo de uno o más procesos de negocio de una empresa y define las formas en que las operaciones se llevan a cabo para lograr los objetivos deseados. Tiene como objetivo ilustrar un proceso completo, permitiendo a los analistas, gerentes, consultores y otro personal, analizar y mejorar la eficiencia y calidad de los procesos (18).*

Curtis, afirma que existen cuatro puntos de vista en cuanto al modelado de los procesos de negocio: vista funcional (qué), la cual representa la dependencia funcional entre los elementos del proceso, vista dinámica (cuándo, cómo), que proporciona una secuenciación y control de la información sobre el proceso, vista informacional, que incluye la descripción y relación entre las entidades que son producidas, consumidas o incluso manipuladas por los procesos y la vista organizacional (quién, dónde) que describe quién desarrolla cada tarea o función y dónde se desarrolla dentro de la organización (19)

DEFINICIÓN 1.1.3.5 *Se conoce como minería de proceso al conjunto de técnicas que permiten el análisis de diferentes procesos de negocio, basándose en los registros de información proporcionados por sistemas de información transaccionales, de administración de flujo de trabajo (WFM por sus siglas en inglés), de Planificación de los Recursos Empresariales (ERP por sus siglas en inglés), de Gestión de la Relación con los Clientes (CRM por sus siglas en inglés), de Gestión de Cadenas de Abastecimiento (SCM por sus siglas en inglés), etc.*

La minería tiene como objetivo descubrir, monitorear y mejorar los procesos reales (no los procesos supuestos) a través de la extracción de conocimiento de los registros de eventos ampliamente disponibles en los sistemas de información (20) .

DEFINICIÓN 1.1.3.6 *Una actividad es el conjunto de acciones que se llevan a cabo para cumplir las metas de un programa o subprograma de operación, que consiste en la ejecución de ciertos procesos.*

DEFINICIÓN 1.1.3.7 *Una traza es el seguimiento del estado de la memoria y de la entrada y salida de un programa según se ejecuta el mismo.*

DEFINICIÓN 1.1.3.8 *Un log de eventos es un archivo especial que registra los **eventos** importantes que tienen lugar en el equipo.*

1.1.4. Técnicas principales de minería de proceso

Las técnicas de MP posibilitan, hacer recomendaciones y predicciones, teniendo en cuenta el análisis de los datos actuales y de los históricos. Estas se pueden clasificar en tres grupos. 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 proceso 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. La Fig. 1.1 muestra la ubicación de las tres principales técnicas de minería de proceso(20).

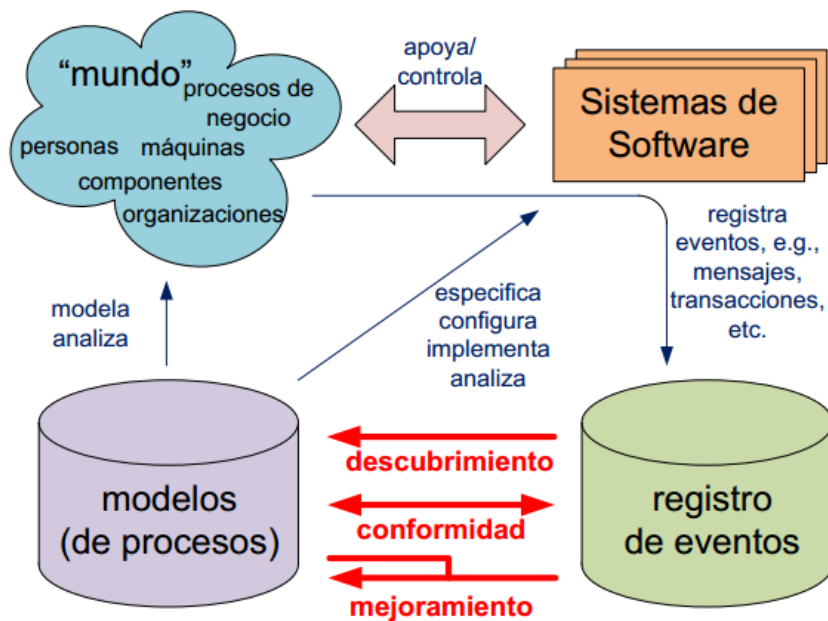


Figura 1.1. Posicionamiento de los tres tipos principales de minería de procesos: (a) descubrimiento, (b) verificación de conformidad, y (c) mejoramiento. Tomado de (20).

La Fig. 1.2 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 (20).

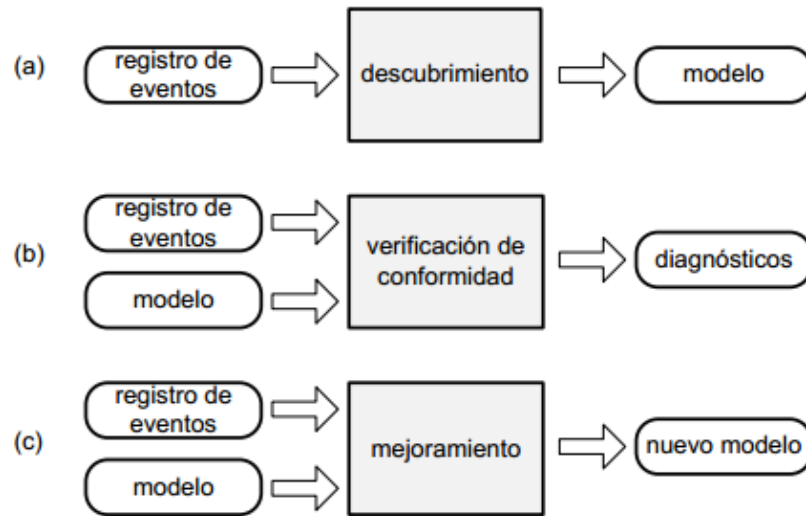


Figura 1.2. Los tres tipos básicos de minería de procesos explicados en términos de entradas y salidas: (a) descubrimiento, (b) verificación de conformidad, y (c) mejoramiento. Tomado de (20).

Las diferentes técnicas de la MP permiten el descubrimiento de información desde los registros de eventos, permitiendo conocer el flujo de ejecución de un proceso, las redes sociales y sus métricas de desempeño. De esta manera se puede aprovechar todos los datos de eventos registrados de forma significativa para proveer un mejor entendimiento, identificar cuellos de botella, prever problemas, registrar violaciones de políticas, recomendar contramedidas y simplificar procesos. Además, disminuir los tiempos de diseño y con ello sus costos.

1.2. Métricas para la inferencia de redes sociales

Varios estudios realizados (21, 22) muestran que el uso de las tecnologías de la información y la comunicación en distintos dominios genera gran cantidad de datos. Estos datos contienen información sobre los perfiles de los actores de dichos dominios, así como su comportamiento. La información derivada puede ser utilizada para comprender los procesos que se llevan a cabo y los factores que influyen en su rendimiento.

El análisis anterior se realiza a partir de técnicas que tienen como objetivo inferir las redes sociales a partir de los registros de eventos basados en las interacciones observadas entre los actores y en función de cómo las instancias de proceso se enrutan entre estos actores (20, 23). Estas interacciones pueden obtenerse tras una de estas cinco métricas: Handover of work, Subcontracting, Working

together, Similar task y Reassignment. Los plugins de minería social en la herramienta ProM generan gráficos donde cada nodo representa un recurso.

La idea principal de estas métricas es de supervisar cómo las instancias de proceso individual se enruta entre los actores. Estas técnicas permiten inferir redes sociales usando una de las cuatro métricas que existen para realizar este tipo de minería. Estas métricas se pueden describir de la siguiente manera (23):

- *Métricas basadas en la causalidad* monitorean para casos individuales como el trabajo se mueve entre los actores. Ejemplos de tales parámetros son: *Handover of work* y *Subcontracting*. Vamos a explicar de una manera corta, que la información de los registros de eventos se utiliza en esta métrica. Consideraremos la métrica *Handover of work* y el registro de eventos se representa por Tabla 1.1 *Handover of work* determina quién da trabajo a quién, y del registro de eventos esta información puede ser extraída de dos actividades posteriores en el mismo caso. Por ejemplo, en case1 María se inicia y termina la actividad, siguiente en el mismo caso, John inicia y termina la actividad. Así podemos asumir que María ha delegado o pasado el trabajo a John.
- *Métricas basadas en casos conjuntos* cuentan con qué frecuencia dos individuos están realizando actividades para el mismo caso. Ejemplo de esto son *Working together*. Vamos a explicar de una manera corta que la información, a partir de los registros de eventos, se utiliza para hacer el análisis de *Working together*. Por ejemplo, el registro de eventos se representa en la Tabla 1.1 consideraremos caso 1a (Trace 1a, María) y el evento 1b (Trace 1b, John). María y Juan, a pesar de la realización de diferentes actividades, realizan actividades en el mismo caso, por lo tanto, podemos suponer que trabajan juntos.
- *Métricas basadas en actividades conjuntas* no tienen en cuenta cómo las personas trabajan juntas en los casos compartidos, pero se centran en las actividades que realizan. Un ejemplo de la aplicación de esta métrica es *Similar task*. Vamos a explicar de una manera corta, que la información de los registros de eventos se utiliza para hacer análisis de tareas similares. Cada actor tiene un perfil que almacena la frecuencia con la que el actor ejecuta cada tarea. Esta métrica determina la similitud de dos actores basados en la similitud de su perfil. Por ejemplo, en el registro de eventos representado por la Tabla 1.1 podemos observar que María sólo realiza actividades de tipo A, John sólo realiza actividades de tipo B y Ángela sólo realiza actividades de tipo A. Así que, de acuerdo a esto, ya que María y Ángela realizan el mismo tipo de actividades, son más similares que María y Juan, que tiene completamente diferentes perfiles.
- *Métricas basadas en los tipos de eventos especiales* consideran el tipo de evento. Con el uso de estas métricas obtenemos observaciones que son de especial interés para el análisis de

redes sociales, ya que representan relaciones jerárquicas explícitas. Un ejemplo de la aplicación de esta métrica es *Reassignment*.

Traza	Evento	Propiedades			
		Actividad	Recurso	Timestamp	Tipo
1	1a	A	Mary	20-11-2007 08:00	start
	1b	A	Mary	20-11-2007 08:13	complete
	1c	B	John	20-11-2007 08:16	start
	1d	B	John	20-11-2007 08:40	complete
2	2a	A	Ángela	20-11-2007 09:30	start

Tabla 1.1. Registro de eventos para describir las métricas de inferencia de redes sociales. Tomado de (23).

Las métricas proporcionan cinco tipos de técnicas para generar redes sociales. Son *Handover of work*, *subcontracting*, *working together*, *similar task* y *reassignment*.

- *Handover of work*: Este indicador determina quien le pasa trabajo a quién. Esta información puede ser extraída de un registro de eventos encontrando actividades posteriores en el mismo caso (es decir, la instancia de proceso), donde la primera actividad se completa con un individuo y la segunda se completa con otro individuo.
- *Subcontracting*: Esta medida es similar a handover of work. Mientras en la relación anterior entre dos individuos es unidireccional, en éste es bidireccional. Teniendo en cuenta un solo caso de un registro de evento y dos individuos, saber que el individuo i subcontrata individuo j, cuando se encuentra en el intervalo entre dos actividades ejecutado por el individuo i hay una actividad ejecutada por el individuo j.
- *Working together*: Dos individuos trabajan juntos si llevan a cabo actividades en el mismo caso de un registro de eventos. Esta técnica sólo cuenta con qué frecuencia los individuos trabajan en el mismo caso.
- *Similar task*: Todas las técnicas anteriores se basan en los casos conjuntos, éste es basado en actividades conjuntas. La idea principal es determinar quién realiza el mismo tipo de actividades. Para ello, cada individuo tiene su propio perfil en función de la frecuencia que llevan a cabo actividades específicas. A continuación, los perfiles se comparan para determinar la semejanza. Un ejemplo de esta técnica se muestra en la figura 3:
- *Reassignment*: La idea básica de esta métrica es detectar la asignación de actividades de un individuo a otro: si i frecuentemente delega trabajo a j, pero no vice versa es probable que i sea jerárquicamente más alto que j.

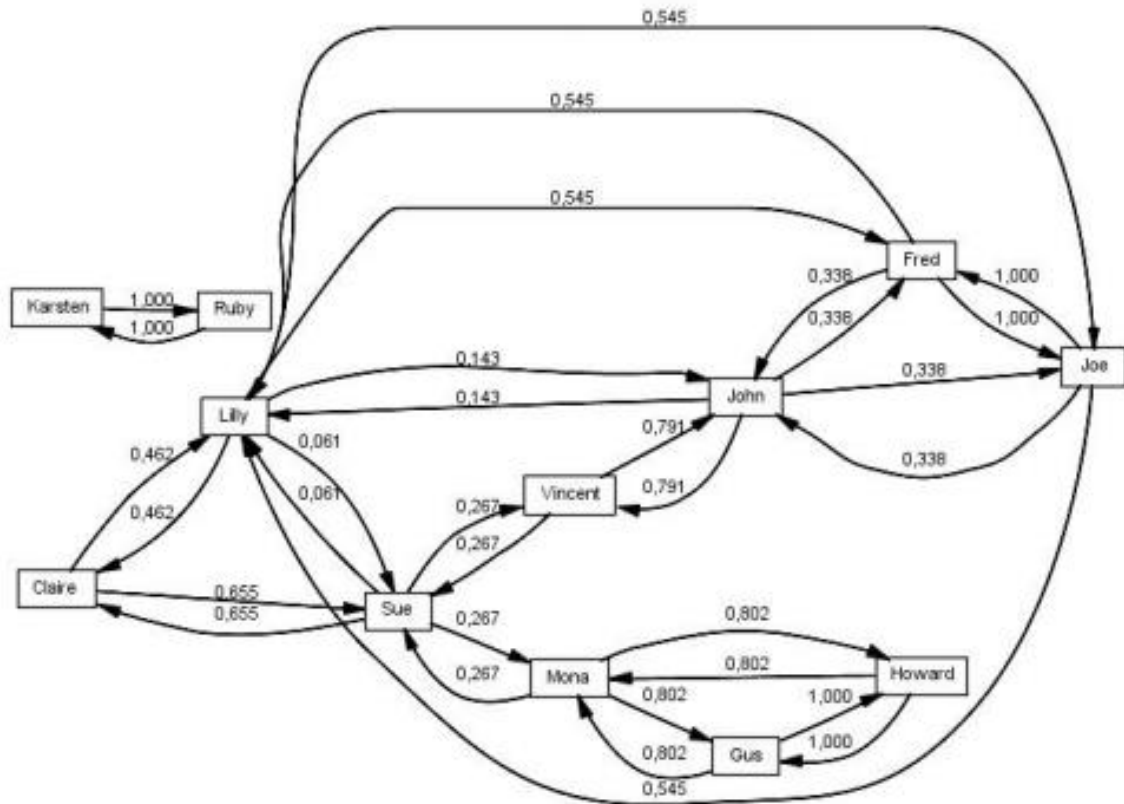


Figura 1.3: Ejemplo de red social partiendo de la técnica similar task. Tomado de (23).

1.3. Análisis de redes sociales en la minería de proceso

Análisis de Redes Sociales (ARS) se refiere a la colección de métodos, técnicas y herramientas en la sociometría con vistas al análisis de redes sociales (24–26). Hay varios indicadores que se presentan en Aalst, Reijers y Song (5). En este contexto, basado en la causalidad (manejo de casos a otros actores), casos conjuntos (frecuencia de dos individuos están realizando actividades para el mismo caso), actividades conjuntas (frecuencia de los usuarios que realizan actividades específicas), eventos especiales, cambio de destino, horarios, etc. Estas métricas permiten derivar sociogramas que a su vez permiten la aplicación de técnicas de ARS, tales como intermediación, cercanía y centralidad.

Song y Aalst añaden, que después de generar una red social, diversas técnicas de ARS tales como la densidad, centralidad, la cohesión, la equivalencia, etc. se pueden aplicar. Por ejemplo, la intermediación se puede utilizar para encontrar posibles sobrecarga de recursos (27) .

En las redes sociales generados por la aplicación de *Handover of work*, los nodos sin arcos entrantes son originadores que solo inician los procesos, mientras que los nodos sin arcos salientes son creadores que realizan sólo las actividades finales. En las redes sociales generados por la aplicación de *Subcontracting*, el nodo de inicio de un arco representa un contratista y el nodo final subcontratista. Por lo tanto, los nodos con un alto grado de centralidad de salida son originadores que suele

desempeñar el papel de los contratistas y los nodos con un alto grado de centralidad de entrada originadores que por lo general actúan como subcontratistas. En las redes sociales generada por la aplicación de métricas basadas en casos conjuntos, alta densidad significa que más originadores trabajan juntos.

Otros autores (28) hacen referencia a las medidas de centralidad aplicables en la teoría de grafos (centralidad, intermediación y cercanía) para el análisis de redes sociales. Estas medidas son aplicadas luego de tener creada la red. Para crear la red se utilizan las métricas antes vistas que infieren una red social bajo diferentes criterios. Exponen que el grado de centralidad identifica los nodos más activos en la red. Un nodo con alto grado de centralidad mantiene gran número de enlaces a otros. Como tal, un nodo central ocupa una posición estructural (ubicación de red) que puede actuar como un conducto para el intercambio de información (28).

La intermediación es una medida para determinar en qué medida un nodo tiene control sobre la información que fluye entre otros nodos. Los nodos con alta centralidad de intermediación pueden tender un puente sobre dos nodos no adyacentes y pueden tener un impacto en el flujo de datos entre las comunidades. Por lo tanto, la intermediación es un indicador clave para determinar los enlaces entre las comunidades (29).

Un nodo con la mayor la cercanía significa que el nodo tiene el camino más corto a otros nodos en la red. Así la cercanía describe la eficacia de la propagación de información de un nodo a todos los demás. En las aplicaciones de reenvío de mensajes y de difusión de datos, la cercanía se puede utilizar para seleccionar nodos de retransmisión para entregar el mensaje con éxito y / o más rápido que los demás nodos de la comunidad (30).

A partir de lo antes expuesto en este acápite se puede decir que en el ARS en la minería de proceso los términos de cercanía, centralidad e intermediación son claves para el desarrollo de esta investigación. Así mismo sucede con las métricas para inferir redes sociales. Pero sería más provechoso y aportaría más información a la hora de realizar el ARS si se complementa la relación que existe entre estas métricas y dichas medidas. Por lo que se puede arribar a:

- La medida de *centralidad* permite determinar el o los nodos más populares de la red. De esta forma se puede determinar cuáles son los recursos indispensables del proceso y los que en un futuro tienen más probabilidad de extender el modelo de proceso.
- La medida de *intermediación* permite encontrar sobrecargar en los recursos del proceso. Esta medida es muy importante debido a que a través de su aplicación se pueden determinar los recursos que pudiesen inferir en un posible cuello de botella en la ejecución del proceso.
- La medida de *cercanía* permite encontrar el camino mínimo de un nodo hacia todos los demás. Esta medida es de gran importancia a la hora de calcular la eficiencia de los flujos de información

entre los recursos del proceso. Conociendo la eficiencia en la ejecución de las actividades del proceso se puede lograr la optimización en el flujo de información y en los recursos del proceso.

1.4. Comprensión del proceso

La comprensión del proceso parte de la comprensión de sus representaciones (31). Estas representaciones son los modelos asociados a los procesos, los cuales poseen un conjunto de términos que se deben dominar para la comprensión del proceso desde su modelo (32–37). Además, la comprensión del proceso puede realizarse también desde la MP a partir de la realización de una comparación entre las trazas de un registro de eventos y el modelo de procesos (20).

En el estudio realizado por (9), se propone un conjunto de características que deben poseer las herramientas de MP para contribuir a la comprensión del proceso en la fase de diagnóstico. Esta propuesta se basa en la combinación de los objetivos del diagnóstico y los factores que influyen en la comprensión del proceso, ya sean relativos a las características del modelo o a su alineación con la ejecución real del mismo.

La primera característica propuesta es la vista jerárquica del proceso, la cual permite su análisis por niveles de abstracción y la ubicación en contexto de la ejecución de las actividades y los patrones interesantes. Esta vista se logra mediante la modularización o separación en subprocesos, lo cual facilita el análisis del proceso por partes y disminuye la carga cognitiva extrínseca (38), mejorando la comprensión (9).

Para disminuir la complejidad del modelo es importante obtener una vista global del proceso. Esta vista debe facilitar la abstracción de los patrones específicos concentrándose en los comportamientos y patrones generales y de más alto nivel. De esta manera el análisis puede realizarse de lo general a lo particular, de los patrones comunes a las anomalías, asimilando paulatinamente el comportamiento del proceso (9).

El control o tratamiento del ruido influye en la estructura del modelo construido y por ende en las dimensiones *Aptitud* y *Precisión* (39). El ruido descartado del modelo afecta su *Aptitud*, mientras que incluirlo puede disminuir la *Precisión* (40). Estas dimensiones expresan la veracidad del modelo, la cual afecta el nivel semántico de su comprensión. Visualizar el ruido suele conducir a modelos con muchos nodos y alta densidad y por ende complejos. Sin embargo, ocultar los comportamientos poco frecuentes dificulta la identificación de las desviaciones y anomalías, un objetivo clave del diagnóstico, lo cual limita la comprensión a nivel pragmático (9).

La *Generalización* de un modelo depende del tratamiento o control de la ausencia de información (39). Un modelo ajustado únicamente al comportamiento reflejado en el registro de eventos afecta su veracidad y puede conducir a interpretaciones erróneas sobre la ejecución del proceso, lo que afecta la comprensión a nivel pragmático. Esto también puede suceder cuando en el modelo se considera la

ausencia de información, pero no se visualizan los comportamientos ausentes que fueron estimados. El tratamiento de la ausencia de información define además la inserción de actividades invisibles, las cuales aumentan la complejidad del modelo (9).

Las características relacionadas con el control de ruido y la ausencia de información, ambas se controlan en el modelo desarrollado por (9). Por tanto, la presente investigación enfoca su atención en la comprensión del proceso haciendo uso de las características de vista jerárquica del proceso (descomposición del proceso global en subprocesos) y vista global del proceso (ver el proceso de lo general a lo particular).

La descomposición de un proceso, con un alto número de interacciones entre los recursos, en subprocesos permite obtener información útil para su análisis. Por tanto, se hace necesario un algoritmo de inferencia de redes sociales que mejore la comprensión de dichos procesos complejos en su dimensión social.

1.5. Conclusiones del capítulo

Luego de realizar un estudio de los conceptos básicos del análisis de redes y minería de procesos, las métricas para inferir redes sociales, las medidas de centralidad en las redes y la comprensión de los procesos, se concluye lo siguiente:

- Las métricas utilizadas para inferir redes sociales posibilitan describir el comportamiento de los recursos del proceso, así como mostrar las relaciones existentes entre ellos.
- El cálculo de las medidas de centralidad en redes sociales permite realizar análisis del comportamiento de los recursos en un proceso determinado; sin embargo, la comprensión de dicho proceso se ve dificultada cuando existe un alto número de interacciones sus recursos, provocando que el análisis de dichas métricas también se vea afectado.
- Existe la necesidad de un algoritmo para inferir redes sociales en subprocesos generados a partir de procesos complejos, que mejore su comprensión su dimensión social.

Capítulo 2

Algoritmo para la inferencia de redes sociales a partir de la interacción de los recursos en la ejecución de las actividades de un proceso

2. ALGORITMO PARA LA INFERENCIA DE REDES SOCIALES A PARTIR DE LA INTERACCIÓN DE LOS RECURSOS EN LA EJECUCIÓN DE LAS ACTIVIDADES DE UN PROCESO

En este capítulo se presenta un algoritmo para la inferencia de redes sociales a partir de un subproceso y aplicando métricas establecidas. Haciendo uso del mismo se calculan tres factores fundamentales en el ARS: popularidad, eficiencia y sobrecarga. En la Figura 2.1 se muestra un esquema que representa los componentes del algoritmo que se propone.

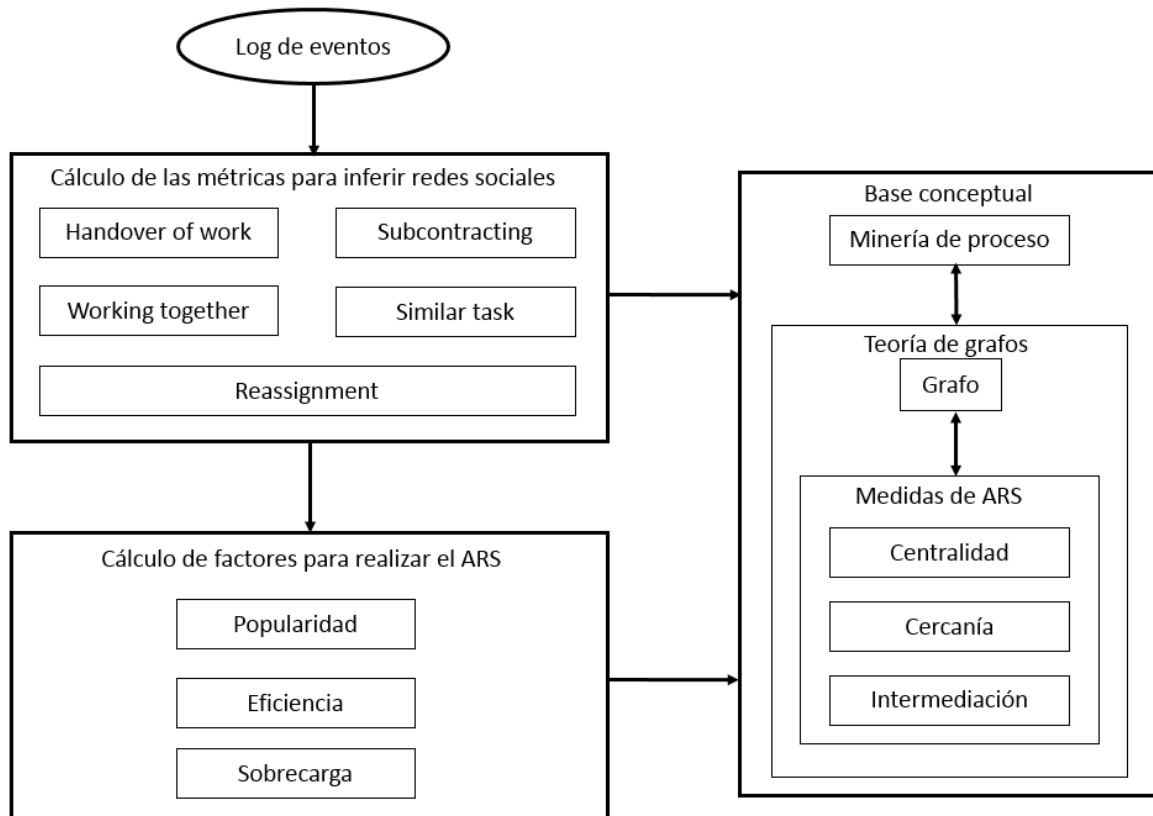


Figura 2.1: Componentes del algoritmo propuesto.

A continuación, se expone en detalle cada uno de los componentes de este algoritmo, así como las relaciones existentes entre los mismos.

El algoritmo propuesto recibe como entrada un log de eventos que representa un subproceso del proceso global. Dicho log de eventos está compuesto por trazas y estas a su vez por eventos. Es importante para la aplicación de las medidas expuesta tener logs de eventos completos y con todos valores de los atributos.

Luego se procede al cálculo de métricas para inferir las redes sociales bajo diferentes criterios establecidos por dichas métricas: *handover of wok*, *subcontracting*, *working together*, *similar task* y *reassignment*.

Las métricas establecidas para crear redes sociales son atributos que distinguen el comportamiento de los recursos dentro de un proceso y la relación que tienen entre ellos. Las redes sociales que se obtienen con la aplicación de dichas métricas son la entrada para realizar el ARS. A partir de estos criterios se calculan tres factores muy importantes en el ARS: centralidad, cercanía e intermediación. Estas tres medidas, como bien se exponen en el capítulo 1 son factores imprescindibles para determinar popularidad, sobrecarga y cuellos de botella en la red.

Finalmente se obtiene como resultado la visualización de redes sociales inferidas por los logs de eventos que se recibe como entrada, donde cada log de eventos se refiere a un subproceso diferente. Además, con la aplicación de las medidas de centralidad se obtiene la popularidad y eficiencia de los recursos, así como posibles sobrecargas de los mismos.

2.1. Base conceptual

En este epígrafe se enuncian varios conceptos referidos en la literatura. Se utiliza el concepto de grafo para modelar una red social. Haciendo uso de éste, se describen varias medidas de ARS definidas en el capítulo anterior, las cuales serán utilizadas en los demás componentes del modelo y se enuncian a continuación:

- Centralidad
- Cercanía
- Intermediación

Estas medidas de centralidad se utilizarán para calcular los factores popularidad, eficiencia y sobrecarga de recursos a partir el algoritmo propuesto.

Para el cálculo de dichas medidas de centralidad, se diseñaron los siguientes algoritmos 1, 2 y 3.

- **Algoritmo 1** *CalcularCentralidad*

Entrada: Un vértice v_i

Salida: Grado de centralidad C_i^{DET} para el vértice v_i

1: $C_i^{DET} \leftarrow -1$

2: **Si** $v_i \in V$ **entonces**

3: $C_i^{DET} \leftarrow \text{grado}(v_i)$

4: **Fin Si**

5: **Retornar** C_i^{DET}

- **Algoritmo 2** *CalcularCercanía*

Entrada: Un vértice v_i , el vector D_n de distancias mínimas de v_i a los demás vértices

Salida: La cercanía C_i^{CLO} para el vértice v_i

1: $C_i^{CLO} \leftarrow 0$


```

2: Si  $v_i \in V$  entonces
3:    $sum \leftarrow 0$ 
4:   Para todo  $w \in V$  hacer
5:      $sum \leftarrow sum + D_n[w]$ 
6:   Fin Para
7:    $C_i^{CLO} \leftarrow \frac{sum}{Longitud(D_n)}$ 
8: Fin Si
9: Retornar  $C_i^{CLO}$ 

```

Sea $P[v]$ el conjunto de predecesores en los caminos mínimos que terminan en el vértice v , $\sigma[v]$ la cantidad de caminos mínimos que terminan en el vértice v y $d[v]$ la longitud de los caminos mínimos que terminan en v .

- **Algoritmo 3** *Calcular Intermediación*. Tomado de (41)

Entrada: Un grafo G

Salida: Vector de intermediación C_i^{BET} para los vértices del grafo G

```

1:  $C_i^{BET}[v] \leftarrow 0, v \in V$ 
2: Para todo  $v \in V$  hacer
3:    $P[v] \leftarrow Lista()$ 
4:    $\sigma[v] \leftarrow 0$ 
5:    $d[v] \leftarrow -1$ 
6: Fin Para
7: Para todo  $s \in V$  hacer
8:    $S \leftarrow Pila()$ 
9:    $\sigma[s] \leftarrow 1$ 
10:   $d[s] \leftarrow 0$ 
11:   $Q \leftarrow Cola()$ 
12:  Encolar( $s, Q$ )
13:  Mientras  $!Vacia(Q)$  hacer
14:     $v \leftarrow Desencolar(Q)$ 
15:    Adicionar( $v, S$ )
16:    Para todo  $w \in Adyacentes(v)$  hacer
17:      Si  $d[w] < 0$  entonces
18:        Encolar( $w, Q$ )
19:         $d[w] \leftarrow d[v] + 1$ 
20:      Fin Si
21:      Si  $d[w] = d[v] + 1$  entonces
22:         $\sigma[w] = \sigma[w] + \sigma[v]$ 
23:        Adicionar( $v, P[w]$ )
24:      Fin Si
25:    Fin Para

```

```

26:   Fin Mientras
27:   Para todo  $v \in V$  hacer
28:        $\delta[v] \leftarrow 0$ 
29:   Fin Para
30:   Mientras  $\neg \text{Vacía}(S)$  hacer
31:        $w \leftarrow \text{Sacar}(S)$ 
32:       Para todo  $v \in P[w]$  hacer
33:            $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]} (1 + \delta[w])$ 
34:       Fin Para
35:       Si  $w \neq s$  entonces
36:            $C_i^{BET}[w] \leftarrow C_i^{BET}[w] + \delta[w]$ 
37:       Fin Si
38:   Fin Mientras
39: Fin Para
40: Retornar  $C_i^{BET}$ 

```

2.1.1. Análisis de complejidad

En este epígrafe se calcula la complejidad temporal, según el enfoque teórico, de los algoritmos 1, 2 y 3. Para el cálculo de la misma se determina la complejidad de cada paso del algoritmo y se utilizan las reglas de la suma y la multiplicación de la notación asintótica "O grande" (O) para obtener la complejidad de dichos algoritmos.

Análisis del Algoritmo 1: *Calcular Centralidad*

Haciendo uso de la estructura de datos lista de vértice - lista de adyacencia, ambas implementadas haciendo uso de la estructura diccionario, se garantiza que la operación de saber si un vértice pertenece al conjunto de vértices del grafo es $O(1)$ y la operación de obtener el grado de un vértice es $O(1)$. A partir de estos criterios, se muestra la complejidad de cada paso del algoritmo.

- Paso 1 $O(1)$.
- Pasos 2-4 $O(1)$.
 - Paso 2 $O(1)$.
 - Paso 3 $O(1)$.
- Pasos 5 $O(1)$.

Aplicando la regla de la suma se puede concluir que la complejidad del Algoritmo 1 es $O(1)$.

Análisis del Algoritmo 2: *CalcularCercanía*

Sea D^n una colección unidimensional de longitud n . A continuación, se muestra la complejidad de cada paso del algoritmo.

- Paso 1 $O(1)$.
- Pasos 2-8 $O(|V|)$. Note que este tiempo se logra debido a que en la operación de los pasos 4-6 la complejidad es $O(|V|)$, luego la complejidad del paso 7 es $O(1)$, por tanto, por la regla de la suma se puede comprobar que la complejidad es $O(|V|)$.
- Paso 9 $O(1)$.

Se puede concluir que la complejidad del Algoritmo 2 es $O(|V|)$.

Análisis del Algoritmo 3: *CalcularIntermediación*

Según (41), la complejidad del Algoritmo 3 es $O(|V| |E| + |V| \log(|V|))$.

2.2. Aplicación de las técnicas para inferir redes sociales

En esta sección, se definen los parámetros que se han desarrollado por Aalst y Song para establecer relaciones entre los individuos de los registros de eventos. Estos parámetros están dirigidos a los cuatro tipos de métricas introducidos en el capítulo anterior. Es importante precisar que cada métrica asignará un peso W_{ij} a la relación entre los individuos i y j .

Las métricas basadas en causalidad tienen en cuenta tanto la entrega de trabajo como la subcontratación. Son estas dos técnicas las que se han implementado hasta la actualidad para la inferencia de redes sociales bajo el criterio de causalidad. La idea básica es que los actores están relacionados si hay una relación causal a través de la aprobación de un caso de un actor a otro. Para ambas situaciones, se aplican tres tipos de refinamientos. En primer lugar, se puede diferenciar con respecto al grado de causalidad, por ejemplo, la longitud de traspaso. Esto significa que se puede considerar no sólo la sucesión directa sino también la sucesión indirecta. En segundo lugar, se puede ignorar múltiples transferencias dentro de una instancia o no. En tercer lugar, se puede considerar las transferencias arbitrarias de trabajo o sólo tener en cuenta aquellos en los que existe una dependencia causal (para este último se necesita saber o ser capaz de derivar el modelo de proceso). Sobre la base de estos refinamientos, se derivan $2^3 = 8$ variantes, tanto para las métricas “*handover of work*” y “*subcontracting*”. Estas variantes se basan en el mismo registro de eventos (23).

A continuación, se formulan en términos matemáticos las cinco métricas implementadas para inferir redes sociales comenzando por las basadas en causalidad (“*handover of work*” y “*subcontracting*”),

luego las basadas en casos conjuntos (“*working together*”), las basadas en actividades conjuntas (“*similar task*”) y las basadas en tipo de eventos especiales (“*reassignment*”). Además, se muestra una tabla con un log de eventos la cuál será utilizada para ejemplificar las métricas que se utilizarán en la investigación.

Identificador de caso	Identificador de actividad	Actor
Caso 1	Actividad A	John
Caso 2	Actividad A	John
Caso 3	Actividad A	Sue
Caso 3	Actividad B	Carol
Caso 1	Actividad B	Mike
Caso 1	Actividad C	John
Caso 2	Actividad C	Mike
Caso 4	Actividad A	Sue
Caso 2	Actividad B	John
Caso 2	Actividad D	Pete
Caso 5	Actividad A	Sue
Caso 4	Actividad C	Carol
Caso 1	Actividad D	Pete
Caso 3	Actividad C	Sue
Caso 3	Actividad D	Pete
Caso 4	Actividad B	Sue
Caso 5	Actividad E	Clare
Caso 5	Actividad D	Clare
Caso 4	Actividad D	Pete

Tabla 2.1: Log de eventos. Tomado de (23).

Handover of work: Sea L un log. Para $p_1, p_2 \in P$ y β ($0 < \beta < 1$):

$$- p_1 \neg_L p_2 = \frac{\sum_{c \in L} |p_1 \neg_c^1 p_2|}{\sum_{c \in L} |c| - 1}$$

donde $p_1 \neg_L p_2$ significa dividir el número total de sucesiones directas de p_1 a p_2 en un registro de proceso mediante el número máximo de posibles sucesiones directas en el log.

$$- p_1 \neg'_L p_2 = \frac{\sum_{c \in L \wedge p_1 \neg_c^1 p_2} 1}{|L|}$$

donde $p_1 \neg'_L p_2$ ignora múltiples transferencias dentro de una instancia (es decir, el caso). Por ejemplo, en la Tabla 2.2, $John \neg_L Mike$ es igual a $2/14$ y $John \neg'_L Mike$ es igual a $2/5$. Tenga en cuenta que esta métrica \neg_L define una función W de peso, es decir, $p_1 \neg_L p_2 = \frac{1}{4} W_{p_1, p_2}$ es el peso de la relación de p_1 a p_2 en el sociograma correspondiente. Como se ha indicado antes, un umbral puede ser utilizado para eliminar enlaces de la sociograma.

$$- p_1 \neg_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} |p_1 \neg_c^n p_2|}{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} (|c| - n)}$$

$$- p_1 \neg'_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{1 \leq n < |c| \wedge p_1 \neg_c^n p_2} \beta^{n-1}}{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1}}$$

donde $p_1 \neg_L^\beta p_2$ y $p_1 \neg'_L^\beta p_2$ se enfocan en la sucesión indirecta mediante la introducción de un " factor de causalidad próximo " β en esta notación. Si en el contexto de un caso hay n eventos en el intervalo entre dos actores, el factor de causalidad próximo es β^n . $p_1 \neg_L^\beta p_2$ considera todas las sucesiones posibles, mientras $p_1 \neg'_L^\beta p_2$ ignora múltiples transferencias dentro de uno de los casos. Por ejemplo, en la Tabla 2.2, si β es igual a 0.5, entonces $John \neg_L Pete$ es igual a $2.5/19.5$ and $John \neg'_L Pete$ es igual a $2.5/8.5$. Si se usa un β cercano a 1 el efecto entre las distancias de los actores decrece. Por ejemplo, si se supone que sólo existe el caso 1 en la Tabla 2, se calcula el traspaso de las métricas de Juan en la Actividad de A a Mike, Juan en la actividad B, y Pete, de acuerdo con diferentes valores de β . La Tabla II muestra los resultados. Si el valor β aumenta en valor, la varianza de los valores resultantes disminuye.

$$- p_1 \sqsupset_L p_2 = \frac{\sum_{c \in L} |p_1 \sqsupset_c^1 p_2|}{\sum_{c \in L} |c| - 1}$$

$$- p_1 \sqsupset'_L p_2 = \frac{\sum_{c \in L \wedge p_1 \sqsupset_c^1 p_2} 1}{|L|}$$

$$- p_1 \sqsupset_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} |p_1 \sqsupset_c^n p_2|}{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1} (|c| - n)}$$

$$- p_1 \sqsupset'_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{1 \leq n < |c| \wedge p_1 \sqsupset_c^n p_2} \beta^{n-1}}{\sum_{c \in L} \sum_{1 \leq n < |c|} \beta^{n-1}}$$

Estos últimos cuatro indicadores ($p_1 \sqsubseteq_L p_2$, $p_1' \sqsubseteq_L p_2$, $p_1 \sqsubseteq_L^\beta p_2$ y $p_1' \sqsubseteq_L^\beta p_2$) son similares a los cuatro tipos anteriores de métricas, pero tienen en cuenta las dependencias causales reales. Por ejemplo, $p_1 \sqsubseteq_L p_2$ significa que el número total de sucesiones directas de p_1 a p_2 en un log se divide por el número máximo de posibles sucesiones directas en el log cuando p_1 y p_2 son causalmente relacionada.

Beta	$John \sqsubseteq_L^\beta Mike$	$John \sqsubseteq_L^\beta John$	$John \sqsubseteq_L^\beta Pete$
0.1	0.3116 (1/3.21)	0.0312 (0.1/3.21)	0.0031 (0.01/3.21)
0.5	0.2352 (1/4.25)	0.1176 (0.5/4.25)	0.0588 (0.25/4.25)
0.9	0.1783 (1/5.61)	0.1604 (0.9/5.61)	0.1444 (0.81/5.61)

Tabla 2.2: Handover of work acorde al factor de causalidad próximo. Tomado de (23).

Subcontracting: Sea L un log. Para $p_1, p_2 \in P, c=(c_0, c_1, \dots) \in L, |c| > 2$,

β ($0 < \beta < 1$) y utilizando la misma notación usada en la especificación de la métrica anterior:

$$- p_1 \sqsubseteq_L p_2 = \frac{\sum_{c \in L} |p_1 \sqsubseteq_c^2 p_2|}{\sum_{c \in L} |c| - 2}$$

donde $p_1 \sqsubseteq_L p_2$ significa dividir el número total de ocurrencias directas de subcontratación entre p_1 y p_2 en un registro de proceso por el número máximo de posibles ocurrencias directas de subcontratación en el log.

$$- p_1' \sqsubseteq_L p_2 = \frac{\sum_{c \in L \wedge p_1 \sqsubseteq_c^2 p_2} 1}{|L|}$$

donde $p_1' \sqsubseteq_L p_2$ ignora ocurrencias múltiples de subcontratación dentro de una instancia (es decir, un caso). Por ejemplo, en la Tabla 2.1 $John \sqsubseteq_L Mike$ es igual a 2/9 y $John' \sqsubseteq_L Mike$ es igual a 2/5.

$$- p_1 \sqsubseteq_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} |p_1 \sqsubseteq_c^n p_2|}{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} (|c| - n)(n - 1)}$$

$$- p_1' \sqsubseteq_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{2 \leq n < |c| \wedge p_1 \sqsubseteq_c^n p_2} \beta^{n-2}}{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2}}$$

donde $p_1 \sqsubseteq_L^\beta p_2$ y $p_1' \sqsubseteq_L^\beta p_2$ acuerdan con la situación donde la distancia entre estas dos actividades ejecutadas por el actor p_1 es mayor que 2. Nuevamente se introduce un “factor de causalidad próximo” β en un estilo similar a la métrica *handover of work*. Si dentro del contexto de un caso hay n eventos entre dos actividades ejecutadas por el mismo actor, el factor de causalidad próximo es β^n . $p_1 \sqsubseteq_L^\beta p_2$ considera todas las ocurrencias posibles de subcontratación, mientras $p_1' \sqsubseteq_L^\beta p_2$ ignora

subcontrataciones múltiples dentro de un caso. Por ejemplo, en la Tabla 2.2, si β es igual a 0.5 entonces $John \neg_L Mike$ es igual a $2/13$ y $John' \neg_L Mike$ es igual a $2/7$.

$$- p_1 \sqsupset_L p_2 = \frac{\sum_{c \in L} |p_1 \sqsupset_c^2 p_2|}{\sum_{c \in L} |c| - 2}$$

$$- p_1' \sqsupset_L p_2 = \frac{\sum_{c \in L \wedge p_1 \sqsupset_c^2 p_2} 1}{|L|}$$

$$- p_1 \sqsupset_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} |p_1 \neg_c^n p_2|}{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2} (|c| - n)(n - 1)}$$

$$- p_1' \sqsupset_L^\beta p_2 = \frac{\sum_{c \in L} \sum_{2 \leq n < |c| \wedge p_1 \neg_c^n p_2} \beta^{n-2}}{\sum_{c \in L} \sum_{2 \leq n < |c|} \beta^{n-2}}$$

Otra vez $p_1 \sqsupset_L p_2$, $p_1' \sqsupset_L p_2$, $p_1 \sqsupset_L^\beta p_2$ y $p_1' \sqsupset_L^\beta p_2$ son similares pero si toman en cuenta las dependencias causales reales. Por ejemplo, $p_1 \sqsupset_L p_2$ significa que el total de números de subcontrataciones directas de p_1 a p_2 en un registro de procesos es dividido por el número máximo de posibles subcontrataciones directas en el registro cuando p_1 y p_2 son causalmente relacionados.

Working together: Sea L un log. Para $p_1, p_2 \in P$ y utilizando la notación definida anteriormente:

$$- p_1 \neg_L p_2 = \frac{\sum_{c \in L} |p_1 \neg_c p_2|}{\sum_{c \in L} g(c, p_1)} \quad \text{si } \sum_{c \in L} g(c, p_1) \neq 0, \text{ si no } p_1 \neg_L p_2 = 0, \text{ donde para } c = (c_0, c_1, \dots) \in L: p_1 \neg_c p_2 = 1 \text{ si } \exists_{0 \leq i, j < |c| \wedge i \neq j} \pi_p(c_i) = p_1 \wedge \pi_p(c_j) = p_2, \text{ si no } p_1 \neg_c p_2 = 0: g(c, p_1) = 1 \text{ si } \exists_{0 \leq i < |c|} \pi_p(c_i) = p_1, \text{ si no } g(c, p_1) = 0$$

Para este tipo de métrica se ignoran las dependencias causales y simplemente se tiene en cuenta la frecuencia con que dos individuos realizan actividades para el mismo caso.

Note que en esta definición se divide el número de casos conjuntos por el número de casos en el cuál p_1 apareció. Esto es importante al usar una notación relativa. Por ejemplo, si se supone que p_1 participa en tres casos, p_2 participa en seis casos, y trabajan juntos tres veces. En esta situación p_1 siempre trabaja junto con p_2 , pero p_2 no. Por lo tanto, el valor para $p_1 \neg_L p_2$ tiene que ser más grande que el valor para $p_2 \neg_L p_1$. Por ejemplo, en la Tabla 2.1 John aparece en dos casos, Pete en cuatro casos y trabajan juntos en dos casos. Así, $John \neg_L Pete = 2/2$ y $Pete \neg_L John = 2/4$

Similar task: Sea L un log. Para $p_1 \in P, a_1 \in A$, y $c = (c_0, c_1, \dots) \in L$:

$$- p_1 \Delta_c a_1 = \sum_{0 \leq i < |c|} \begin{cases} 1 & \text{si } \pi_a(c_i) = a_1 \wedge \pi_p(c_i) = p_1 \\ 0 & \text{en otro caso} \end{cases}$$

$$- p_1 \Delta_L a_1 = \sum_{c \in L} p_1 \Delta_c a_1$$

Note que Δ define una matriz con filas P y columnas A . La Tabla 2.3 muestra la matriz recurso-actividad derivada de la Tabla 1.1.

Luego de creada la matriz, se mide la distancia entre dos actores comparando los vectores fila correspondiente. Una medida de distancia simple es la *distancia de Minkowski* la cual puede ser vista como una generalización de la *distancia Euclidiana*. Pero la *distancia Minkowski* solo muestra buenos resultados si los actores ejecutan volúmenes de trabajo comparables. Sin embargo, también se usa la *distancia de Hamming* la cual no considera la frecuencia absoluta pero solo si esta es 0 o no. Otra métrica es el *coeficiente de correlación de Pearson* usado para encontrar la relación entre casos.

El algoritmo realizado para la técnica “*Similar task*” utiliza en su implementación estos tres tipos de distancias. A continuación, se muestra detalladamente su forma de ejecución bajo los tres criterios dichos anteriormente.

Distancias ($\Delta_L^{MD,n}, \Delta_L^{HD}, \Delta_L^{PC}$): Sea L un log y Δ_L una matriz recurso-actividad. Para $p_1, p_2 \in P$,

$n \in \{1,2,3, \dots\}$:

$$- p_1 \Delta_L^{MD,n} p_2 = (\sum_{a \in A} |(p_1 \Delta_L a) - (p_2 \Delta_L a)|^n)^{1/n}$$

La *distancia de Minkowski* $\Delta_L^{MD,n}$ tiene un parámetro n : $n=1$ es la distancia rectilínea también referida a la distancia de Manhattan, $n=2$ es la *distancia Euclidiana*, y para valores grandes de n se utiliza la métrica de la *distancia de Chebyshev*.

$$- p_1 \Delta_L^{HD} p_2 = \frac{\sum_{a \in A} \delta(p_1 \Delta_L a, p_2 \Delta_L a)}{|A|} \quad \text{donde} \quad \delta(x, y) = \begin{cases} 0 & \text{si } (x > 0 \wedge y > 0) \vee (x = y = 0) \\ 1 & \text{en otro caso} \end{cases}$$

La distancia de Hamming Δ_L^{HD} no tienen ningún parámetro pero podría ser mejorado con un valor en forma de umbral.

$$- p_1 \Delta_L^{PC} p_2 = \frac{\sum_{a \in A} ((p_1 \Delta_L a) - \bar{X})((p_2 \Delta_L a) - \bar{Y})}{\sqrt{\sum_{a \in A} ((p_1 \Delta_L a) - \bar{X})^2 \sum_{a \in A} ((p_2 \Delta_L a) - \bar{Y})^2}} \quad \text{donde} \quad \bar{X} = \frac{\sum_{a \in A} (p_1 \Delta_L a)}{|A|}, \quad \bar{Y} = \frac{\sum_{a \in A} (p_2 \Delta_L a)}{|A|}$$

En el caso del *coeficiente de correlación de Pearson*, los resultados están en el rango de +1 a -1. Una correlación de +1 significa que hay una perfecta relación lineal positiva entre variables. Una correlación de -1 quiere decir que hay una relación lineal negativa máxima entre variables. En otras palabras, si la distancia entre actores es pequeña, la correlación es cercana a 1, si es grande, la correlación es cercana a -1.

Para ilustrar las limitaciones de las métricas simples como la *distancia de Minkowski* es necesario apoyarse en la Tabla 2.3. Claramente, desde un punto de vista intuitivo la distancia entre Sue y Carol podría ser más pequeña que la distancia entre Carol y Clare porque Carol y Clare no tienen actividades en común. La *distancia de Minkowski* ($n=1$) entre Sue y Carol es igual a 3 y la distancia entre Carol y

Clare es igual a 4. Sin embargo, si Sue hubiese ejecutado la actividad B y C también tres veces, la distancia entre Sue y Carol sería de 7 y así indicar incorrectamente que Carol está más cerca de Clare que de Sue. La *distancia de Hamming* es más robusta y podría indicar en ambos casos que Carol está más cerca de Sue: $Sue \Delta_L^{HD} Carol$ es igual a 1/5 y $Carol \Delta_L^{HD} Clare$ es igual a 4/5. Si se calcula el *coeficiente de correlación de Pearson*, $Sue \Delta_L^{PC} Carol$ da como resultado 0.2182 y $Carol \Delta_L^{PC} Clare$ es igual a -0.6667. Por eso, el resultado de la *correlación de Pearson* se puede llevar a la misma conclusión que se arribó en la *distancia de Hamming*.

Recurso/Actividad	Actividad A	Actividad B	Actividad C	Actividad D	Actividad E
John	2	1	1	0	0
Sue	3	1	1	0	0
Mike	0	1	1	0	0
Carol	0	1	1	0	0
Pete	0	0	0	4	0
Clare	0	0	0	1	1

Tabla 2.3: Matriz recurso-actividad. Tomado de (23).

Se puede deducir que, si el volumen de trabajo varía considerablemente, las métricas no son adecuadas. Por ejemplo, es difícil comparar el perfil de una parte del tiempo de un trabajador con el del tiempo completo de un trabajador. Así, en algunos casos primero se aplica la función $\log_k(X+1)$ en los valores de la matriz recurso-actividad, es decir, usar una escala logarítmica para Δ_L . Generalmente se adiciona "+1" para evitar valores negativos.

Reassignment: Sea L un log. Para $p_1, p_2 \in P$:

$$- p_1 \nabla_L^{reassign} p_2 = \frac{(\sum_{c \in L} |p_1 \nabla_c^{reassign} p_2|)}{(\sum_{c \in L} (|c|-1))}$$

donde $p_1 \nabla_L^{reassign} p_2$ se obtiene al dividir el total de número de reasignaciones de p_1 a p_2 en el log de eventos por el número máximo de reasignaciones en el log. Por ejemplo, si hay 10 eventos en el log y *John* tiene reasignada una actividad a *Mike* a la vez, $John \nabla_L^{reassign} Mike$ es igual a 1/9.

$$- p_1 \nabla_L^{reassign} p_2 = \frac{(\sum_{c \in L \wedge p_1 \nabla_c^{reassign} p_2} 1)}{|L|}$$

donde $p_1 \nabla_L^{reassign} p_2$ ignora reasignaciones múltiples dentro de una instancia.

En esta sección se formularizaron las métricas introducidas en el capítulo 1 de la investigación. Es importante tener en cuenta que cada una de las métricas son derivadas de algún log de eventos, pues estos algoritmos tienen como entrada dichos logs. Los resultados de la aplicación de estas métricas pueden ser representados en término de grafos ponderados (P, R, W) donde P es el conjunto de

actores, R el conjunto de relaciones y W la función que indica el peso de la relación. Así, se establecería una relación vinculando la información modelada al concepto de redes sociales.

Partiendo de las métricas formuladas anteriormente y con el objetivo de modelar la información a través de redes sociales se diseñaron los Algoritmos 4, 5, 6, 7 y 8.

- **Algoritmo 4** *HandoverOfWork*

Entrada: Un log de eventos

Salida: Matriz D normalizada para la construcción de la red social

1: $numOriginator \leftarrow$ longitud de la lista de originadores

2: $D \leftarrow$ crear matriz $D[numOriginator, numOriginator]$

3: **Por cada** $t \in L$ **hacer**

4: **Para todo** $e \in T$ **hacer**

5: $xEvent \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima

6: $xEvent2 \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima + 1

7: $fil \leftarrow$ posición en la lista de originadores del recurso de $xEvent$

8: $col \leftarrow$ posición en la lista de originadores del recurso de $xEvent2$

9: **Si** $fil \neq -1$ y $col \neq -1$ **entonces**

10: $D \leftarrow D[fil+1, col+1]$

11: **Fin Si**

12: **Fin Para**

13: **Fin Por**

14: **Retornar** D normalizada

- **Algoritmo 5** *Subcontracting*

Entrada: Un log de eventos

Salida: Matriz D normalizada para la construcción de la red social

1: $numOriginator \leftarrow$ longitud de la lista de originadores

2: $D \leftarrow$ crear matriz $D[numOriginator, numOriginator]$

3: **Por cada** $t \in L$ **hacer**

4: **Para todo** $e \in T$ **hacer**

5: $xEvent \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima

6: $xEvent2 \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima + 1

7: $xEvent3 \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima + 2

8: $fil \leftarrow$ posición en la lista de originadores del recurso de $xEvent$

9: $fil2 \leftarrow$ posición en la lista de originadores del recurso de $xEvent2$

10: $fil3 \leftarrow$ posición en la lista de originadores del recurso de $xEvent3$

11: **Si** $fil \neq -1$ y $fil2 \neq -1$ y $fil3 \neq -1$ **entonces**

12: $D \leftarrow D[fil+1, fil2+1]$

13: **Fin Si**

14: **Fin Para**

15: **Fin Por**

16: **Retornar** D normalizada

- **Algoritmo 6** *WorkingTogether*

Entrada: Un log de eventos

Salida: Matriz D para la construcción de la red social

1: $numOriginator \leftarrow$ longitud de la lista de originadores

2: $D \leftarrow$ crear matriz $D[numOriginator, numOriginator]$

3: $Aux \leftarrow$ crear matriz $Aux[numOriginator, numOriginator]$ mediante *HandoverOfWork*

4: **Para todo** $i < numOriginator$ **hacer**

5: **Para todo** $j < numOriginator$ **hacer**

6: $D[i, j] \leftarrow (Aux[i, j] + Aux[j, i]) / 2$

7: $D[j, i] \leftarrow D[i, j]$

8: **Fin Para**

9: **Fin Para**

10: **Retornar** D

- **Algoritmo 7** *SimilarTask*

Entrada: Un log de eventos

Salida: Matriz D para la construcción de la red social

1: $numOriginator \leftarrow$ longitud de la lista de originadores

2: $D \leftarrow$ crear matriz $D[numOriginator, numOriginator]$

3: **Por cada** $t \in L$ **hacer**

4: **Para todo** $e \in T$ **hacer**

5: $xEvent \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima

6: $fil \leftarrow$ posición en la lista de originadores del recurso de $xEvent$

7: $col \leftarrow$ posición en la lista de originadores del recurso de $xEvent$

8: $D \leftarrow D[fil+1, col+1]$

9: **Fin Para**

10: **Fin Por**

11: **Retornar** D

- **Algoritmo 8** *Reassignment*

Entrada: Un log de eventos

Salida: Matriz D normalizada para la construcción de la red social

1: $numOriginator \leftarrow$ longitud de la lista de originadores

2: $D \leftarrow$ crear matriz $D[numOriginator, numOriginator]$

3: $contador \leftarrow 0$

4: **Por cada** $t \in L$ **hacer**

5: $contador \leftarrow$ longitud de $t - 1$

6: **Para todo** $e \in T$ **hacer**

7: $xEvent \leftarrow$ crear evento con el elemento de la traza en la posición k -ésima

```

8:          fil ← posición en la lista de originadores del recurso de xEvent
9:          Si transición de e = “reassign” entonces
10:             Para todo e ∈ T hacer
11:                xEvent2 ← crear evento con el elemento de la traza en la
                           posición k-ésima + 1
12:                col ← posición en la lista de originadores del recurso de xEvent2
13:                Si col != -1 entonces
14:                   Si nombre de xEvent = nombre de xEvent2
15:                      D ← D[fil+1, col+1]
16:                   Fin Si
17:                Fin Si
18:            Fin Para
19:        Fin Si
20:    Fin Para
21: Fin Por
22: Retornar D normalizada

```

2.2.1. Análisis de complejidad

En este epígrafe se calcula la complejidad temporal, según el enfoque teórico, de los algoritmos propuestos en la sección anterior.

Análisis del algoritmo 4: *HandoverOfWork*

Es importante mencionar que este algoritmo tiene cuatro implementaciones diferentes debido a que posee cuatro criterios diferentes para su ejecución. Todas las implementaciones del algoritmo poseen la misma complejidad temporal en el peor de los casos. Por lo que se utiliza la variante más genérica del algoritmo, la cual considera causalidad, sucesiones directas y múltiples transferencias.

Criterios de ejecución del algoritmo *HandoverOfWork*:

- Considerando causalidad, considerando sucesiones directas, considerando múltiples transferencias.
- Ignorando causalidad, considerando sucesiones directas, ignorando múltiples transferencias.
- Ignorando causalidad, ignorando sucesiones directas, considerando múltiples transferencias.
- Ignorando causalidad, ignorando sucesiones directas, ignorando múltiples transferencias.

Haciendo uso de la estructura de datos lineales (listas) y estructuras de datos homogéneos (arreglos) se facilita el trabajo con matrices. A partir de este criterio, se muestra la complejidad de cada paso del algoritmo.

- Paso 1-2 $O(1)$.
- Pasos 3-13 $O(n*m)$. Note que en estos 10 pasos se puede ver la aparición de un ciclo anidado donde el primer ciclo tiene una complejidad $O(n)$, siendo n la cantidad de trazas del log y el segundo posee una complejidad de $O(m)$, siendo m la cantidad de eventos de cada traza.
 - Paso 4-12 $O(m)$. Note que dentro de este ciclo solo se encuentran operaciones con complejidad $O(1)$ por lo que la complejidad de este bucle sigue siendo $O(m)$ por lo tanto, se multiplica.
- Pasos 14 $O(1)$.

Se puede concluir que la complejidad del algoritmo en el peor de los casos es $O(n*m)$ aplicando de esta misma forma a las restantes tres variaciones de su ejecución.

Análisis del algoritmo 5: *Subcontracting*

Este algoritmo al igual que el anterior tiene cuatro criterios para su ejecución. Se decidió tomar la variante ignorando causalidad, considerando sucesiones directas y considerando múltiples transferencias para su análisis. Es importante destacar que ninguno de los otros criterios varía en complejidad, al igual que en el algoritmo anterior.

Criterios de ejecución del algoritmo *Subcontracting*:

- Ignorando causalidad, considerando sucesiones directas, considerando múltiples transferencias.
- Ignorando causalidad, considerando sucesiones directas, ignorando múltiples transferencias.
- Ignorando causalidad, ignorando sucesiones directas, considerando múltiples transferencias.
- Ignorando causalidad, ignorando sucesiones directas, ignorando múltiples transferencias

A partir de estos criterios y usando el mismo tipo de estructura de datos que en el algoritmo anterior, se muestra la complejidad de cada paso del algoritmo.

- Paso 1-2 $O(1)$.
- Pasos 3-15 $O(n*m)$. Al igual que en el algoritmo anterior se puede ver la aparición de un ciclo anidado donde el primer ciclo tiene una complejidad $O(n)$ y el segundo posee una complejidad de $O(m)$ por lo tanto, se multiplica.
 - Paso 4-14 $O(m)$. Note que dentro de este ciclo solo se encuentran operaciones con complejidad $O(1)$ por lo que la complejidad de este bucle sigue siendo $O(m)$.

- Pasos 16 $O(1)$.

Se puede concluir que la complejidad del algoritmo en el peor de los casos es $O(n*m)$.

Análisis del algoritmo 6: *WorkingTogether*

El presente algoritmo se ejecuta bajo tres criterios. Para su análisis se tomó como variante considerando la distancia sin causalidad ($\beta = 0.5$). A continuación, se exponen los tres criterios.

Criterios de ejecución del algoritmo *Subcontracting*:

- Considerando la distancia sin factor causalidad próximo ($\beta = 0.5$).
- Considerando la distancia con factor causalidad próximo.
- Con relación simultánea de transferencia.

En sus tres variantes la complejidad temporal sigue siendo la misma en el peor de los casos.

- Paso 1-3 $O(1)$.
- Pasos 4-9 $O(n*n)$. Note que en este caso los dos ciclos tienen que hacer la misma cantidad de iteraciones por lo que la complejidad es cuadrática.
 - Paso 5-8 $O(n)$. Note que todas las operaciones dentro del ciclo tienen complejidad $O(1)$ por lo que la complejidad de este bucle es $O(n)$.
- Pasos 10 $O(1)$.

Se concluye que la complejidad del algoritmo es $O(n^2)$.

Análisis del algoritmo 7: *SimilarTask*

Los criterios que utiliza este algoritmo están determinados por la forma de calcular la distancia entre vértices. Con ese objetivo utilizan un criterio ignorando la distancia que es en su forma básica y los otros tres criterios son calculando la distancia mediante la “*distancia Euclidiana*”, “*distancia de Hamming*” y el “*coeficiente de correlación de Pearson*”. Para todos estos criterios el algoritmo se ejecuta realizando múltiples transferencias. El análisis de complejidad que se muestra a continuación es en su forma básica.

- Paso 1-2 $O(1)$.
- Pasos 3-10 $O(n*m)$. Note que el primer ciclo recorre todas las trazas del log siendo n la cantidad de trazas por lo que su complejidad sería $O(n)$. El ciclo anidado recorre todos los elementos de cada traza, siendo m la cantidad de elementos de cada traza por lo que su complejidad es $O(m)$.

- Paso 5-8 $O(m)$. Note que las cuatro operaciones dentro del ciclo tienen complejidad $O(1)$ por lo que la complejidad de este bucle es $O(m)$.
- Pasos 10 $O(1)$.

Se llega a la conclusión de que la complejidad del algoritmo en el peor de los casos es $O(n*m)$ siendo esta misma la complejidad para sus tres restantes criterios.

Análisis del algoritmo 8: Reassignment

- Paso 1-3 $O(1)$.
- Pasos 4-21 $O(n*m*k)$. Note que el primer ciclo recorre todas las trazas del log siendo n la cantidad de trazas por lo que su complejidad sería $O(n)$. Los ciclos anidados recorren todos los elementos de cada traza, siendo $m = k$ la cantidad de elementos de cada traza.
 - Paso 5 $O(1)$.
 - Paso 6-20 $O(m*k)$. Note que k va a ser igual a m en el peor de los casos debido a que el bucle más profundo está condicionado, por lo que puede que no siempre k itere las mismas veces que m por esta razón su complejidad es $O(m*k)$.
 - Paso 7-8 $O(1)$.
 - Paso 9-19 $O(k)$. Note que la condición tiene un ciclo dentro por lo que su complejidad es $O(k)$ debido a que la complejidad de las operaciones dentro del bucle (Paso 11-18) tienen complejidad $O(1)$.
- Pasos 21 $O(1)$.

Se arriba a la conclusión de que la complejidad del algoritmo en el peor de los casos es $O(n*m*k)$. Este algoritmo, a diferencia de los demás es el único con complejidad mayor a $O(n*m)$. Además, en su ejecución se basa en un único criterio, analizado anteriormente.

2.3. Cálculo de factores para realizar el ARS

En este epígrafe se enuncian varios conceptos referidos a los tres factores para el ARS que se exponen en la presente investigación. Se utiliza los conceptos de popularidad, eficiencia y sobrecarga. Para el cálculo de estos factores se hará uso de las medidas: centralidad, cercanía e intermediación.

A continuación, se describen en detalles cada uno de estos factores.

Popularidad

La popularidad de un nodo se mide por el número de conexiones que tiene, como el número de amigos en Facebook. Esta es una manera de codificar en el lenguaje de las redes el concepto que se tiene en mente: ser popular es ser reconocido por la mayoría en algún aspecto. En las redes, la popularidad se traduce en “más conectividad”. Este principio se aplica habitualmente para el crecimiento de redes complejas dentro del concepto de conexión preferencial, en el que se sugiere que, cuanto más conectado está un nodo, más probable es que reciba nuevos enlaces (42).

Este concepto enriquece el ARS en el contexto de la investigación. Se puede relacionar estrechamente con la medida de *centralidad* brindando información de diferentes puntos de vista. Primero, conociendo el nodo con mayor centralidad en la red se puede determinar que ese nodo tiene gran probabilidad de tener un enlace que engrandezca la red. Esto podrá ser visto cuando se decida ampliar el modelo de procesos, o sea, que se decida introducir una nueva actividad al proceso. Siendo así se puede inferir que el nodo con mayor centralidad es el recurso en el proceso con mayor probabilidad de que se vincule al recurso de la nueva actividad que se decida agregar al proceso.

Desde otro punto de vista, la centralidad de un nodo en la red indica su nivel de importancia con respecto a los demás. Dentro de un proceso siempre van a existir recursos con mayor importancia que otros. A través de este factor se puede entonces determinar los recursos indispensables del proceso teniendo gran impacto en la toma de decisiones.

Eficiencia

Lograr la eficiencia en el flujo de información es uno de los objetivos que se pretende dentro de una red social. Cuando esto ocurre, en la mayoría de los casos lo que se pretende es optimizar los recursos de la red. La optimización de recursos es uno de los factores que siempre se tiene en cuenta en cualquier ámbito viéndose con mayor desempeño en el entorno empresarial (30).

En el ARS, la cercanía es una medida importante, pues conociendo cuáles son los caminos más cortos de un nodo hacia todos los demás dentro de un grafo se pueden optimizar los procesos de búsquedas. En la MP, esta medida puede ser aplicada para conocer cuán eficiente es el flujo de información entre los recursos de un proceso determinado. La eficiencia dentro de un proceso puede estar condicionada por dos factores importantes: el tiempo y/o el costo de realizar una actividad del proceso (43). Estos dos factores juegan un papel fundamental a la hora de aplicar esta medida debido a que, basada en ellos es que dicha medida determina los flujos con menor tiempo y/o coste para su realización de forma ordenada.

Se puede decir entonces que existe una estrecha relación entre la “*eficiencia*” y la medida de “*cercanía*” para el ARS. Por lo que a través de la aplicación de esta medida se puede determinar la eficiencia en el flujo de información entre los recursos de un proceso.

Sobrecarga

La sobrecarga de un recurso del proceso evita que dicho recurso pueda transferir la información hacia los demás recursos del proceso en el momento necesario, o sea, en un mismo instante de tiempo sus entradas van a ser mayores que sus salidas (29). A partir del conocimiento de la sobrecarga de los recursos del proceso es posible detectar la existencia de posibles cuellos de botellas en una red social.

La intermediación determina la frecuencia con que un nodo en un grafo forma parte de un camino entre otros dos nodos (11). Teniendo en cuenta esto se puede determinar cuál es el recurso del proceso que está haciendo función de puente entre otros. Por lo que se puede determinar que este nodo posee sobrecarga de trabajo. A partir de este análisis, se puede identificar posibles cuellos de botella.

Los cuellos de botella son las actividades que disminuyen la velocidad de los procesos, incrementan los tiempos de espera y reducen la productividad, trayendo como consecuencia final el aumento en los costos. Como bien se puede apreciar, estos cuellos de botella pueden verse influenciados por la existencia de sobrecarga en alguno de los recursos de un proceso (44).

Se considera oportuno enfatizar que la propuesta realizada constituye un algoritmo extensible, debido a la posibilidad de adicionar más factores para el ARS sin afectar los demás componentes del algoritmo, lo cual permitirá enriquecer los análisis posteriores que se realicen en su aplicación.

2.4. Conclusiones del capítulo

En el presente capítulo se ha descrito un algoritmo para la inferencia de redes sociales a partir de un log de eventos, que representa los subprocesos que componen a determinado proceso. Como parte del mismo se han enunciado varias definiciones y se ha realizado el diseño y análisis de varios algoritmos para la inferencia de redes sociales en la MP. En el Algoritmo propuesto el cálculo de los factores popularidad, eficiencia y sobrecarga, a partir de las medidas de centralidad enunciadas, constituye la principal novedad. Por lo que se puede concluir que:

- El uso de los conceptos relacionados con la teoría de grafos y las medidas de centralidad contribuyen a realizar ARS, lo que permite determinar comportamientos de los actores en las redes.
- Las métricas descritas para inferir redes sociales a partir de subprocesos, permiten obtener información relevante acerca de las interacciones de los recursos en un proceso determinado.
- El algoritmo presentado es extensible, debido a que se pueden adicionar más factores para el ARS sin afectar los demás componentes del mismo, enriqueciendo de esta forma los análisis posteriores

Capítulo 3

Validación de la propuesta

3. VALIDACIÓN DE LA PROPUESTA

En este capítulo, se hace una descripción de la realización computacional del algoritmo propuesto. Se describe un caso de estudio que se utilizó para validar la propuesta. Además, Se muestran los resultados obtenidos a partir del cálculo de los factores popularidad, eficiencia y sobrecarga. Finalmente se realiza una discusión de los resultados obtenidos.

3.1. Realización computacional del algoritmo propuesto

Para la realización computacional del algoritmo propuesto se desarrolló un plugin para la herramienta de minería de procesos ProM.

ProM es un marco de trabajo para el desarrollo de herramientas de minería de procesos en un ambiente estandarizado (45). Está desarrollado en Java y se encuentra disponible bajo licencia GPL1.0. ProM está concebido para admitir la adición de plugins y de esta manera posibilitar el desarrollo de nuevos algoritmos y técnicas en el campo de la minería de procesos. Los complementos necesitan determinada cantidad de parámetros de entrada y producen uno o varios objetos de salida. Los parámetros de entrada pueden ser registros de eventos u objetos obtenidos a partir del procesamiento realizado por otros complementos. Mientras que los objetos de salida obtenidos pueden ser empleados como parámetros de entrada de otros complementos.

Esta herramienta de código abierto y distribuida gratuitamente, ha sido el estándar impuesto para la minería de procesos durante la última década. Permite el proceso de descubrimiento, la comprobación de la conformidad, análisis de redes sociales, la minería de organización, la minería de decisión (46). Provee más de 600 técnicas con capacidades de minería de procesos, una cifra que imposibilita saber cuáles usar en determinados contextos o situaciones (47) . Además de que cada técnica tiene un conjunto de métricas y parámetros de necesaria configuración. Las herramientas desarrolladas en ProM han sido empleadas en el análisis de procesos provenientes de diferentes dominios entre los que se encuentran gubernamental, hospitalario y sistemas ERP (48).

La herramienta requiere experiencia en minería de procesos y no está respaldada por una organización comercial. Estas herramientas de Minería de procesos, aunque son eficientes en su campo, poseen alta complejidad en configuraciones técnicas y procedimientos engorrosos (49), tanto para extraer registros de eventos como para obtener modelos de la ejecución de los procesos. Por lo tanto, tiene las ventajas y desventajas comunes para el software de código abierto (47) .

Para el desarrollo del plugin se utilizó XP como metodología de desarrollo de software, debido a las características y la cantidad de recursos humanos disponibles para realizar la presente investigación.

En XP la producción de código debe realizarse en parejas de programadores. Puesto que este estilo de programación brinda una amplia gama de beneficios, dentro los cuales están: la mayoría de los

errores son detectados conforme son introducidos en el código, por consiguiente, la tasa de errores del producto final es baja; se posibilita la transferencia de conocimientos de programación entre los miembros del equipo y los programadores se comunican entre ellos, mejorando así el flujo de información y la dinámica del equipo (50).

Haciendo uso de la metodología anterior se desarrolló una aplicación informática en forma de plugin para la herramienta de trabajo ProM, la cual provee las siguientes funcionalidades:

- Calcular métricas para inferir red social.
- Construir red social para el subproceso seleccionado.
- Calcular los factores popularidad, eficiencia y sobrecarga.

Para describir las funcionalidades anteriores se utilizaron las Historias de Usuario (HU) que se muestran en las tablas 3.1, 3.2 y 3.3.

El plugin fue desarrollado utilizando Java como lenguaje de programación en su versión 1.8, la herramienta ProM en su versión 6.5.1, la biblioteca de clases SocialNetwork en su versión 2.0 para la representación de redes sociales, la biblioteca de clases Log 1.0 para el filtrado de logs de eventos y como entorno de desarrollo NetBeans en su versión 8.1 RC2.

Historia Usuario “Calcular métricas para inferir red social”	
Número: 1	Nombre UH: Calcular métricas para inferir redes sociales
Modificación de HU número:	
Referencia:	
Programador: Leansi Vega Rouco	Iteración asignada: 1
Prioridad en el negocio: Alta.	Puntos estimados (días): 3
Riesgo de desarrollo: Alto.	Puntos reales (días): 3
Descripción: El plugin debe ser capaz de calcular las métrica establecidas en la minería de redes sociales a partir de varios logs de eventos recibidos por parámetros.	
Observaciones:	

Tabla 3.1: HU “Calcular métricas para inferir red social”.

Historia Usuario “Construir red social para el subproceso seleccionado”	
Número: 2	Nombre UH: Construir red social para el subproceso seleccionado
Modificación de HU número:	
Referencia:	
Programador: Leansi Vega Rouco	Iteración asignada: 1
Prioridad en el negocio: Alta.	Puntos estimados (días): 1
Riesgo de desarrollo: Medio.	Puntos reales (días): 1
Descripción: El plugin debe ser capaz, a partir de la métrica seleccionada por el usuario, de construir la red social correspondiente al subproceso seleccionado.	
Observaciones:	

Tabla 3.2: HU “Construir red social para el subproceso seleccionado”.

Historia Usuario “Calcular los factores de ARS”	
Número: 3	Nombre UH: Calcular los factores de ARS
Modificación de HU número:	
Referencia:	
Programador: Leansi Vega Rouco	Iteración asignada: 1
Prioridad en el negocio: Alta.	Puntos estimados (días): 2
Riesgo de desarrollo: Medio.	Puntos reales (días): 2
Descripción: El plugin debe ser capaz de calcular los factores de ARS de la red social que se esté analizando.	
Observaciones:	

Tabla 3.3: HU “Calcular los factores de ARS”.

En las tablas 3.4, 3.5 y 3.6 se muestran las tarjetas Clase, Responsabilidad y Colaboración (CRC), las cuales describen las relaciones de las clases utilizadas para el desarrollo de la aplicación.

CRC “SocialNetwork”	
Responsabilidad:	Colaboradores:
Calcular métricas para inferir red social a partir de un log de eventos	JScrollPane, JCheckBox, JComboBox, JPanel, JSlider, Transformer, SocialNetworkAnalysisUI, XLog, XEvent, XTrace

Tabla 3.4: CRC “Calcular métricas para inferir red social”.

CRC “SocialNetworkAnalysisUI”	
Responsabilidad:	Colaboradores:
Construir red social	BasicOperation, OperationFactory, UtilOperation, SocialNetworkAnalysisUtil, XLog, VisualizationViewer, AggregateLayout, Graph, VisualizationServer, VertexShapeSizeAspect, VertexStrokeHighlight, DirectionDisplayPredicate, JCheckBox, JComboBox, JPanel, JSlider, Transformer

Tabla 3.5: CRC “Construir red social para el subproceso seleccionado”.

CRC “SocialNetworkAnalysisUtil”	
Responsabilidad:	Colaboradores:
Calcular los factores de ARS	Betweenness, Closeness, DegreeCentrality, XLog, VisualizationViewer, Layout, Graph, VertexShapeSizeAspect, JComboBox, JPanel, Transformer, Dimension,

Tabla 3.6: CRC “Calcular los factores de ARS”.

En la Figura 3.1 se muestra una imagen de la herramienta ProM haciendo uso del plugin desarrollado, en la que se visualizan las relaciones sociales entre los recursos de un proceso determinado. En el panel izquierdo se muestra el **Árbol de Variantes (AV)** obtenido por el método de descomposición de un proceso en subproceso utilizado en (9). Los patrones utilizados son los encargados de realizar dicha descomposición bajo criterios diferentes y tienen como hijos las actividades asociadas a ese

subproceso. En el panel derecho se muestra un componente desplegable donde se visualiza la red social del proceso global o un subproceso de él que se seleccione en el AV.

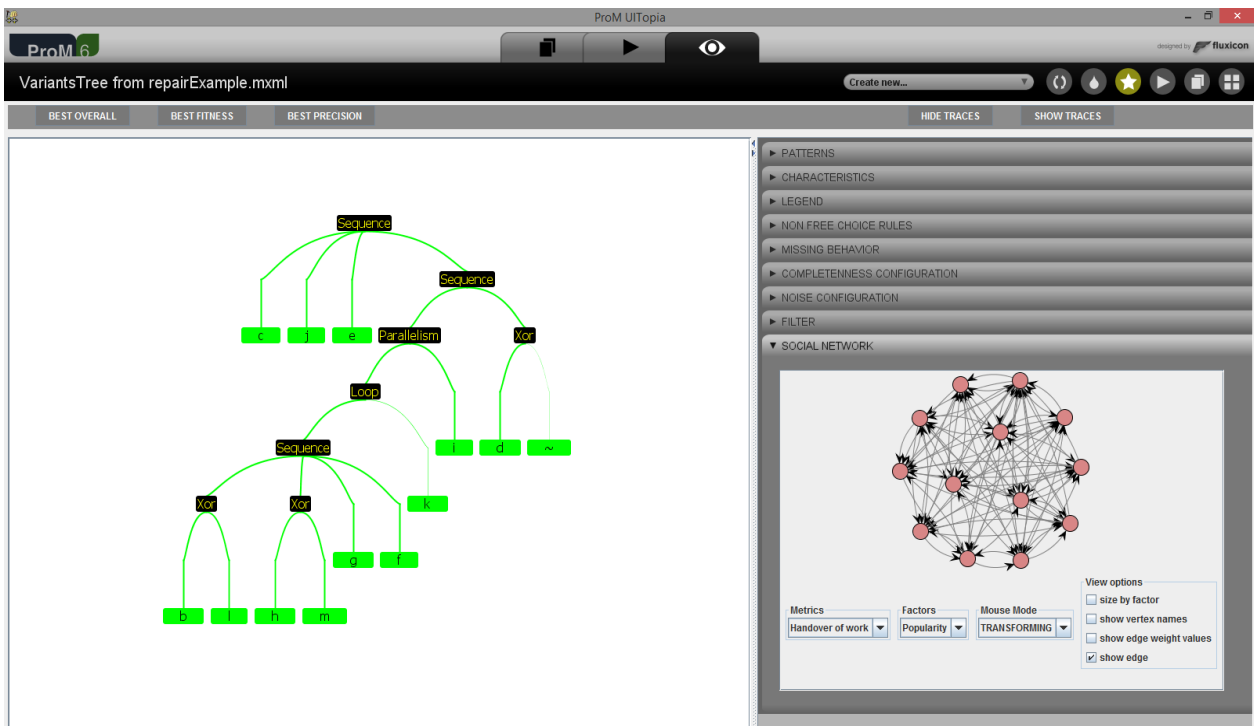


Figura 3.1: Red social asociada al proceso global.

La parte inferior del panel contiene los parámetros de configuración de la red social. El primer parámetro de configuración del panel es *Metrics* el cual brinda la opción de inferir la red social con las métricas existentes (*Handover of work*, *Subcontracting*, *Working together*, *Similar task* y *Reassignment*). El segundo es *Factors* mediante el cual se pueden determinar los factores popularidad, eficiencia y sobrecarga (*Popularity*, *Efficiency* y *Overload*). Como tercer parámetro de configuración se encuentra *Mouse Mode* que tiene la función de mover la red completamente por todo el panel a través de la opción *TRANSFORMING* o desplazar uno o varios nodos de la red por el panel con la opción *PICKING*. Por último, *View options* permite añadir otros elementos a la visualización de la red social. La opción *size by factors* asigna un tamaño a los nodos de la red según el factor que se esté calculando. Seguidamente *show vertex names* como su nombre lo indica, muestra el nombre de los nodos de la red. Además, la opción *show edge weight values* muestra el valor de las aristas, o sea, el peso de la relación entre dos nodos de la red. La última opción *show edge* da la posibilidad de mostrar las aristas de la red.

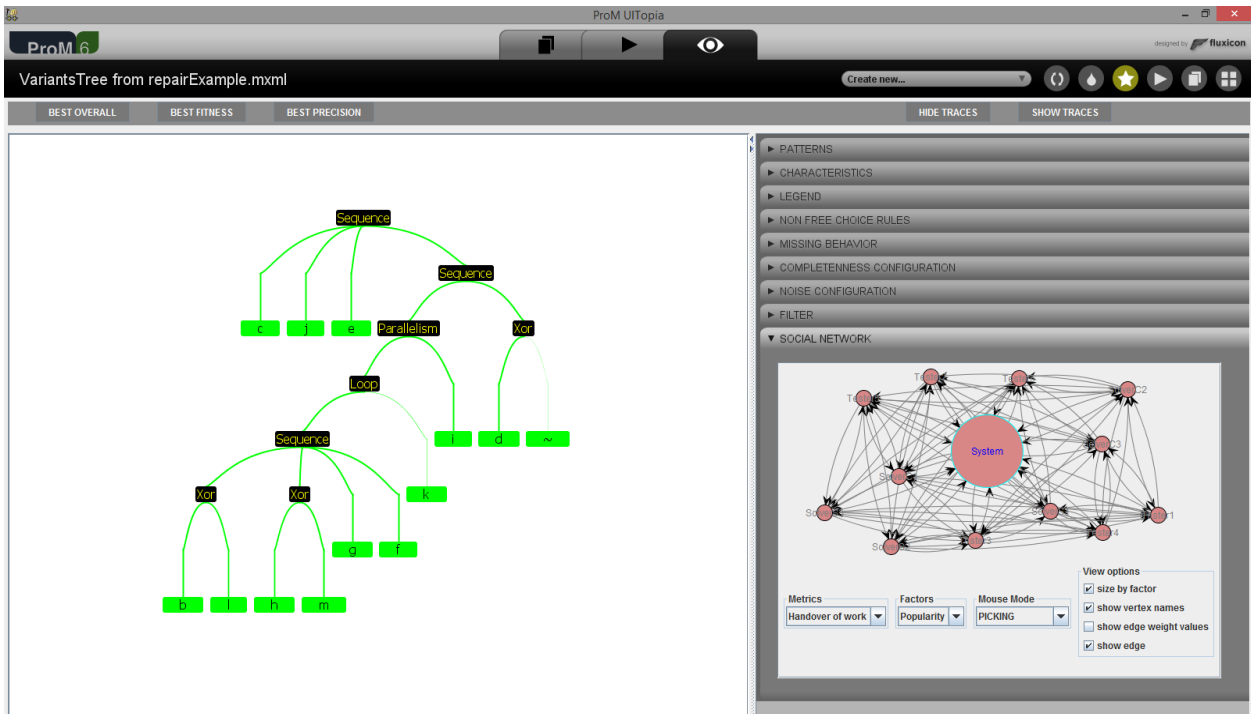


Figura 3.2: Visualización de la red social mediante los parámetros de configuración.

En la Figura 3.2 se visualiza la red social obtenida con la métrica *Handover of work*. Donde se hace uso de la opción *size by factors*, *show vertex names*, *show edge*, *PICKING* y el factor *popularity*. A través de estos parámetros se puede determinar que el nodo con mayor popularidad de la red es el nodo "System".

3.2. Caso de estudio

A continuación, se describe el caso de estudio utilizado para aplicar el modelo propuesto, así como la descripción detallada del log de eventos asociado a dicho caso de estudio.

El log de eventos que a continuación se detalla pertenece a uno de los servicios hoteleros que se efectúan en nuestro país. Específicamente hace referencia al registro de llamadas que se controlan en las recepciones de los hoteles. Los recursos que se ven involucrados en el proceso global y en los subprocesos del log de eventos son recursos humanos que representan a cada uno de los recepcionistas del hotel. Las actividades que realizan cada uno de ellos representan las formas de relacionarse uno con los otros.

El log de eventos contiene 162 instancias de proceso y un total de 1042 eventos repartido en 12 clases diferentes de las cuales solo *Check_In+Complete* y *Corrección/Cargo_de_Llamadas+Complete* son tareas iniciales, aunque en su gran mayoría los procesos comienzan con *Check_In+Complete* (159 ocurrencias) que representa un 98% y solo un 2% comienzan con la otra tarea inicial (3 ocurrencias). Hay también 5 clases que finalizan el proceso: *Cierre_de_Facturas+Complete* con 66 ocurrencias (41%) y *Check_Out_es_Cierre_de_Facturas+Complete* con 47 ocurrencias (30%),

Check_Out_de_Habitaciones con 39 ocurrencias (24%), mientras que los 2 restantes finalizan el proceso en muy pocos casos Modificación_de_Habitaciones con 9 que representa solo el 6% y solo un proceso culmina con Corrección/Cargo_de_Llamadas (0.6%). En general la clase que más participa en el proceso es Corrección/Cargo_de_Llamadas con 272 ocurrencias en todo el proceso, participación del 26%.

Como recursos se tienen 9 clases predominando la participación de R8 con 263 ocurrencias (25%), R9 con 225 ocurrencias (22%), R1 con 210 ocurrencias (20%), R4 con 199 ocurrencias para un 20%, contrastando con esto los que menos participación presentan son R5 con 16 ocurrencias (1.5%), R7 con 10 (1%), R6 y R3 con 4 ocurrencias (0.4) cada uno.

3.3. Cálculo de los factores popularidad, eficiencia y sobrecarga

En este epígrafe se realiza el cálculo de los factores popularidad, eficiencia y sobrecarga para el proceso global y para cada subproceso contenido en el log de eventos descrito en el caso de estudio. El objetivo de realizar dicho análisis es demostrar que la comprensión del proceso global resulta menos compleja comprendiendo los subprocesos que lo compone, mediante la inferencia de una red social por cada uno de ellos y el cálculo de dichos factores.

Para realizar el cálculo de los factores popularidad, eficiencia y sobrecarga se utilizó la métrica *Handover of work*, debido a que en la red inferida por esta métrica se puede apreciar con mayor facilidad la presencia de dichos factores, lo cual posibilita una mayor comprensión en el análisis.

La Figura 3.3 muestra la red social asociada al proceso global del log de eventos del caso de estudio presentado en el epígrafe anterior.

Como se puede apreciar en la red social inferida, el alto número de interacciones entre los recursos dificulta el análisis de los factores popularidad, eficiencia y sobrecarga, lo cual provoca poca comprensión del proceso global.

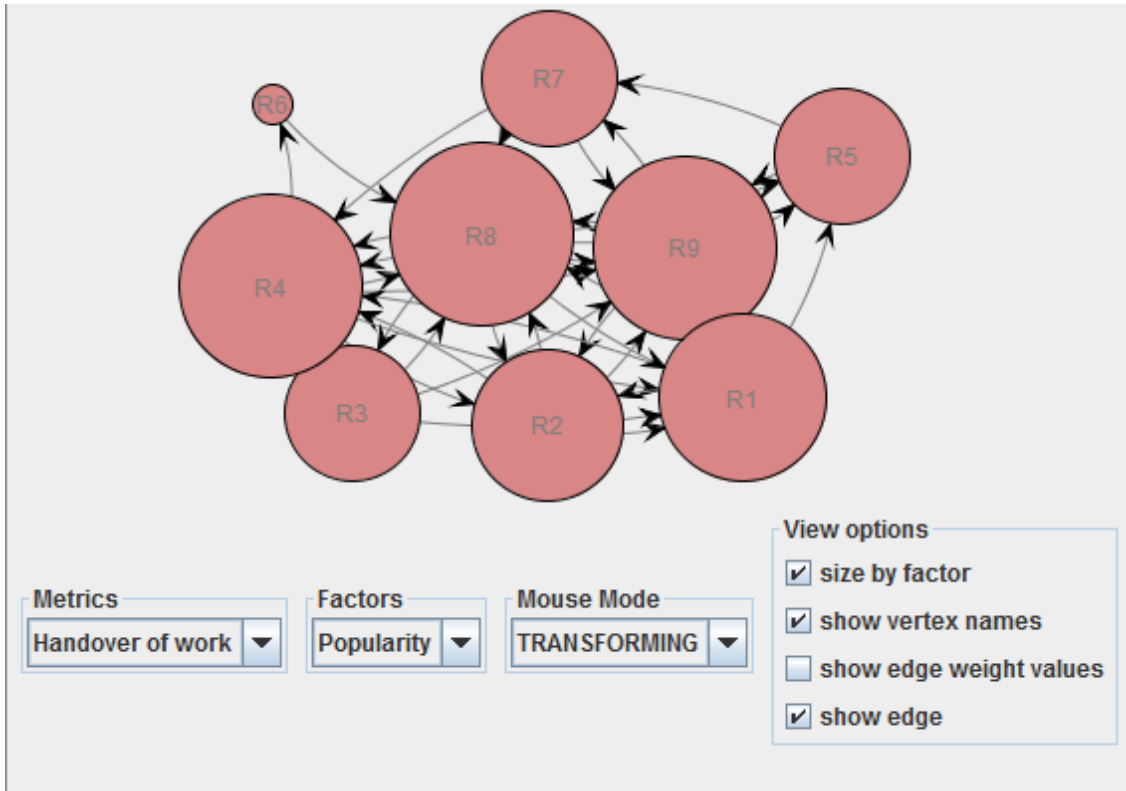


Figura 3.3: Red social asociada al proceso global.

Los subprocesos son representados a través de patrones por cada rama del AV. Para lograr una mayor organización en la descripción de cada subproceso se realizará en forma de un recorrido por niveles.

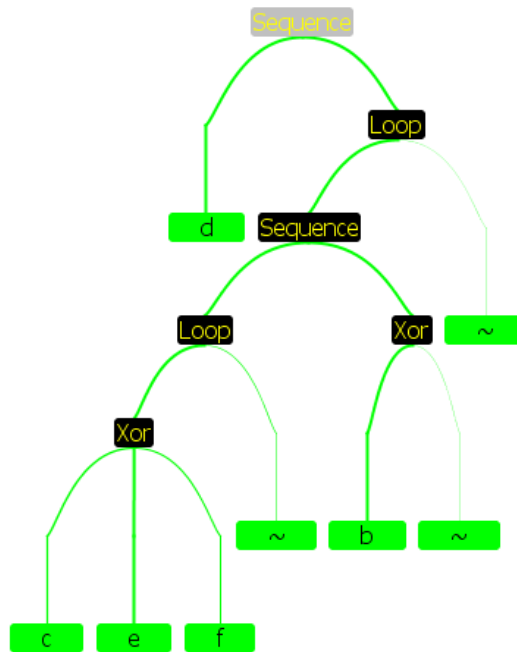


Figura 3.4: Árbol de variantes obtenido a partir del log de eventos del caso de estudio.

La visualización del proceso global permite realizar el análisis de lo general a lo particular. Así, se debe tener una vista global del proceso para disminuir la complejidad del modelo, poniéndose en evidencia uno de los factores para la comprensión del proceso. Pero en este sentido se dificulta el ARS debido a que el cálculo de los factores de ARS se torna complejo. Según la red inferida para el proceso global en la Figura 3.4, se puede apreciar que existen tres recursos con la misma popularidad (R4, R8 y R9).

A continuación, se muestra la red social inferida para cada subprocesso. Para facilitar la comprensión del análisis, este se realizará recorriendo el árbol de variantes por niveles. En la Figura 3.5 se puede apreciar que los recursos más populares del subprocesso son R4, R8 y R9. Aunque disminuyan considerablemente las relaciones entre los recursos siguen siendo ellos los que más se relacionan en el subprocesso. Con estos datos se puede concluir que son estos recursos indispensables para el subprocesso y son los que tienen gran probabilidad de extender el modelo de proceso en el futuro.

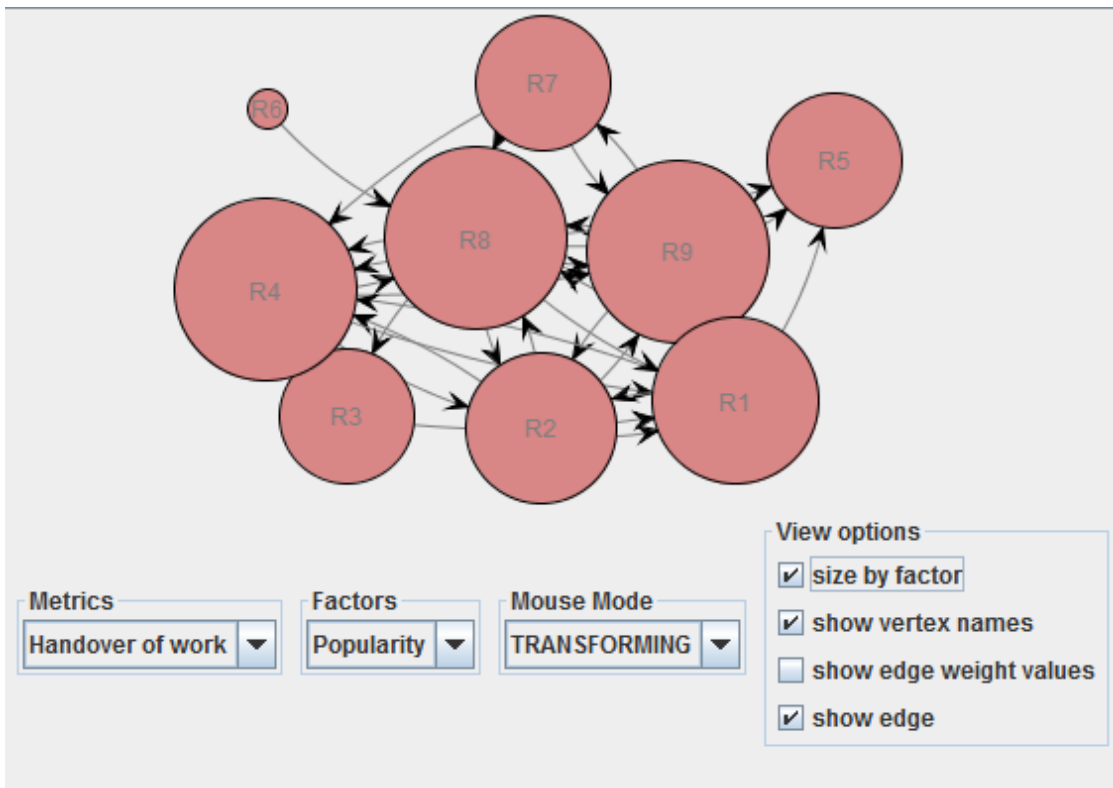


Figura 3.5: Cálculo de popularidad para el subprocesso asociado a la primera ejecución del patrón Loop.

La Figura 3.5 muestra que los recursos R1, R2, R3, R4, R6 y R8 son los de mayor eficiencia en el subprocesso. Esta información brinda la posibilidad de optimizar los recursos del subprocesso y conocer sus flujos de trabajo óptimos para reducir el tiempo de ejecución del subprocesso.

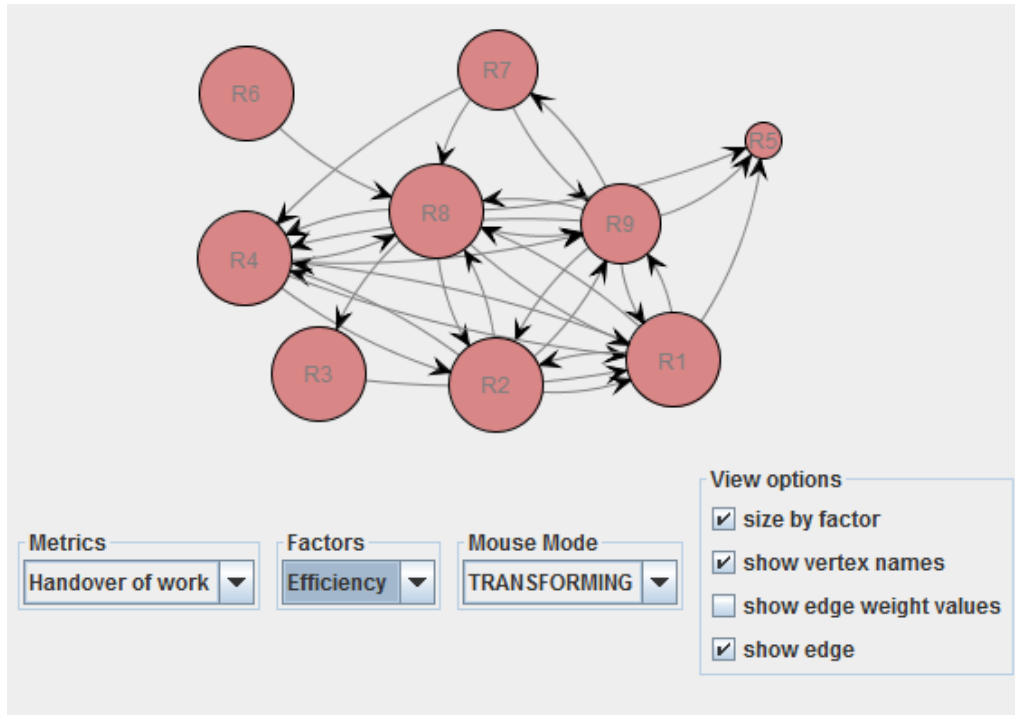


Figura 3.6: Cálculo de eficiencia para el subproceso asociado a la primera ejecución del patrón Loop.

La Figura 3.7 muestra el cálculo del factor *sobrecarga* para el subproceso más abarcador. Como se puede distinguir en la red, el recurso con mayor sobrecarga de trabajo del subproceso es R8. Este dato es de vital importancia en el ARS porque puede que en algún momento sea este recurso el causante de un cuello de botella en el flujo de trabajo del subproceso.

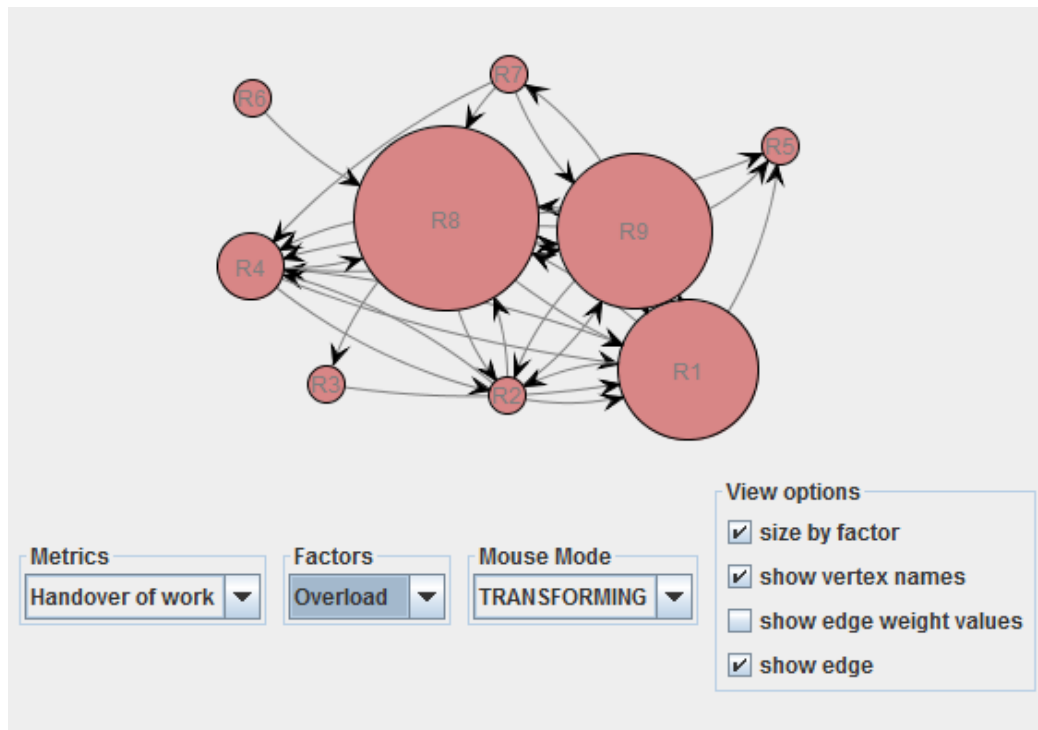


Figura 3.7: Cálculo de sobrecarga para el subproceso asociado a la primera ejecución del patrón Loop.

La red social inferida del subproceso asociado a la segunda aparición del patrón *Sequence* es exactamente igual a la anterior, por lo que el análisis de los factores popularidad, eficiencia y sobrecarga es el mismo.

A continuación, se analiza la red social del subproceso obtenido por la segunda aparición del patrón *Loop*.

La Figura 3.8 muestra la red social asociada al subproceso obtenido por la segunda ejecución del patrón *Loop*. De esta forma se puede decir que los recursos más populares para este subproceso son R1, R8 y R9.

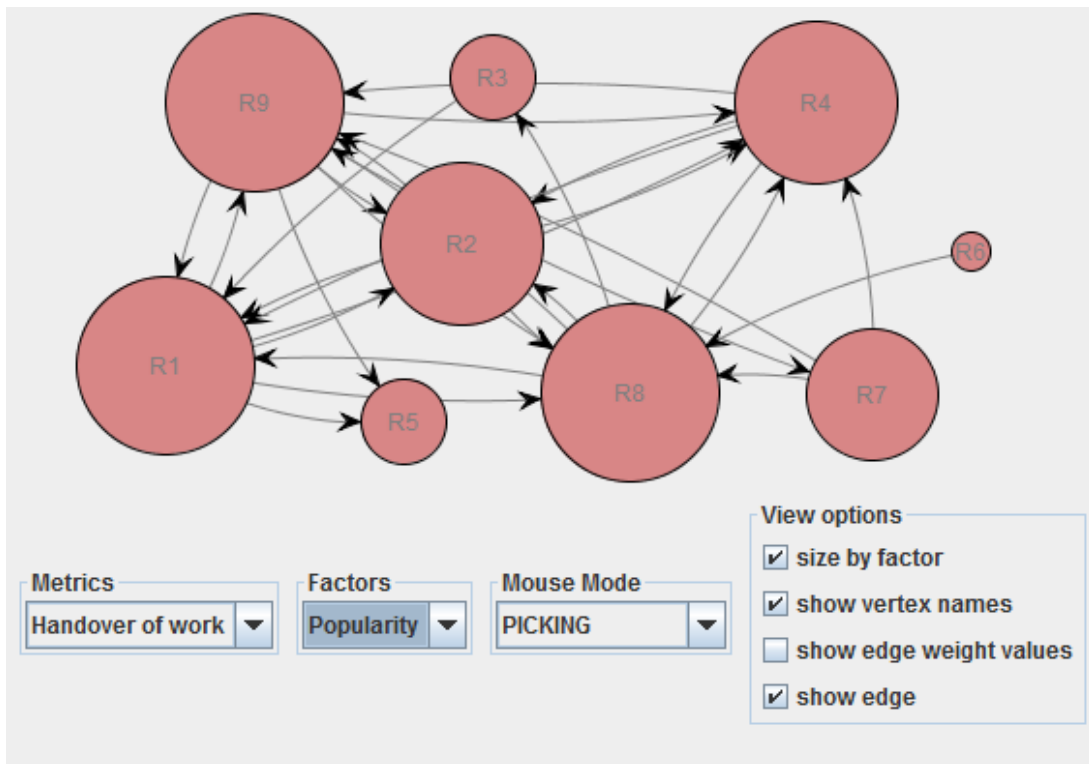


Figura 3.8: Cálculo de popularidad para el subproceso asociado a la segunda ejecución del patrón *Loop*.

La Figura 3.9 muestra el comportamiento de la red social según la eficiencia. Se puede apreciar que los nodos con mayor eficiencia son R1, R2, R3, R4, R6 y R8.

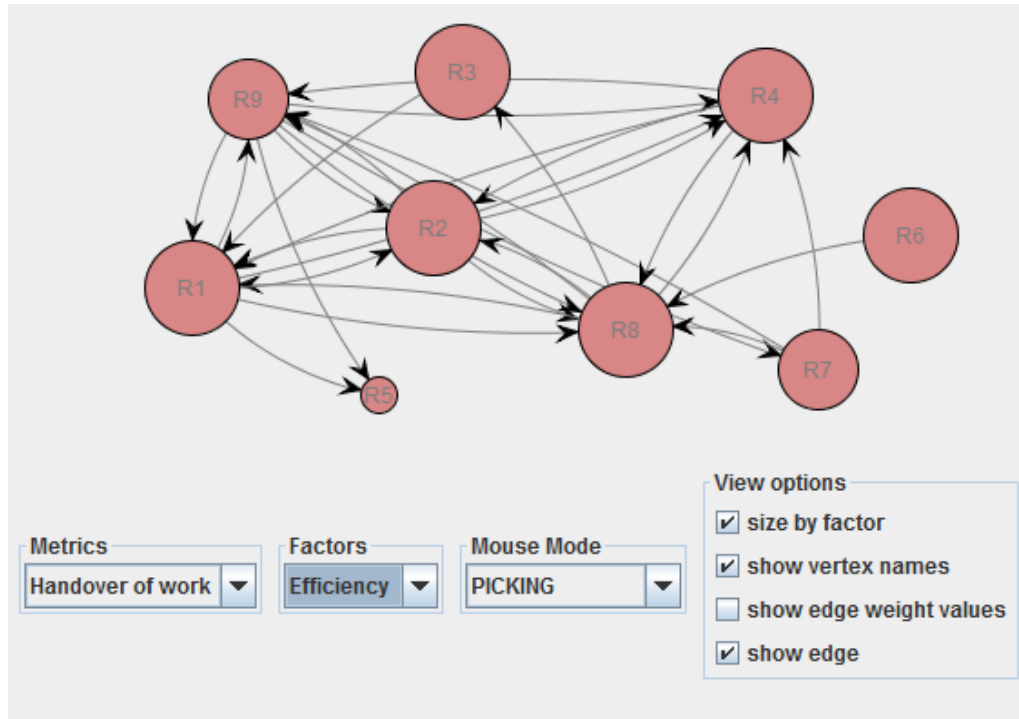


Figura 3.9: Cálculo de eficiencia para el subproceso asociado a la segunda ejecución del patrón Loop.

La Figura 3.10 muestra la sobrecarga de los recursos para el subproceso asociado a la segunda ejecución del patrón Loop. Se puede concluir que los recursos con mayor sobrecarga de este subproceso son R8 y R9.

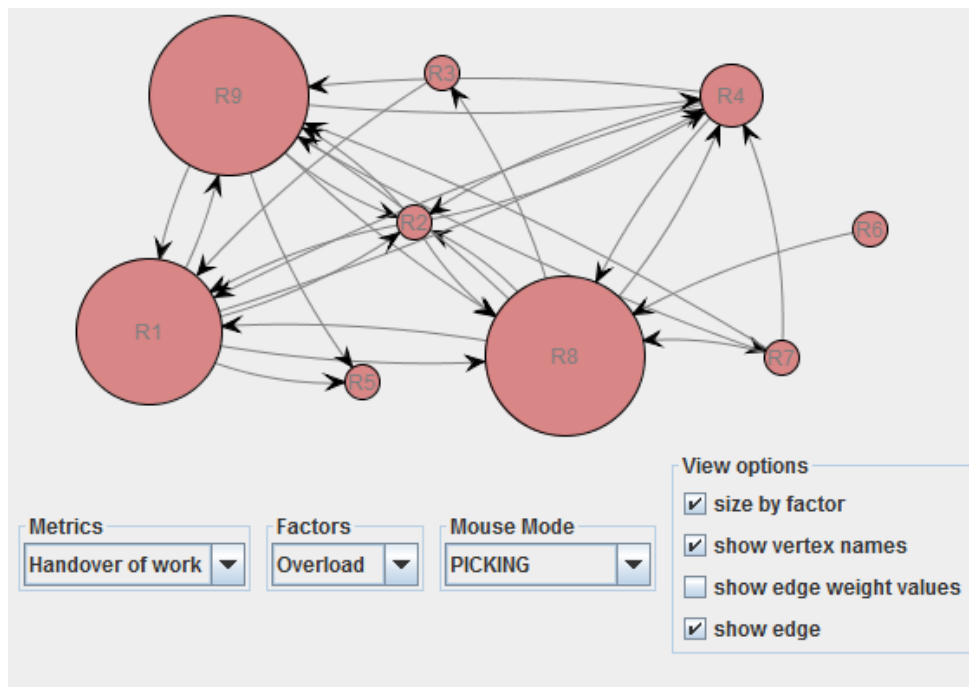


Figura 3.10: Cálculo de sobrecarga para el subproceso asociado a la segunda ejecución del patrón Loop.

El siguiente análisis se realiza para el subproceso obtenido de la primera ejecución del patrón Xor.

La red social del subproceso obtenido por la primera ejecución del patrón Xor que se muestra en la Figura 3.11 determina que los nodos con mayor popularidad son R1, R8 y R9. Note que a medida que se descompone el proceso global en subprocesos se decrementa la cantidad de recursos y las relaciones entre ellos. La red muestra además un recurso que no tiene relación con ningún otro lo que significa que ejecuta tareas por él mismo sin depender de nadie más.

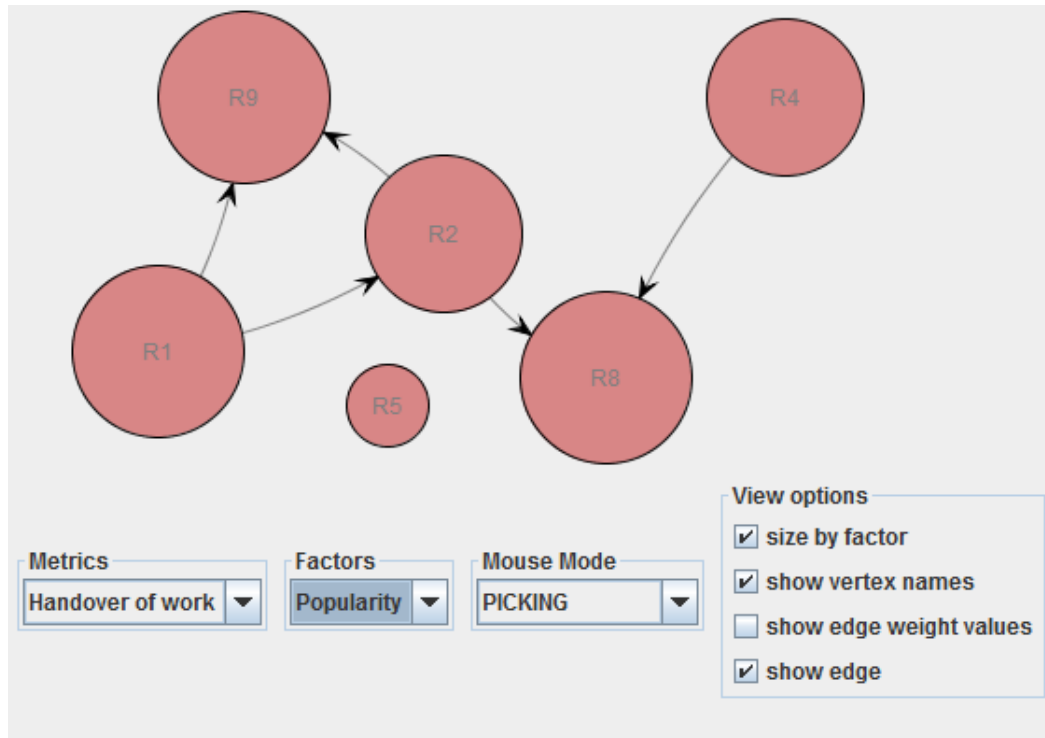


Figura 3.11: Cálculo de popularidad para el subproceso asociado a la primera ejecución del patrón Xor.

En la red social de la Figura 3.12 se muestran los recursos más eficientes en el flujo de trabajo del subproceso obtenido por la primera ejecución del patrón Xor. Se puede decir entonces, que los recursos con mayor eficiencia son R1, R2 y R4.

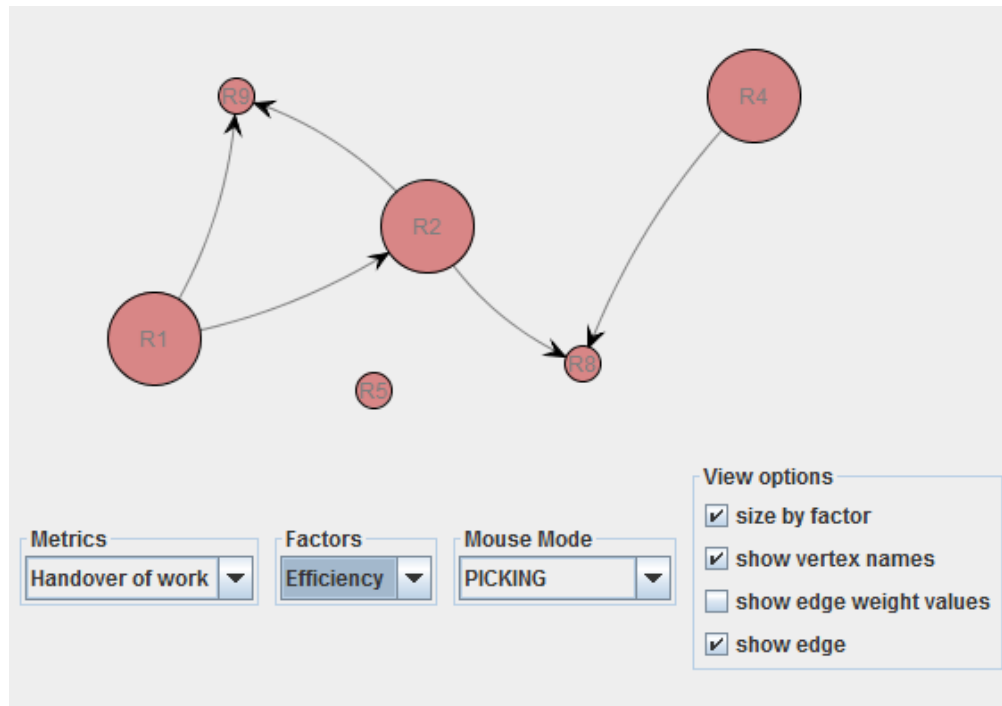


Figura 3.12: Cálculo de eficiencia para el subproceso asociado a la primera ejecución del patrón Xor.

En la Figura 3.13 se muestra la red social asociada al proceso obtenido por la primera ejecución del patrón Xor. La red brinda información relevante acerca del recurso con mayor sobrecarga. R2 es el que más se destaca en cuanto a este factor. Se puede apreciar claramente que es el que más actúa como intermediador entre otros dos recursos.

En el subproceso obtenido por la segunda ejecución del patrón Xor sucede lo mismo que en el subproceso obtenido por la segunda ejecución del patrón Sequence. Esto es debido a la introducción de una actividad invisible. Se puede concluir que el análisis para este subproceso es exactamente igual al del subproceso obtenido por la segunda ejecución del patrón Loop.

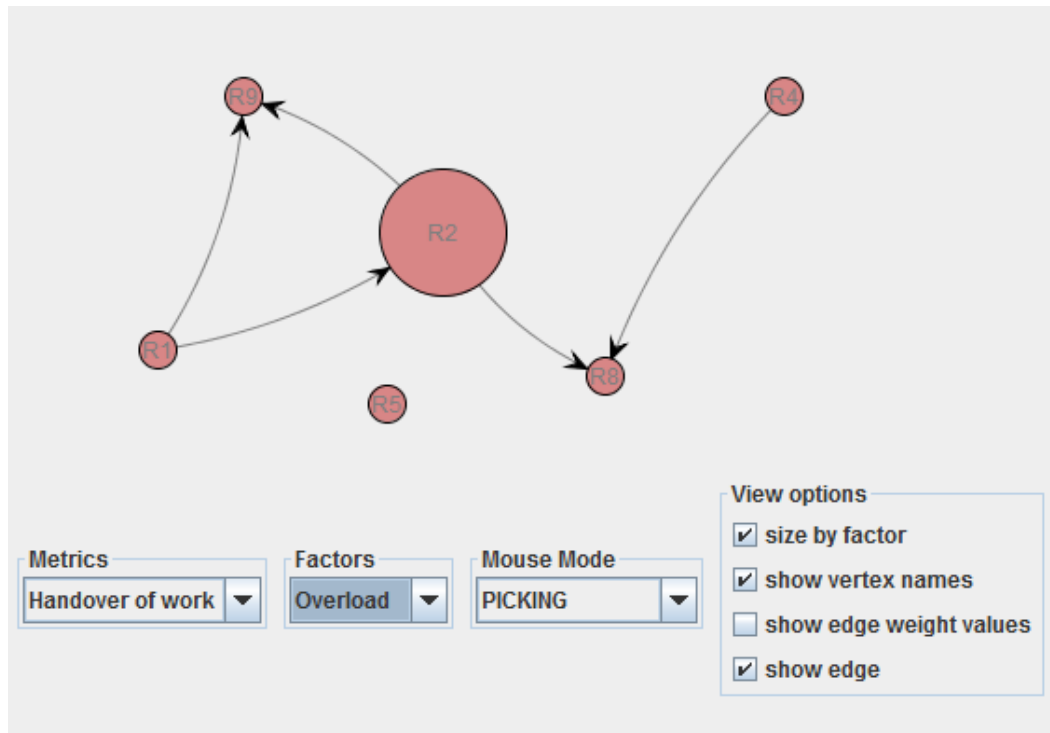


Figura 3.13: Cálculo de sobrecarga para el subproceso asociado a la primera ejecución del patrón Xor.

Por todo lo anterior se puede afirmar que el resultado obtenido al aplicar el algoritmo propuesto permitió el cálculo de los factores popularidad, eficiencia y sobrecarga, a partir de las redes sociales inferidas para los subprocesos que componen a un proceso complejo, lo cual garantizó una mejora en la comprensión de dicho proceso en su dimensión social.

3.4. Conclusiones del capítulo

Con los resultados obtenidos a partir del trabajo realizado se puede concluir que:

- El cálculo del factor popularidad en las redes sociales inferidas a partir de los subprocesos, permitió identificar los recursos indispensables del proceso, así como aquellos que mayor probabilidad tienen de extender en modelo de proceso en un futuro.
- Al realizar el cálculo de la eficiencia, se logró identificar los flujos de trabajo óptimos, lo cual permite reducir el tiempo de ejecución del subproceso.
- El cálculo del factor sobrecarga en las redes sociales inferidas, mostró los recursos con mayor sobrecarga de trabajo, lo cual constituye una alta probabilidad de que estos recursos provoquen cuellos de botellas.
- El plugin desarrollado permitió validar el algoritmo propuesto, mejorando la comprensión de procesos en su dimensión social, a partir del cálculo de los factores popularidad, eficiencia y sobrecarga.

Conclusiones generales

Como resultado de la presente investigación se obtuvo un algoritmo de inferencia de redes sociales a partir de eventos con alto número de interacciones entre los recursos, mejorando la comprensión de procesos complejos en su dimensión social; el mismo garantiza extensibilidad. En base a los resultados obtenidos se arribó a las siguientes conclusiones:

1. El cálculo de las medidas de centralidad en redes sociales permite realizar análisis del comportamiento de los recursos en un proceso determinado; sin embargo, la comprensión de dicho proceso se ve dificultada cuando existe un alto número de interacciones sus recursos, provocando que el análisis de dichas métricas también se vea afectado.
2. Las métricas existentes para inferir redes sociales a partir de subprocesos, permiten obtener información relevante acerca de las interacciones de los recursos en un proceso determinado.
3. La extensibilidad del algoritmo presentado radica en la posibilidad de adicionar más factores para el ARS sin afectar los demás componentes del mismo, enriqueciendo de esta forma los análisis posteriores.
4. La realización del caso de estudio evidenció la efectividad del algoritmo propuesto, mostrando que dicho algoritmo permite identificar recursos indispensables en un proceso, flujos de trabajo óptimos, recursos con mayor probabilidad para extender el modelo de proceso y recursos con sobrecarga de trabajo. A partir de los resultados obtenidos se justifica el uso del algoritmo propuesto para inferir redes sociales a partir de eventos con alto número de interacciones entre los recursos, facilitando la comprensión de procesos complejos en su dimensión social.

Recomendaciones

Como continuidad de la presente investigación el autor recomienda las siguientes acciones:

1. Añadir parámetros de visualización a la red inferida para enriquecer la comprensión del proceso, tales como: mostrar las aristas con mayor peso, mostrar las actividades en las que interviene un recurso determinado.
2. Definir otros factores de relevancia para el análisis de redes sociales que contribuyan a mejorar la comprensión de procesos.
3. Extender el algoritmo propuesto hacia otras variantes de descomposición de procesos en subprocesos.

Referencias bibliográficas

1. PÉREZ GONZÁLEZ, Daniel and OTHERS. Contribución de las tecnologías de la información a la generación de valor en las organizaciones: un modelo de análisis y valoración desde la gestión del conocimiento, la productividad y la excelencia en la gestión [online]. Universidad de Cantabria, 2006. [Accessed 3 June 2016]. Available from: <http://repositorio.unican.es/xmlui/handle/10902/1173>
2. AGRAWAL, Rakesh, GUNOPULOS, Dimitrios and LEYMANN, Frank. Mining process models from workflow logs. In : SCHEK, Hans-Jörg, ALONSO, Gustavo, SALTOR, Felix and RAMOS, Isidro (eds.), *Advances in Database Technology — EDBT'98* [online]. Springer Berlin Heidelberg, 1998. p. 467–483. *Lecture Notes in Computer Science*, 1377. [Accessed 31 May 2016]. ISBN 978-3-540-64264-0. Available from: <http://link.springer.com/chapter/10.1007/BFb0101003>
3. AALST, VAN DER WMP WIL and WEIJTERS, AJMM TON. Process mining : a research agenda. *Computers in Industry*. 2004. Vol. 53, no. 3, p. 231null.
4. VAN DER AALST, Wil, ADRIANSYAH, Arya and VAN DONGEN, Boudewijn. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2012. Vol. 2, no. 2, p. 182–192.
5. SANTOS, F. R. *El concepto de red social*. Reis, 1989.
6. ROS MARTÍN, M. Evolución de los servicios de redes sociales en internet. . 2009. Vol. 18, no. *El profesional de la información*, p. 552–558.
7. MANCHEGO, Alva and EMILIO, Fernando. Sistema de información de detección de plagio en documentos digitales usando el método Document Fingerprinting. [online]. 9 May 2011. [Accessed 31 May 2016]. Available from: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/374>
8. BARÓN, Mónica Andrea Niño and SALINAS, Sonia Ordóñez. Identificación de relaciones entre los nodos de una red social. *Ingeniería* [online]. 30 June 2013. Vol. 18, no. 1. [Accessed 31 May 2016]. DOI 10.14483/udistrital.jour.reving.2013.1.a03. Available from: <http://revistas.udistrital.edu.co/ojs/index.php/reving/article/view/4896>
9. PÉREZ, Damián. Método para el diagnóstico de procesos de negocio a partir de registros de eventos con ruido y ausencia de información. UCI, 2015.
10. TUCKER, A.B, CHAPMAN and HALL/CRC. *Computer Science Handbook*. Second Edition. 2004.
11. WASSERMAN, Stanley and FAUST, Katherine. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994. ISBN 978-0-521-38707-1.
12. KOUTRIKA, Georgia, EFFENDI, Frans Adjie, GYÖNGYI, Zoltán, HEYMANN, Paul and GARCIA-MOLINA, Hector. Combating Spam in Tagging Systems: An Evaluation. *ACM Trans. Web*. October 2008. Vol. 2, no. 4, p. 22:1–22:34. DOI 10.1145/1409220.1409225.
13. DI CARO, Luigi, CANDAN, K. Selçuk and SAPINO, Maria Luisa. Using Tagflake for Condensing Navigable Tag Hierarchies from Tag Clouds. In : *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* [online]. New York, NY, USA : ACM, 2008. p. 1069–1072. [Accessed 31 May 2016]. KDD '08. ISBN 978-1-60558-193-4. Available from: <http://doi.acm.org/10.1145/1401890.1402021>
14. BEAUCHAMP, Murray A. An improved index of centrality. *Behavioral Science*. 1 January 1965. Vol. 10, no. 2, p. 161–163. DOI 10.1002/bs.3830100205.

15. FREEMAN, Linton C. Centrality in social networks conceptual clarification. *Social Networks*. 1 January 1978. Vol. 1, no. 3, p. 215–239. DOI 10.1016/0378-8733(78)90021-7.
16. MENDEZ ANDRADE, WILLIAM GABRIEL and MONTESDEOCA CHUSAN, ADRIAN JOEL. ANÁLISIS DE LOS PROCESOS ADMINISTRATIVOS Y SU INCIDENCIA EN LOS NIVELES DE DISPONIBILIDAD DE LA INFORMACIÓN EN LA EMPRESA PÚBLICA EPUNEMI. ECUADOR : U NIVERSIDAD ESTATAL DE MILAGRO, 2014.
17. WESKE, Mathias. Concepts, Languages, Architectures [online]. vol, 2007. [Accessed 31 May 2016]. Available from: http://otgo.tehran.ir/Portals/0/pdf/Business%20Process%20Management_1.pdf
18. SMITH, Howard and FINGAR, Peter. *Business process management: the third wave*. Meghan-Kiffer Press Tampa, 2003.
19. CURTIS, Bill, KELLNER, Marc I. and OVER, Jim. Process modeling. *Communications of the ACM*. 1992. Vol. 35, no. 9, p. 75–90.
20. VAN DER AALST, W. and OTROS. *Manifiesto sobre Minería de Procesos*. IEEE Task Force on Process Mining. 2011.
21. CAIRNS, Awatef Hicheur, GUENI, Billel, FHIMA, Mehdi, CAIRNS, Andrew, DAVID, Stéphane and KHELIFA, Nasser. Towards Custom-Designed Professional Training Contents and Curriculums through Educational Process Mining. *IMMM14*. 2014. P. 53–58. 00005
22. CALDERS, Toon and PECHENIZKIY, Mykola. Introduction to the special section on educational data mining. *ACM SIGKDD Explorations Newsletter*. 2012. Vol. 13, no. 2, p. 3–6. 00042
23. VAN DER AALST, Wil MP and SONG, Minseok. Mining Social Networks: Uncovering interaction patterns in business processes. In : *Business Process Management* [online]. Springer, 2004. p. 244–260. [Accessed 31 May 2016]. Available from: http://link.springer.com/chapter/10.1007/978-3-540-25970-1_16
24. CELAYA, J. *La empresa en la Web 2.0*. Barcelona, 2000.
25. FURHT, Borko. *Handbook of social network technologies and applications* [online]. Springer Science & Business Media, 2010. [Accessed 31 May 2016]. Available from: <https://www.google.com/books?hl=es&lr=&id=7kCOWGMC46cC&oi=fnd&pg=PR1&dq=Furht,+B.+Handbook+of+social+network+technologies+and+applications,+vol.+1.+Springer,+2010.&ots=PL5SyB6zSk&sig=na1o56a2NAyLQ0-QIWZOMDm5X8o>
26. FRANCO, A. EGONET. [online]. 2009. [Accessed 31 May 2016]. Available from: <http://escoladeredes.net/profiles/blogs/egonet-1>
27. SONG, Minseok and VAN DER AALST, Wil MP. Towards comprehensive support for organizational mining. *Decision Support Systems*. 2008. Vol. 46, no. 1, p. 300–317.
28. XIA, Feng, LIU, Li, LI, Jie, MA, Jianhua and VASILAKOS, Athanasios V. Socially aware networking: A survey. *Systems Journal, IEEE*. 2015. Vol. 9, no. 3, p. 904–921.
29. BARTO, Andrew G. Skill characterization based on betweenness. In : *In Advances in Neural Information Processing Systems 22* [online]. 2009. [Accessed 3 June 2016]. Available from: http://www-all.cs.umass.edu/pubs/2008/simsek_b_NIPS08.pdf00000
30. LATORA, V. and MARCHIORI, M. A measure of centrality based on the network efficiency. *arxiv: con-math 0402050*. . 2004.

31. KROGSTIE, John, SINDRE, Guttorm and JØRGENSEN, Håvard. Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems*. 2006. Vol. 15, no. 1, p. 91–102.
32. MENDLING, Jan, STREMBECK, Mark and RECKER, Jan. Factors of process model comprehension—findings from a series of experiments. *Decision Support Systems*. 2012. Vol. 53, no. 1, p. 195–206.
33. RECKER, Jan C. and DREILING, Alexander. The effects of content presentation format and user characteristics on novice developers' understanding of process models. *Communications of the Association for Information Systems*. 2011. Vol. 28, no. 6, p. 65–84.
34. MENDLING, Jan, REIJERS, Hajo A. and CARDOSO, Jorge. What makes process models understandable? In : *Business Process Management* [online]. Springer, 2007. p. 48–63. [Accessed 31 May 2016]. Available from: http://link.springer.com/chapter/10.1007/978-3-540-75183-0_400251
35. STAMPER, Ronald. Signs, information, norms and systems. [online]. 1996. [Accessed 31 May 2016]. Available from: <http://philpapers.org/rec/STASIN-2>
36. KIRSCHNER, Paul A. Cognitive load theory: Implications of cognitive load theory on the design of learning. *Learning and instruction*. 2002. Vol. 12, no. 1, p. 1–10.
37. KOTOVSKY, Kenneth, HAYES, John R. and SIMON, Herbert A. Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive psychology*. 1985. Vol. 17, no. 2, p. 248–294.
38. MILLER, George A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*. 1956. Vol. 63, no. 2, p. 81.
39. VAN DER AALST, Wil MP, RUBIN, Vladimir, VERBEEK, H. M. W., VAN DONGEN, Boudewijn F., KINDLER, Ekkart and GÜNTHER, Christian W. Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*. 2010. Vol. 9, no. 1, p. 87–111.
40. YZQUIERDO, H. Modelo para la estimación de información ausente en las trazas usadas en la minería de proceso. La Habana, 2013. 00004
41. BRANDES, Ulrik. A faster algorithm for betweenness centrality*. *Journal of mathematical sociology*. 2001. Vol. 25, no. 2, p. 163–177.
42. UNIVERSIDAD ESTATAL DE MILAGRO UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERIA. [online]. 2012. [Accessed 31 May 2016]. Available from: <http://docplayer.es/11189495-Universidad-estatal-de-milagro-unidad-academica-de-ciencias-de-la-ingenieria.html>
43. WEY, Tina, BLUMSTEIN, Daniel T., SHEN, Weiwei and JORDÁN, Ferenc. Social network analysis of animal behaviour: a promising tool for the study of sociality. *Animal behaviour*. 2008. Vol. 75, no. 2, p. 333–344.
44. Faster Betweenness Centrality Based on Data Structure Experimentation. [online]. 2013. [Accessed 3 June 2016]. Available from: <http://www.sciencedirect.com/science/article/pii/S1877050913003463>
45. YANG, Hedong, VAN DONGEN, B., TER HOFSTEDÉ, A., WYNN, M. and WANG, Jianmin. Estimating completeness of event logs. *BPM Center Report*, 12-04-2012 [online]. 2012. [Accessed 31 May 2016]. Available from: <http://bpmcenter.org/wp-content/uploads/reports/2012/BPM-12-04.pdf>

46. VAN DER AALST, W. M. P. Do Petri nets provide the right representational bias for process mining? In : Workshop Applications of Region Theory 2011 (ART 2011) [online]. Citeseer, 2011. p. 85–94. [Accessed 31 May 2016]. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.5330&rep=rep1&type=pdf>
47. VAN DER AALST, Wil MP, VAN DONGEN, Boudewijn F., GÜNTHER, Christian W., ROZINAT, Anne, VERBEEK, Eric and WEIJTERS, Ton. ProM: The Process Mining Toolkit. BPM (Demos). 2009. Vol. 489, p. 31. 00122
48. BOSE, RP Jagadeesh Chandra, VERBEEK, Eric HMW and VAN DER AALST, Wil MP. Discovering hierarchical process models using prom. In : IS Olympics: Information Systems in a Diverse World [online]. Springer, 2011. p. 33–48. [Accessed 31 May 2016]. Available from: http://link.springer.com/10.1007%2F978-3-642-29749-6_3
49. MANS, Ronny S., VAN DER AALST, Wil MP, VANWERSCH, Rob JB and MOLEMAN, Arnold J. Process mining in healthcare: Data challenges when answering frequently posed questions. In : Process Support and Knowledge Representation in Health Care [online]. Springer, 2013. p. 140–153. [Accessed 31 May 2016]. Available from: http://link.springer.com/chapter/10.1007/978-3-642-36438-9_10
50. BECK, Kent. Extreme programming explained: embrace change [online]. addison-wesley professional, 2000. [Accessed 3 June 2016]. Available from: https://www.google.com/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Beck,+K.+%26+Andres,+C.+Extreme+programming+explained:+embrace+change&ots=j9ylrshQxq&sig=hLla5jc0Hls2e4n_64ermYyQy4c