Interoperabilidad de flujos de trabajo intensivos en datos en Industria 4.0: caso de estudio

Rubén Salado-Cid, José Molino, José Raúl Romero
Dpto. de Informática y Análisis Numérico, Universidad de Córdoba
{rsalado, i32moorj, jrromero}@uco.es

***Resumen*—Con la transformación digital del sector industrial propuesta por la Industria 4.0, se ha producido un gran incremento del volumen de datos gestionado por las empresas, situando así a la ciencia de datos como un pilar tecnológico fundamental. La complejidad en el uso de las técnicas computacionales ha promovido el desarrollo de herramientas basadas en flujos de trabajo intensivos en datos para definir el conocimiento del experto a un alto nivel de abstracción, ocultando los detalles de computación a bajo nivel. Sin embargo, la falta de estándares imposibilita la reutilización de este conocimiento, lo que obliga al aprendizaje de varias herramientas, según los requisitos de cada dominio, o bien a quedar limitados por las características de una de ellas. En este trabajo se presenta una propuesta para la interoperabilidad de flujos intensivos en datos mediante la implementación de una serie de transformaciones y su automatización accesible a través de servicios web. Para ello, se analiza detalladamente cada sistema, así como los elementos que definen sus flujos, antes de declarar las equivalencias. Después se ilustra la propuesta con un caso de estudio de un flujo industrial, definido en Taverna, y la obtención de su equivalente en Kepler.**

***Index Terms*—flujos de trabajo intensivos en datos, ciencia de datos, interoperabilidad**

I. Introducción

La transformación digital que se está llevando a cabo en todos los sectores industriales está cambiando la forma en la que las empresas operan, teniendo que adaptar sus entornos de producción a un nuevo paradigma industrial, conocido como Industria 4.0. La Industria 4.0 tiene como base la utilización masiva de sensores, dispositivos y de sistemas de información con el fin de mejorar y optimizar los productos y métodos productivos, mediante el procesamiento de grandes cantidades de datos. De hecho, los datos ocupan un lugar principal en esta nueva revolución industrial y sitúan a la ciencia de datos [1] como un pilar tecnológico sobre el que se sustenta este nuevo paradigma.

Ciertamente, la ciencia de datos proporciona un gran número de herramientas y técnicas computacionales [2], [3] que permiten procesar y analizar enormes cantidades de datos, lo que resultaba inalcanzable con las técnicas tradicionales de procesamiento de datos. Sin embargo, la complejidad inherente a este tipo de tecnologías hace necesario el uso de propuestas que faciliten la representación del conocimiento capturado por los procesos computacionales intensivos en datos y permitan la automatización de su ejecución. Esto es, se debe permitir definir *qué* se quiere hacer independientemente del *cómo*.

Para lograr este propósito surgen los sistemas de gestión de flujos de trabajo (WfMS, *workflow management systems*) [5]. Un flujo de trabajo [4] permite capturar y modelar el conocimiento del experto como una secuencia de acciones o tareas que trabajan en coordinación para llevar a cabo un objetivo determinado. Esta secuencia de acciones es definida a un alto nivel de abstracción, más cercano al dominio del experto, haciendo transparentes los aspectos de bajo nivel de computación. Los WfMS ya se utilizan en numerosos ámbitos industriales como pueden ser la manufacturación [6], ventas on-line (*retail*) [7] o telecomunicaciones [8]. Sin embargo, ninguno de estos sistemas formaliza la manera en que los flujos de trabajo son representados, siendo cada representación propia de la herramienta. Esto conlleva la imposibilidad de reutilizar el conocimiento como activo e incluso limita la expresividad y alcance del mismo, según las funcionalidades y operaciones propias del WfMS. Así, distintos WfMS frecuentemente se especializan en dominios concretos, por lo que existe la necesidad de desarrollar mecanismos de interoperabilidad entre distintos flujos intensivos en datos [9].

Recientemente se propuso un lenguaje común para la definición de flujos de trabajo intensivos en datos [10], si bien se encuentra en un estado muy inicial y no hay soporte. Por ello, con el objetivo de promover la reutilización del conocimiento y operaciones de extracción del mismo sobre datos industriales, sería necesario poder ofrecer interoperabilidad sin requerir la adaptación de las herramientas existentes o de los flujos de trabajo ya definidos, algo poco realista en los tiempos que maneja la industria.

Por ello, en este trabajo se discute una primera propuesta para alcanzar la interoperabilidad entre distintos flujos de trabajo intensivos en datos ya existentes. En primer lugar se ha realizado un estudio de las características proporcionadas por las principales herramientas basadas en flujos de datos. Este estudio permite identificar los elementos más importantes en la definición de los flujos de trabajo de cada WfMS, detectar cuáles son comunes y cuáles exclusivos, así como conocer la forma en la que estos se representan. A partir de este estudio se formaliza la manera en que las herramientas actuales expresan esos elementos con el fin de implementar una serie de transformaciones entre ellos. Para este propósito, se utiliza un lenguaje intermedio, presentado en [12], independiente de cualquier WfMS. Dicha formalización y sus transformaciones

Este trabajo ha sido cofinanciado por el Ministerio de Economía y Competencia del Gobierno de España [proyecto TIN2017-83445-P]

se lleva a cabo de forma transparente en una capa inferior utilizando *ingeniería dirigida por modelos* [11]. Se omitirán los detalles más específicos por situarse fuera del ámbito de este trabajo y por motivos de espacio, si bien, esta capa habilitará la aplicación de las técnicas de ciencias de datos sobre un mayor abanico de áreas de la Industria 4.0. Finalmente, se propone un prototipo de servicio web como demostración de la solución propuesta. Este servicio puede ser integrado en herramientas externas a través de su API REST.

La propuesta se ilustra mediante un caso de estudio en el que un flujo de trabajo intensivo en datos aplicado a Industria 4.0, que ha sido definido en una herramienta específica, es transformado a otro flujo de trabajo equivalente y compatible con otro WfMS distinto pero manteniendo los elementos de conocimiento (procesos y artefactos) originales.

En el resto del artículo se organiza como sigue. En la Sección II se introducen los sistemas de gestión de flujos de trabajo y se destacan algunos de los más relevantes actualmente. En la Sección III se analizan las características más destacadas para representar los flujos de trabajo. A continuación, en la Sección IV se detalla la propuesta para alcanzar la interoperabilidad, cuya demostración es realizada en la Sección V. Finalmente, los resultados obtenidos son discutidos en la Sección VI, y se presentan las conclusiones y líneas de trabajo futuro en la Sección VII.

II. Estado del arte

Los sistemas de gestión de flujos de trabajo son herramientas que permiten la definición y gestión de la ejecución del conjunto de tareas que conforman un proceso computacional, junto con sus dependencias. Para ello, habitualmente estos sistemas proporcionan un editor gráfico que permite la representación visual a alto nivel de los flujos de trabajo. Estos flujos suelen organizarse como un grafo, donde una serie de nodos o componentes son interconectados para definir el orden de ejecución según la lógica de negocio. Una vez definido, la ejecución es gestionada por uno o más motores de ejecución especializados que gestionan los recursos computacionales disponibles e invocan automáticamente las tareas computacionales definidas de forma óptima y transparente.

Actualmente existen una gran variedad de WfMS utilizados en ciencia de datos. Uno de las más destacadas es KNIME [13], de código libre, que proporciona acceso a un gran número de algoritmos configurables para la integración, transformación y visualización de datos. KNIME se utiliza satisfactoriamente en banca, industria o en investigación y desarrollo, entre otros dominios. Otra herramienta muy similar es RapidMiner [14] que proporciona una serie de procesos de minería de datos y de aprendizaje automático para permitir la carga, transformación, preprocesamiento y visualización de datos. Por citar algunos ejemplos, RapidMiner se ha aplicado a la optimización de procesos computacionales en marketing y en gestión política, o a la mejora en cadenas de producción.

Existen otros WfMS más orientados a la aplicación de procesos de ciencia de datos, y que, hasta el momento, se han venido utilizando en dominios de carácter más científico. Taverna [15] es una herramienta que proporciona los recursos necesarios para definir y ejecutar flujos de trabajo con el fin de aceptar o rechazar hipótesis científicas. Por ello, es un WfMS utilizado en un gran rango de dominios, como la industria biomédica y la informática química. Asimismo, Kepler [16] es otro WfMS que ofrece una gran cantidad de algoritmos para el procesamiento de datos, desde su carga hasta la transformación y visualización de los mismos. Kepler ha sido utilizado en dominios como la bioinformática. Si bien el tipo de industria que aplica estas herramientas es de carácter experimental, no existe limitación alguna que impida su uso en otros dominios intensivos en datos, incluyendo los propios de la Industria 4.0.

Como se indicaba anteriormente, la falta de estandarización en este tipo de tecnologías intensivas en datos hace que las características de cada herramienta puedan ser o no compatibles con las demás, provocando una pérdida notable de elementos de conocimiento y procedimientos ya estructurados debido a la imposibilidad de su reutilización.

III. Análisis de las representaciones de los flujos de trabajo

La Tabla I muestra un resumen de las principales características analizadas en los principales WfMS, como son KNIME, RapidMiner, Taverna y Kepler. Estas propiedades se han agrupado en categorías según su funcionalidad: (a) todas las características referidas a la composición y estructuración de un proceso computacional como *flujo de trabajo*; (b) los distintos aspectos que definen los procesos de extracción de conocimiento que pueden llevarse a cabo en los *nodos de ejecución*; (c) el tipo de *metainformación* descriptiva asociada a los mismos; y (d) las características referidas al almacenamiento y gestión de datos del flujo de trabajo (*serialización*).

Las características relacionadas con la composición de un *flujo de trabajo* permiten conocer su capacidad de configuración y adaptación a distintos dominios. Sería deseable en términos de tiempo y coste que un proceso genérico de ciencia de datos pudiera ser utilizado en diferentes dominios industriales. Así, todas las herramientas analizadas estructuran sus flujos de trabajo como grafos dirigidos, cuyos nodos representan tareas a ejecutar y las conexiones dependencias de datos entre ellos, es decir, cuando un nodo depende de los datos generados por otros nodos para poder ejecutarse. No obstante, eventualmente un nodo podría depender únicamente de la ejecución previa de otros nodos (flujo de control) o del cumplimiento de determinadas condiciones que pueden ser definidas mediante una serie de estructuras de control.

Por otro lado, los nodos definen el comportamiento de las operaciones de procesamiento. Este comportamiento viene determinado por una serie de parámetros que permiten su adaptación al dominio. Toda esta parametrización puede venir preestablecida por el WfMS, así como la apariencia de los distintos elementos, con el objetivo de centrarse en la definición del proceso de extracción. Igualmente, los usuarios pueden modificar la apariencia de los nodos para favorecer la interpretabilidad del flujo por los distintos agentes involucrados.

	KNIME	RapidMiner	Taverna	Kepler
Flujo de trabajo				
Estructura	Grafo dirigido	Grafo dirigido	Grafo dirigido	Grafo dirigido
Flujo de datos	Sí	Sí	Sí	Sí
Flujo de control	Sí	Sí	Sí	Sí
Estructuras de control	Condición, bucle, switch	Condición, bucle	Condición, bucle	Condición, bucle, switch
Configuración de nodos	Sí	Sí	Sí	Sí
Configuración de apariencia	Sí	Sí	No	Sí
Parámetros por defecto	Sí	Sí	No	Sí
Nodos de ejecución				
Tipos de datos	Básicos, complejos	Básicos, complejos	Básicos	Básicos, complejos
Fuentes de datos	Memoria, fichero, base de datos	Memoria, fichero, base de datos	Memoria	Memoria, fichero, base de datos
Tareas	Java, JavaScript	Java, Groovy	Java, R	Java, R, Python, Matlab, Groovy, JavaScript
Invocación de servicios	REST	REST	SOAP	REST, SOAP
Flujos de trabajo anidados	No	Sí	Sí	Sí
Metainformación				
Descripción de nodos	Sí	No	No	No
Contexto	Sí	Sí	Sí	Sí
Serialización				
Formato	XML	XML	XML	XML
Estructura	Múltiples ficheros y directorios	Único fichero	Único fichero	Único fichero
Accesible	Sí	Sí	Sí	Sí
Compresión	No	No	No	No

Tabla I
Resumen de características de los principales WfMS

En aquellos procesos en los que se deben aplicar procedimientos computacionales costosos sobre datos bien digitalizados o directamente procedentes de sistemas industriales (p.ej. sistemas de manufactura industrial, sensores, etc.), los *nodos de ejecución* definen cómo estos datos serán transformados para generar otros nuevos que, a su vez, sean consumidos por otros nodos hasta extraer conocimiento de ello. Cada herramienta define los tipos de datos específicos que pueden ser manipulados, desde tipos básicos (cadenas, numéricos, booleanos, etc.) hasta tipos complejos (secuencias, imágenes, estructuras complejas procedentes de sistemas externos, etc.). Igualmente, estos datos pueden proceder de distintas fuentes.

Existen distintos tipos nodos de ejecución que pueden ser representados por los flujos de trabajo. Por una parte, las tareas son operaciones cuya ejecución viene determinada por un algoritmo en algún lenguaje de programación. Por otra parte, es habitual encontrar nodos que permiten la invocación remota de servicios web, permitiendo la conexión entre sistemas remotos, por ejemplo, con otros sistemas industriales o con proveedores externos. En ocasiones, la ejecución es definida por el anidamiento de flujos de trabajo. Esta característica, además de incrementar el nivel de abstracción de los procesos, es un potente mecanismo de reutilización de conocimiento y *know-how* dentro de la propia organización.

Respecto a la *metainformación*, ésta no altera la operación del WfMS, si bien favorece establecer buenas prácticas que faciliten la transferencia del *know-how*, por ejemplo, en equipos con elevada rotación o en la incorporación de nuevos miembros a la organización que deban adquirir destrezas respecto a la organización de los métodos productivos y el procesamiento de sus datos. En general, esta metainformación está relacionada con la descripción del funcionamiento de un nodo específico, del autor o del proyecto.

Finalmente, en términos de interoperabilidad es crítico establecer mecanismos estándares para la *serialización* de los flujos de trabajo. Es habitual que los WfMS utilicen XML, definiendo esquemas propios, para almacenar la estructura y comportamiento de los flujos de trabajo intensivos en datos. La distribución de estos ficheros, su accesibilidad y codificación varía de una herramienta a otra.

IV. Implementación de los mecanismos de interoperabilidad

El análisis de las características de los distintos WfMS (véase Sección III) establece las bases para el desarrollo de las transformaciones que permitan reconvertir los elementos de flujos de trabajo de un sistema en sus equivalentes dentro de la especificación de otro WfMS, deseablemente de forma bidireccional. Esta transformación debe realizarse de manera automática y transparente al experto de negocio, permitiendo reutilizar activos de conocimiento previamente producidos.

IV-A. Implementación de las transformaciones

Para permitir la transformación multisistema, se hace uso de un lenguaje específico del dominio de la ciencia de datos, independiente de cualquier WfMS. Este lenguaje está bien definido y formalizado [12], proporcionando una declaración precisa de cada uno de los elementos que componen un proceso computacional intensivo en datos. Su uso permite reducir drásticamente el número de conversiones necesarias, implementando únicamente las equivalencias del formato de un WfMS A al lenguaje específico de dominio intermedio, y viceversa, de un WfMS B a este lenguaje, y viceversa, y así sucesivamente. En caso de no utilizar tal lenguaje intermedio, la interoperabilidad debería estudiarse por cada par de WfMS. Así, cada vez que un nuevo proceso o artefacto hubiera sido definido en un WfMS C diferente a los ya incorporados, se haría necesario estudiar sus equivalencias par a par con cada uno, creciendo exponencialmente el tiempo y coste requeridos. De este modo, en cambio, el esfuerzo se mantiene constante, independientemente de la variedad de tipos de flujos.

La Figura 1 ilustra la propuesta haciendo uso de un lenguaje específico de dominio intermediario como elemento central.

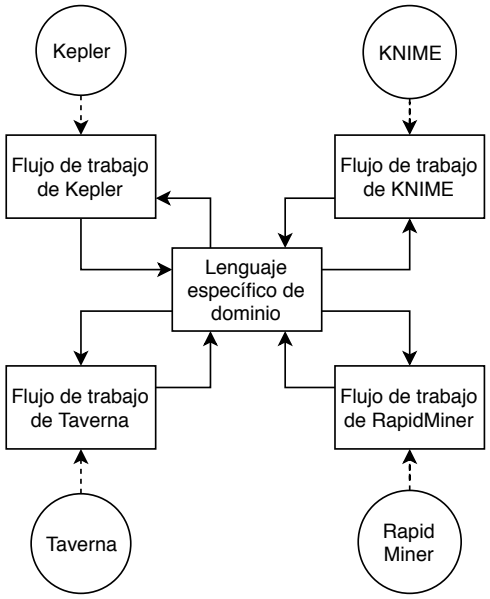

Figura 1. Propuesta para alcanzar la interoperabilidad entre distintos WfMS

Cada círculo representa un WfMS distinto, sobre el que se definen procedimientos propios para la extracción de conocimiento a partir de los datos generados en la organización o proveídos por terceros. Las flechas continuas indican la dirección de la equivalencia entre elementos de los flujos.

Como se indicara en la Sección I, la transformación de estos elementos no resulta trivial, y requiere una definición precisa de todos ellos, así como la declaración procedimental de las equivalencias en lenguajes de programación específicos para este fin, como ATL[1]. A modo ilustrativo, el código mostrado en el Listado 1 muestra un ejemplo de la transformación de un elemento de un flujo de trabajo de Taverna al lenguaje intermedio. En concreto, una conexión de datos, denominada en este sistema *DataLink*, es convertida a una conexión equivalente (elemento *DataLine*), conservando la referencia a los nodos interconectados.

Listado 1. Transformación de un elemento de Taverna al lenguaje intermedio

```
rule DataLinkToDataLine {
  from
    source : Taverna!DataLink
  to
    targetDataLine : DSL!DataLine
      (targetConnector <- LinkerIn,
      sourceConnector <- LinkerOut),
    LinkerOut : DSL!Linker
      (endpoint <- source.source),
    LinkerIn : DSL!Linker
      (endpoint <- source.sink) }
```

Finalmente, una vez se obtiene la especificación del procedimiento intensivo en datos en términos del lenguaje intermedio, éste se puede convertir indistintamente a cualquiera de los otros sistemas de flujos de trabajo, evitando la posible pérdida de *know-how*. Para ello, lenguajes como XSLT o herramientas como Acceleo[2] resultan de gran interés para la serialización.

[1]Más información en http://www.eclipse.org/atl/

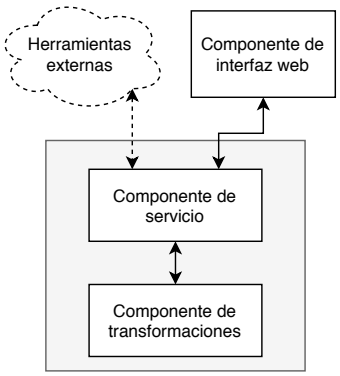

Figura 2. Componentes del prototipo

IV-B. Interoperabilidad como servicio: Prototipo

A modo de prueba de concepto, se ha desarrollado un prototipo que permite que las conversiones implementadas puedan ser fácilmente utilizadas por cualquier organización o integradas en un sistema externo mediante el uso de una API REST. Este prototipo se divide en tres componentes fundamentales (véase la Figura 2):

- El *componente de transformaciones* realiza las conversiones entre distintos flujos de trabajo (véase Sección IV-A).
- El *componente de servicio* ofrece la funcionalidad del componente de transformaciones a través de una API REST y coreografía todas las operaciones invocadas.
- El *componente de interfaz web* ofrece la página que da acceso al componente de servicio, facilitando la serialización de un flujo de trabajo origen en un formato destino requerido por el usuario.

Como se verá en la Sección V (Figura 4), la interacción con el prototipo se divide en dos zonas. A la izquierda se muestra un cuadro donde se introduce la serialización del flujo de trabajo origen, indicando a qué WfMS se corresponde. A la derecha se encuentra el selector de la herramienta destino. El listado de sistemas de gestión de flujos de trabajo es configurable y flexible, dependiendo de las conversiones que hayan sido definidas respecto al lenguaje específico de dominio intermedio. Al invocar el proceso de transformación, el componente de interfaz web se comunica con el de servicio, que informa a su vez al componente de transformaciones, encargado de devolver el resultado.

V. Caso de estudio: Interoperabilidad Taverna–Kepler

Como caso de estudio se ha desarrollado un proceso partiendo de un flujo de trabajo de Taverna. Luego, éste es transformado a un flujo equivalente y compatible con Kepler. Inicialmente se han contemplado los elementos involucrados en la estructuración y composición del proceso, si bien otros

[2]Más información en http://www.eclipse.org/acceleo

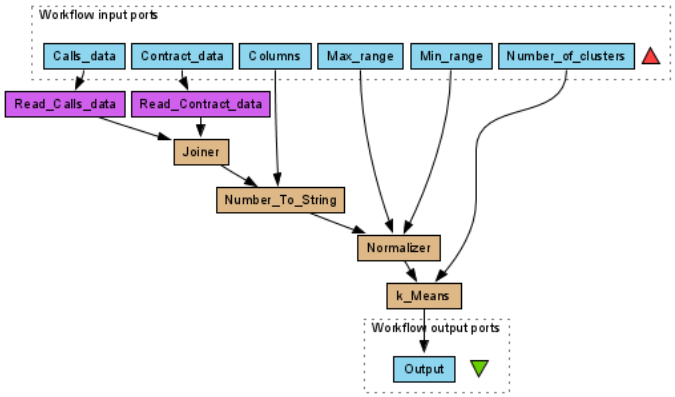

Figura 3. Flujo de trabajo origen en Taverna

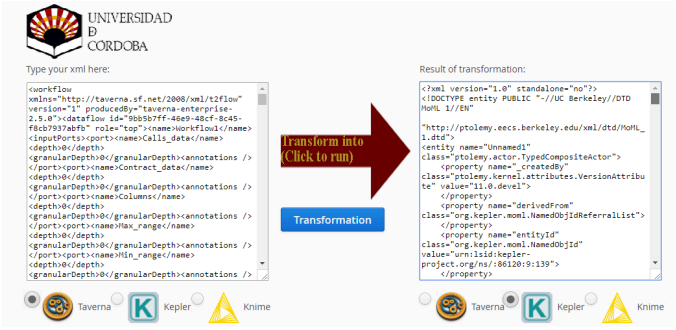

Figura 4. Uso del prototipo para transformar un flujo de trabajo

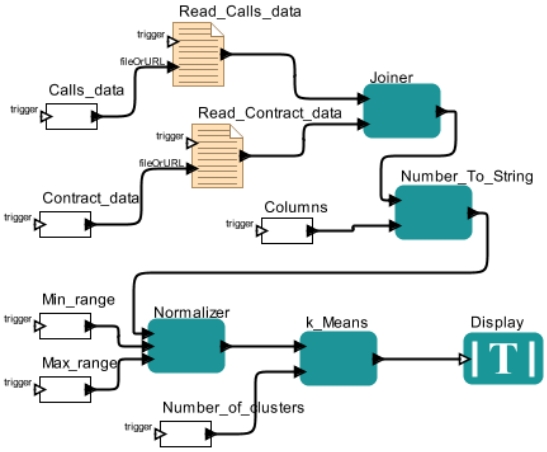

Figura 5. Flujo de trabajo destino en Kepler (generado automáticamente)

aspectos más avanzados relacionados con la ejecución se omiten hasta una fase posterior del desarrollo de esta propuesta.

Se analizará un proceso de segmentación de clientes de una empresa de telecomunicaciones como ejemplo de caso de ciencia de datos focalizado a Industria 4.0. La segmentación de clientes consiste en la agrupación de éstos en subgrupos, denominados *segmentos*, que presentan características comunes de consumo. Debido a estas similitudes, es más probable que los clientes de un mismo segmento respondan de modo parecido a estrategias de marketing para un producto. Igualmente, los clientes de distintos segmentos se diferencian de los del resto de subgrupos, por lo que requieren estrategias de marketing diferenciadas. Así pues, la segmentación permite determinar los rasgos básicos y generales que tendrá el potencial consumidor de un producto, aumentando la eficacia de las estrategias.

La Figura 3 representa el flujo de trabajo de Taverna que define el proceso que realiza el análisis de la información relacionada con los hábitos de consumo (p.ej. número de minutos hablados por día o número total de llamadas) y los contratos de los clientes con su compañía telefónica (p.ej. tipo de plan contratado o su código postal). Esta información es leída de dos ficheros, *Calls_data* y *Contract_data*, respectivamente. A continuación se lleva a cabo una secuencia de preprocesamiento consistente en operaciones de agregación, *Joiner*, conversión de tipos, *Number_To_String*, y normalización de datos, *Normalizer*. Finalmente, se aplica un algoritmo de agrupamiento, k-means, para crear los distintos segmentos que serán utilizados en labores de marketing.

Este flujo de trabajo de Taverna presenta una serie de elementos comunes, estudiados previamente en la Sección III, como son (a) los *tipos de datos básicos*, tanto cadenas como numéricos, (b) los nodos de entrada (*fuentes de datos*) y salida en memoria y de fichero, (c) los *nodos de ejecución* que permiten ejecutar fragmentos de código Java, (d) las *conexiones de datos* y, finalmente, (e) la *serialización* realizada en un único fichero accesible, en formato XML y sin compresión. El WfMS Kepler da soporte a estos mismos elementos, por lo que se pueden establecer correspondencias directas, a excepción de los nodos de ejecución, que son implementados y configurados de forma notoriamente distinta por ambas herramientas. Nótese que este tipo de incompatibilidades son comunes y son causa del aumento de la complejidad de la interoperabilidad.

Para llevar a cabo la conversión de activos entre ambos sistemas, primero debe identificarse el modo en que cada uno de los elementos del flujo de Taverna se serializa en XML. Una vez obtenidas todas las descripciones, se aplican las transformaciones (véase Sección IV-A) sobre cada elemento del flujo, obteniendo descripciones precisas compatibles con la formalización del lenguaje intermedio. Obsérvese que, en este punto, los activos definidos sobre el WfMS origen son independientes del sistema destino que los reincorporará, si bien en este estudio se ha trabajado sobre flujos de Kepler. La correspondiente serialización XML para Kepler se genera utilizando la definición formal obtenida para esta herramienta a partir de las conversiones realizadas desde el lenguaje intermedio. Todo este ciclo de conversión es coordinado y realizado de forma automática por el prototipo desarrollado (véase la Figura 4). Como resultado, la Figura 5 muestra el flujo de trabajo destino, que incluye la definición de todos los activos (procesos y artefactos) extraídos a partir de su descripción en Taverna.

VI. Discusión

La interoperabilidad alcanzada permite que el *know-how* organizacional contenido en un flujo de trabajo representado

para un WfMS particular sea reutilizable e independiente de cualquier otro sistema. Con ello, la propuesta establece un mecanismo de intercambio de activos entre herramientas distintas e incompatibles, siendo éste además de escalabilidad lineal con respecto a la incorporación de nuevos WfMS. El interés de la industria en conseguir activos interoperables se fundamenta en la colaboración entre distintos dominios de trabajo, para cada uno de los cuales predominan además herramientas de representación de procesos diferentes.

En este contexto, el esfuerzo requerido en la concepción y ejecución de tareas es determinante para el éxito de un proyecto. Por ello, el aprovechamiento del conocimiento, la reducción de la curva de aprendizaje de nuevas herramientas y acortar el tiempo requerido para la definición de nuevos procesos computacionales son factores que se ven favorecidos por la reutilización y adaptación de activos ya existentes. Por ejemplo, una parte de las tareas definidas en el proceso de segmentación de clientes estudiado (véase Sección V), como conversiones, normalización e incluso un algoritmo de agrupamiento, podría ser reubicadas de forma natural en un proceso aplicado a la detección de errores en datos procedentes de sensores dispuestos en maquinaria industrial.

No obstante, se han identificado una serie de dificultades relacionadas con la falta de formalización y estandarización. Por un lado, es necesario invertir gran cantidad de tiempo en la identificación y extracción de todas las características y elementos contemplados por cada WfMS, requiriendo para ello un número y variedad suficientes de flujos de trabajo. Nótese que la propuesta presentada requiere llevar a cabo dos conjuntos de conversiones por herramienta (WfMS al lenguaje intermedio, y viceversa), y no par a par entre herramientas. Sin embargo, la definición precisa y formal de los elementos de cada WfMS se hace imprescindible para garantizar la interoperabilidad semántica entre los activos. Por otro lado, debido a que cada herramienta proporciona sus propias características, o características comunes pero implementadas de forma muy distinta, no siempre existe una correspondencia directa entre elementos. En ocasiones, esto se resuelve definiendo equivalencias más complejas entre múltiples elementos de los flujos de origen y destino. Otras veces se trata de un problema de interoperabilidad técnica: por ejemplo, KNIME incrusta en su definición de flujos los archivos Java compilados, mientras que otras herramientas serializan la referencia a una dependencia externa. Para este tipo de interoperabilidades existen soluciones de código más frecuentes (p.ej. encapsulación, uso de servicios, etc.) frente a problemas relacionados con la semántica de las definiciones.

VII. Conclusiones

La dificultad en la definición de procesos computacionales y distintas fuentes de datos en la Industria 4.0 hace necesario el uso de mecanismos que favorezcan su representación y reutilización. Este trabajo presenta una propuesta de interoperabilidad entre flujos intensivos en datos de distintos WfMS mediante la correspondencia entre sus elementos.

Para ello, primero se han estudiado las características fundamentales de cada sistema por separado, definiendo de manera precisa el lenguaje de cada uno mediante un proceso de análisis y reingeniería. Después, tomando como base un lenguaje específico, independiente y bien definido para la ciencia de datos, se establecen correspondencias entre los elementos del lenguaje del WfMS y el lenguaje independiente, atendiendo a criterios de interoperabilidad semántica. De este modo, para cada nueva herramienta únicamente se requieren un número constante de equivalencias. Así pues, se ha comprobado para dos WfMS actuales, Taverna y Kepler, la posibilidad de migrar activos de un sistema a otro con el objetivo de reutilizarlos en situaciones diferentes. El proceso se ha automatizado mediante un prototipo web que ofrece su funcionalidad con API REST. En el futuro se completará la transformación entre los principales WfMS y se abordarán los retos discutidos.

Referencias

[1] L. Cao, "Data science: A comprehensive overview", ACM Computing Surveys, vol. 50, 2017.

[2] J. Dean, and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Communications of the ACM, vol. 51, pp. 107–113, 2008.

[3] I. Polato, R. Ré, A. Goldman, and F. Kon, "A comprehensive view of Hadoop research – A systematic literature review", Journal of Network and Computer Applications, vol. 46, pp. 1–25, 2014.

[4] Workflow Management Coalition, "Terminology & glossary", Technical Report WFMC-TC-1011, 1999.

[5] J. Yu, and R. Buyya, "A taxonomy of workflow management systems for grid computing", Journal of Grid Computing, vol. 3, no. 3, pp. 171 – 200, 2006.

[6] P. Korambath, J. Wang, A. Kumar, L. Hochstein, B. Schott, R. Graybill, M. Baldea, and J. Davis, "Deploying Kepler Workflows as Services on a Cloud Infrastructure for Smart Manufacturing", Procedia Computer Science, vol. 29, pp. 2254 – 2259, 2014.

[7] S.B. Japali, B. Archana, "Product recommendation for the day using fuzzy c-means and association rule generator in KNIME", Proceedings of the 2017 International Conference On Smart Technology for Smart Nation, pp. 556 – 559, 2018.

[8] I. Garcia-Magariño, G. Gray, R. Lacuesta, J. Lloret, "Survivability strategies for emerging wireless networks with data mining techniques: a case study with NetLogo and RapidMiner", IEEE Access, 2018.

[9] R. Salado-Cid, and J.R. Romero, "Enabling the definition and reuse of multi-domain workflow-based data analysis", Proceedings of the 16th International Conference on Intelligent Systems Design and Applications, pp. 687 – 696, 2016.

[10] P. Amstutz, M.R. Crusoe, N. Tijanić, "Common Workflow Language, v1.0 Specification", Common Workflow Language working group, 2016.

[11] D.C. Schmidt, "Guest Editor´s Introduction: Model-Driven Engineering", IEE Computer Society, pp. 25 – 31, 2006.

[12] R. Salado-Cid, J.R. Romero, "Lenguaje específico para el modelado de flujos de trabajo aplicados a ciencia de datos", Actas de XXI Jornadas en Ingeniería del Software y Bases de Datos, pp. 227 – 240, 2016.

[13] M.R. Berthold, N. Cebron, F. Dill, T.R. Gabriel, T. Kotter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "KNIME: The Konstanz Information Miner", Studies in Classification, Data Analysis, and Knowledge Organization, Springer, 2007.

[14] M. Hofmann, and R. Klinkenberg, "RapidMiner: Data mining use cases and business analytics applications", Chapman & Hall/CRC, 2013.

[15] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame and F. Bacall, "The Taverna workflow suite: Designing and executing workflows of web services on the desktop, web or in the cloud", Nucleic Acids Research, vol. 41, no. W1, pp. 557 – 561, 2013.

[16] I. Altintas, B. Ludäscher, S. Klasky, M.A. Vouk, "Introduction to scientific workflow management and the Kepler system", Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, 2006.