

Adaptación de Workflows basada en Ontologías

Álvaro E. Prieto ¹, Adolfo Lozano-Tello ¹

aeprieto@unex.es, alozano@unex.es

¹ Escuela Politécnica, Universidad de Extremadura, Avda. de la Universidad s/n, 10003, Cáceres, España

DOI: 10.17013/risti.14.67-82

Resumen: Los workflows para procesos administrativos son utilizados en empresas e instituciones públicas pero, para poder utilizarlos adecuadamente en sus distintas áreas y departamentos, deben ser adaptados a las características propias de cada uno de ellos, respetando las normas que regulan el proceso a nivel general. Este problema, llamado Problema de la Adaptación Jerárquica, también implica establecer las medidas que se deben tomar cuando la normativa general cambia, para mantener la consistencia entre los distintos niveles mediante la propagación de los cambios a todas las adaptaciones. Para resolver este problema, en este trabajo se presenta el Método de Adaptación Jerárquica. Un método basado en ontologías que define las reglas que debe satisfacer un workflow genérico para ser considerado adaptable a diferentes casos de aplicación y las reglas que deben satisfacer las adaptaciones. Además, proporciona las operaciones que facilitan tanto la adaptación de los workflows administrativos como la propagación de los cambios.

Palabras-clave: adaptación jerárquica; workflows; ontologías.

Workflow Adaptation based on Ontologies

Abstract: Administrative workflows are used in enterprises and public institutions but, in order to use them adequately in their different areas and departments, they must be adapted to the particular conditions of each one, complying with the general regulations of the process established at the top level. This problem, called Hierarchical Adaptation Problem, also implies establishing the proper measures to accomplish when the general regulation is changed. Such measures must maintain the consistency among the different levels by means of the propagation of the changes to all the adaptations. To solve this problem, this work presents the Hierarchical Adaptation Method. A method based on ontologies that defines the rules that must satisfy a generic workflow to be considered adaptable to different application cases and the rules that must satisfy the adaptations. Moreover, it provides the operations that facilitate both adaptation of administrative workflows and propagation of changes.

Keywords: hierarchical adaptation; workflows; ontologies.

1. Introducción

En instituciones públicas y grandes empresas se utiliza un tipo de proceso de negocio conocido como proceso administrativo. Este tipo de proceso caracteriza por estar regulados por leyes o normativas que definen claramente las actividades que componen el proceso, quién debe realizarlas, cómo deben ser realizadas, cuándo y en qué plazos (Feldman & Khademian, 2000), de manera que tienen una estructura clara y bien definida, donde se conoce con antelación cada posible ruta que puede tomarse a través del proceso (Moore, Stader, Macintosh, Casson-du Mont, & Chung, 1999). Ejemplos de este tipo de procesos serían la solicitud de algún tipo de permiso o material, la gestión de incidencias, la aprobación de presupuestos de gastos, la petición de préstamos bancarios o cualquier proceso administrativo que implique una gestión de expedientes en organizaciones.

Para la gestión automatizada de los procesos administrativos pueden utilizarse workflows (Hollingsworth, 1995). Los workflows para estos procesos suelen proporcionar un conjunto de formularios para ser rellenados y enviados a través de una serie de fases que permiten un encaminamiento sencillo de la información manejada, siguiendo un conjunto de reglas conocidas por todos los participantes involucrados y que normalmente deben terminar con la aprobación por parte de un usuario de la petición o solicitud que inició el proceso. Además, estos workflows no suelen requerir accesos a otros sistemas de información, ni realizar cálculos complejos, pero sí deben disponer de mecanismos que permitan coordinar a los usuarios responsables de cada etapa del proceso y avisarles cuándo deben realizar una tarea (McReady, 1992) (Georgakopoulos, Hornick, & Sheth, 1995) (Alonso, Agrawal, Abbadí, & Mohan, 1997).

Normalmente, estos workflows no tienen que gestionar un elevado número de actividades, los usuarios que intervienen son reducidos y no se manejan grandes cantidades de datos. En cambio, la dificultad en la gestión se produce cuando deben llevarse a cabo en diferentes instituciones con posibles variaciones, situación que se denomina Problema de la Adaptación Jerárquica.

Este problema se produce cuando el workflow de un proceso administrativo debe adaptarse a las características propias de los distintos niveles de las instituciones donde será utilizado, sin que las variaciones que deban realizarse en cada caso particular afecten a las restricciones fijadas para el workflow original. Además, este problema también incluye cómo propagar los cambios que puedan ocurrir en el workflow original a todas sus adaptaciones.

Este problema puede afectar tanto a instituciones públicas como a empresas privadas. Un ejemplo de este problema en instituciones públicas sucede dentro de la Unión Europea cuando regula un proceso administrativo, ya que, antes de ser aplicado en sus estados miembros, deberá ser adaptado a las características particulares de cada uno de ellos, respetando las restricciones fijadas originalmente. Además, este problema puede ser aún más complejo en países como España, dividido en regiones con bastante autonomía legislativa, debido a que existe una ley de procedimiento administrativo¹

¹ Ley 30/1992, de 26 de noviembre, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común.

que establece el esquema «bases más desarrollo» que permite a estas regiones dictar sus propias normas siempre que se ajusten a las bases estatales. Esto también va a implicar que, si la Unión Europea modifica la normativa que regula el proceso, los cambios no solo deben ser propagados a la adaptación realizada en España a nivel estatal sino también a la adaptación realizada a nivel regional, tal y como se muestra en la Figura 1.

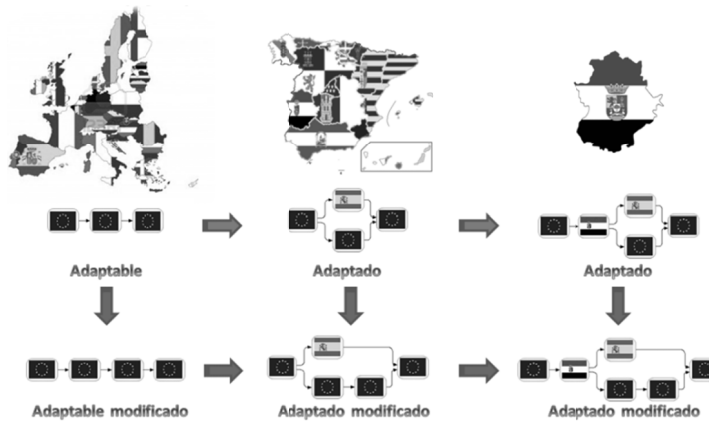


Figura 1 – El Problema de la Adaptación Jerárquica en España.

Las instituciones privadas también se pueden ver afectadas por este problema. Por ejemplo, en muchos países existe un banco central que obliga a las entidades financieras que quieran operar en su jurisdicción a seguir unas determinadas normas en procesos, como la concesión de préstamos a clientes. Dichas normas, que podrían incluir desde los tipos de préstamos que pueden ofrecer o el rango de interés en que se pueden conceder, hasta los plazos de devolución o las actividades que deben realizarse antes de aprobarlo o denegarlo, deben ser adaptadas a la idiosincrasia de cada banco antes de ser aplicadas, pero siempre respetando las restricciones que haya fijado el banco central.

En el Problema de la Adaptación Jerárquica pueden distinguirse 4 etapas interrelacionadas. Estas cuatro etapas, que se muestran en la Figura 2, son:

1. Especificación del workflow genérico. En esta etapa, los ingenieros deben especificar el workflow genérico a partir de la normativa general que regula el proceso. Deberá hacerse en un lenguaje que permita especificar las actividades, su orden y sus plazos, los responsables de realizarlas y los datos necesarios. Además, el lenguaje de especificación utilizado deberá permitir el establecimiento de restricciones para la adaptación sobre los elementos anteriores para que sea posible abordar el resto de etapas del problema.
2. Especificación de restricciones de adaptación. En esta etapa, los ingenieros, a partir de la normativa general del proceso, deberán establecer de forma coherente las restricciones de adaptación que permitan indicar si los distintos elementos del workflow son obligatorios o necesarios en cualquier posible adaptación.

3. Adaptación. En esta etapa, a partir de la especificación del workflow adaptable y, teniendo en cuenta las características de cada caso particular, se deberá especificar el workflow adaptado que siempre debe pero respetando las restricciones fijadas.

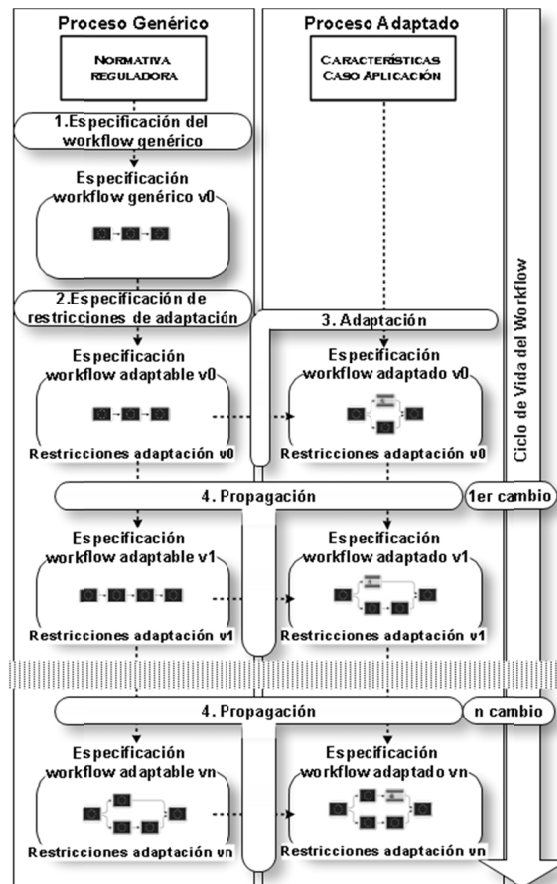


Figura 2 – Etapas del problema de la Adaptación Jerárquica.

4. Propagación. Esta etapa se produce si la normativa que regula el proceso genérico cambia. Esto provoca que sea necesario modificar el workflow adaptable y propagar esos cambios a todos los workflows adaptados a partir de él.

A la vista de estas cuatro etapas, cualquier propuesta que pretenda solucionar el Problema de la Adaptación Jerárquica debería proporcionar los mecanismos necesarios para afrontar todas ellas en su conjunto. Pero, actualmente, los ingenieros responsables de la especificación e implantación de estos workflows no cuentan con ningún método que les permita afrontar este problema con garantías de hacerlo de forma correcta.

En este trabajo se presenta una propuesta para afrontar este problema denominada Método de Adaptación Jerárquica. Este Método se fundamenta en la especificación de los workflows en forma de ontologías. El Método propone una forma de especificar workflows para procesos administrativos y sus restricciones de adaptación junto con las operaciones necesarias para adaptarlos a las características particulares de los entornos donde deban ser aplicados, respetando las restricciones fijadas en el workflow genérico. Además, también proporciona las operaciones necesarias para propagar los cambios que puedan sufrir los workflows genéricos a los workflows adaptados.

Así, en la sección 2 se analiza si las propuestas más conocidas en el campo de especialización y herencia de workflows son aplicables a este problema, en la sección 3 se presenta el Método de Adaptación Jerárquica y, por último, en la sección 4, se presentan los resultados obtenidos en la validación del método por parte de un grupo de Ingenieros de Software.

2. Trabajos Relacionados

Existen propuestas en el campo de herencia y especialización de workflows y procesos que, por sus características, parece el campo de investigación donde los ingenieros pudieran encontrar una forma de afrontar el Problema de la Adaptación Jerárquica.

Una de las primeras propuestas en este campo es la de Wyner y Lee (Wyner & Lee, 1995) en la que defienden que un proceso general sería un tipo de proceso abstracto que contiene todas las posibles variantes, y que las especializaciones realmente restringen el modelo general.

Esta propuesta, aunque interesante a nivel lógico, no es útil si se traslada al Problema de la Adaptación Jerárquica de forma ortodoxa. Esto es debido a que aplicando esta perspectiva de forma estricta, el workflow original debería contener todas las actividades, datos manejados y participantes involucrados de todas las posibles adaptaciones, lo que obligaría a conocer de antemano todos los posibles casos particulares donde va a utilizarse.

La propuesta de herencia de workflows de van der Aalst (Aalst & Basten, 2002), posiblemente la más conocida en este campo, tiene como principal objetivo migrar instancias en ejecución de un workflow modificado. Dado que esta propuesta de especialización sirve para un problema centrado en el nivel de ejecución de los workflows y que el Problema de Adaptación Jerárquica está enfocado en el nivel de definición, no es posible aplicar esta propuesta en este problema.

En cambio, existe una extensión de esta propuesta realizada por Wyner y Lee (Wyner & Lee, 2005) que sí es interesante para resolver el Problema de la Adaptación Jerárquica. En concreto, proponen la “congelación” de elementos limitando las posibilidades a la hora de realizar nuevas adiciones al original. Como se verá en la siguiente sección, el Método de Adaptación Jerárquica propuesto extiende y amplía esta idea de manera que, en el diseño del workflow original, puedan fijarse distintos niveles de restricciones a la adaptación. El problema es que para llevar esto a cabo, tal como los propios Wyner y Lee explican al realizar su propuesta, los lenguajes de representación tradicionales de workflow como WF-net restringen mucho las posibilidades de aplicación de estas ideas.

Este es uno de los motivos principales por los que el Método de Adaptación Jerárquica se basa en la especificación de los workflows en forma de ontologías.

Hay también una propuesta de especialización de procesos en representación del conocimiento basado en herencia no monótona (Bernstein & Grosz, 2003) (Ferndriger et al., 2008). Este trabajo propone una noción de especialización que permite tanto añadir como borrar, lo que implica que la herencia no será monótona ya que afecta a los conceptos heredados. Como se verá en la siguiente sección, en el Método de Adaptación Jerárquica también se va a aplicar la idea de herencia no monótona aunque dicha herencia no monótona va a estar limitada en función de las restricciones de adaptación que se fijen en cada workflow.

La última propuesta analizada define la especialización en función de las actividades y el orden parcial que forman (Choppy, Desel, & Petrucci, 2011). Según esta propuesta la especialización puede producirse si se añade o elimina información, actividades o características al especializado siempre que se respete el orden parcial de ejecución de actividades en las especializaciones. Como se verá también en la siguiente sección, esta idea también se aplica en el Método de Adaptación Jerárquica pero a nivel de definición y no de ejecución.

Para concluir esta sección, se puede decir que ninguna de las propuestas más conocidas en el campo de especialización y herencia de workflows y procesos es aplicable directamente a solucionar el Problema de la Adaptación Jerárquica. Entre las razones se pueden citar, en primer lugar, que todas las propuestas están demasiado orientadas al comportamiento y, por tanto, a las actividades de los procesos sin considerar qué ocurre con los datos y participantes involucrados. En segundo lugar, porque la mayoría de ellas restringen la especialización a una serie de normas predefinidas que no permiten al diseñador del proceso flexibilizar la especialización en función de la naturaleza particular del proceso gestionado por el workflow. Y, en último lugar pero no menos importante, ninguna de estas propuestas aborda el problema de la propagación de cambios desde los workflows de los procesos genéricos a los workflows de los procesos adaptados.

El Método de Adaptación Jerárquica que se presenta a continuación trata de eliminar estas carencias con el objetivo de proporcionar una solución completa al Problema de la Adaptación Jerárquica.

3. El Método de Adaptación Jerárquica

El Método de Adaptación Jerárquica está fundamentado sobre la especificación de workflows en forma de ontologías. Son varios los motivos para esta elección. En primer lugar, la precisión y completitud de las ontologías para la representación de los elementos de los workflows y su versatilidad para representar no solo las actividades, los participantes y los datos involucrados en el workflow de un proceso, sino también para poder establecer las características particulares de adaptación de cada uno de estos elementos en cada workflow. En segundo lugar, la posibilidad de dividir la especificación de un workflow en dos ontologías, una para los datos y participantes del dominio y, otra para describir las propiedades del proceso que gestiona el workflow, y las actividades que lo componen, van a facilitar tanto la reutilización de los datos y

participantes entre los workflows de un mismo dominio como la adaptación de los workflows a los casos particulares. En tercer lugar, el uso de ontologías va a permitir establecer una noción de adaptación más abierta, ya que es el diseñador del workflow del proceso genérico quién establece las restricciones de adaptación concretas de cada workflow. Por último, la especificación de workflows en forma de ontologías, donde cada elemento tiene un propósito perfectamente definido, va a permitir desarrollar el conjunto de operaciones de propagación que se necesitan para que cualquier posible cambio que pueda necesitar el workflow del proceso genérico pueda ser transmitido a todos los adaptados a partir de él.

Sobre esta base, el Método de Adaptación Jerárquica proporciona el conjunto de métodos y operaciones necesarios para afrontar cada una de las cuatro etapas que componen el Problema de la Adaptación Jerárquica y que son resumidas a continuación.

3.1. Primera etapa: especificación del workflow genérico.

Para la primera etapa, en el Método de Adaptación Jerárquica se ha actualizado la propuesta de representación de workflows de procesos administrativos en forma de ontologías que fue propuesta en (Á. E. Prieto & Lozano-Tello, 2009) y reestructurada en (A. E. Prieto & Lozano-Tello, 2012).

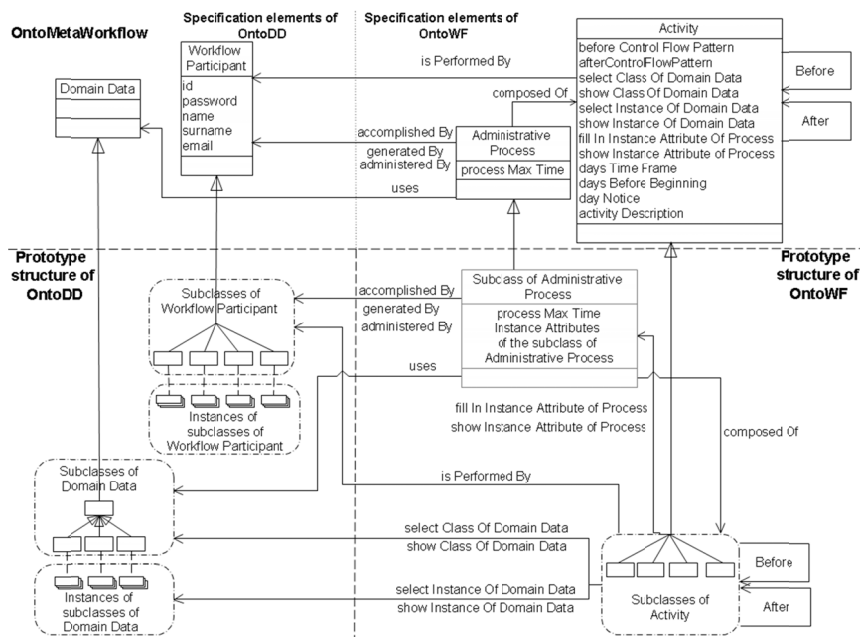


Figura 3 – Elementos de especificación de *OntoMetaWorkflow* junto con el uso de alguno de esos elementos en las ontologías *OntoDD* y *OntoWF*.

Así, la especificación del workflow genérico se va a realizar a partir de la ontología

*OntoMetaWorkflow*². Esta ontología establece el marco base de reglas de representación definiendo los elementos comunes a los workflows para procesos administrativos. A partir de *OntoMetaWorkflow*, un ingeniero de ontologías debe construir o reutilizar, en primer lugar, la ontología *OntoDD*. Esta ontología contendrá los datos relevantes de un dominio concreto y los usuarios que pueden participar en los posibles workflows que se especifiquen en ese dominio. En segundo lugar, se especificará el workflow con la lógica del proceso administrativo dentro de una ontología denominada *OntoWF*. *OntoWF* es una ontología que contendrá las propiedades concretas de un proceso administrativo junto con sus actividades, el orden entre ellas, qué tipo de usuario de los especificados en la ontología *OntoDD* puede realizar las actividades y qué datos especificados en la ontología *OntoDD* serán utilizados por cada actividad.

Para hacer frente al Problema de la Adaptación Jerárquica, se han añadido a *OntoMetaWorkflow* un conjunto de elementos que permiten indicar las restricciones de adaptación a los previamente existentes de especificación (o definición) y ejecución. Estos nuevos elementos de adaptación se utilizan a partir de la segunda etapa y se detallan en la siguiente subsección. Para esta primera etapa se hace uso de los elementos de especificación. Estos elementos permiten especificar las actividades que componen el workflow del proceso, los responsables de realizarlas y los datos necesarios para llevarlas a cabo. En la Figura 3 se muestran estos elementos junto con la forma en que son utilizados en las ontologías *OntoDD* y *OntoWF*.

Además, para afrontar el resto de etapas del Problema de la adaptación Jerárquica, se han definido 24 operaciones básicas³ de modificación que pueden aplicarse sobre un workflow especificado usando *OntoMetaWorkflow* de manera que tras el cambio siga siendo un workflow correcto.

3.2. Segunda etapa: especificación de restricciones de adaptación

En esta etapa, el Método de Adaptación Jerárquica utiliza los elementos de adaptación de *OntoMetaWorkflow* para establecer qué características son esenciales o no en un workflow. Estos elementos van a permitir indicar las características de adaptación del workflow de un proceso genérico. Estas características restringirán las posibles adaptaciones que de él se hagan a casos concretos de aplicación de manera que pueda asegurarse que los workflows adaptados sean adaptaciones jerárquicas válidas del workflow del dominio genérico.

Se han definido tres tipos de elementos de adaptación en *OntoMetaWorkflow*: elementos para indicar obligatoriedad (elementos *Mandatory* en *OntoMetaWorkflow*), elementos para indicar inflexibilidad (elementos *Rigid* en *OntoMetaWorkflow*) y elementos para indicar requerimiento (elementos *Required* en *OntoMetaWorkflow*).

² La nueva versión de *OntoMetaWorkflow* y las herramientas mencionadas en (A. E. Prieto & Lozano-Tello, 2012) están disponibles en <http://uex.be/weapon>

³ Los detalles de estas operaciones están disponibles, entre las páginas 64 y 78, de esta Tesis Doctoral: <http://uex.be/tdaeprieto>

En la Figura 4 se muestran estos elementos junto con la forma en que son utilizados en las ontologías *OntoDD* y *OntoWF*.

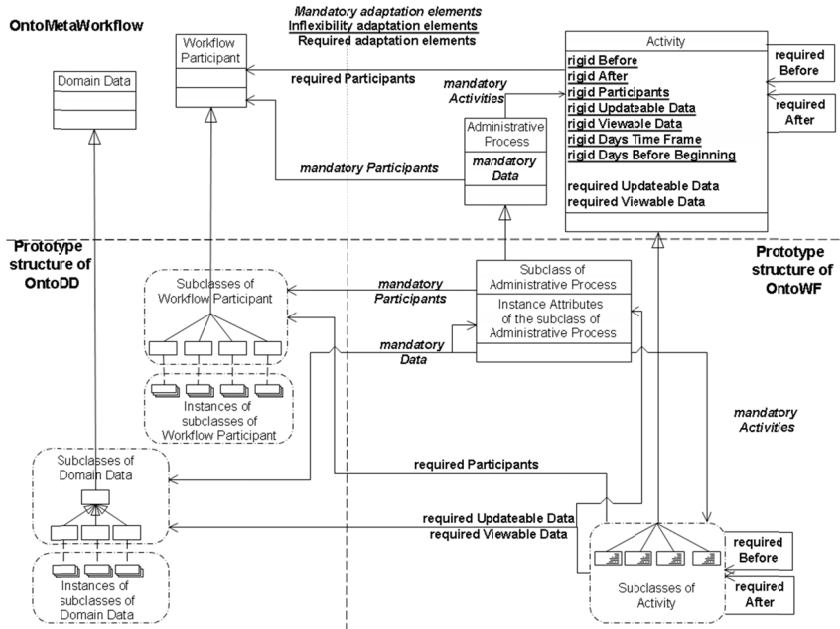


Figura 4 – Elementos de adaptación de *OntoMetaWorkflow* junto con el uso de alguno de esos elementos en las ontologías *OntoDD* y *OntoWF*.

Los primeros (*Mandatory*) se definen sobre el proceso administrativo especificado en *OntoWF* y sirven para indicar las actividades (*Mandatory Activities*), tipos de participantes (*Mandatory Participants*), datos del dominio y propiedades del proceso (*Mandatory Data*) que siempre deben aparecer en cualquier posible adaptación de un workflow.

El segundo tipo de elementos (*Rigid*) se define sobre las actividades y sirven para indicar que no se permite ningún cambio en las actividades inmediatamente anteriores e inmediatamente posteriores a la actividad en cuestión (*Rigid Before* and *Rigid After*), en los tipos de participante que pueden realizarla (*Rigid Participants*), en los datos del dominio y propiedades del proceso que utiliza (*Rigid Updateable Data* and *Rigid Viewable Data*) y en los plazos de tiempo para comenzar y finalizar la actividad (*Rigid Days Time Frame* and *Rigid Days Before Beginning*). De este modo, si en el workflow adaptado alguna actividad no puede satisfacer estas restricciones, es obligatorio eliminar dicha actividad del workflow adaptado.

El tercer tipo de elementos (*Required*) se define sobre las actividades pero en este caso se utilizan para indicar las restricciones mínimas en cuanto a qué actividades deben realizarse antes y después de una actividad (*Required Before* and *Required After*), qué tipos de participante deben estar al menos disponibles para realizarla (*Required Participants*) y qué datos del dominio y propiedades del proceso deben ser utilizados como mínimo en la actividad (*Required Updateable Data* and *Required Viewable*

Data) pero permitiendo que, en todos los casos, puedan añadirse nuevos elementos de alguno de estos tipos. Al igual que con el segundo tipo de elementos, si alguna actividad no es capaz de satisfacer los requisitos fijados, será obligatorio eliminar dicha actividad del adaptado.

A partir de estos elementos de adaptación, en esta etapa del Método de Adaptación Jerárquica se establece que un workflow adaptable será aquel workflow correctamente especificado usando *OntoMetaWorkflow* y que toma valores en algunos de los elementos de adaptación cumpliendo siempre la siguiente restricción: el workflow compuesto exclusivamente de las actividades, datos del dominio, propiedades del proceso y participantes del workflow afectados por los elementos de adaptación del tipo obligatorio debe especificar por sí solo un workflow correctamente especificado usando *OntoMetaWorkflow*. Esta definición implica que para que un workflow sea adaptable no es suficiente con que tome cualquier valor en alguno de los elementos de adaptación sino que es necesario que esos valores estén fijados de forma coherente. Para facilitar la especificación correcta de workflows adaptables, el Método de Adaptación Jerárquica proporciona el método para especificar un workflow adaptable que ayuda a fijar los valores en estos elementos de adaptación en un workflow correctamente especificado usando *OntoMetaWorkflow* y que comprende los siguientes pasos:

1. Indicar qué actividades son obligatorias en cualquier adaptación jerárquica del workflow incluyendo dichas actividades en la relación *Mandatory Activities*. Al menos deben serlo la actividad inicial y una actividad final.
2. A continuación hay que fijar los requisitos de adaptabilidad en lo que a la ubicación de las actividades se refiere. Para ello hay que realizar los siguientes pasos:
 1. Para cada actividad en la ontología *OntoWF*, empezando por la actividad inicial, siguiendo todos los posibles caminos y el orden en que están situadas las actividades y terminando en las actividades finales, indicar si las actividades inmediatamente posteriores (las incluidas en la relación *After*) de la actividad tratada siempre deben ser las mismas usando el atributo *Rigid After*. Si la actividad está incluida en la relación *Mandatory Activities* del workflow, solo debe indicarse este atributo a verdadero si todas las actividades afectadas también lo están.
 2. Para cada actividad en la ontología *OntoWF*, empezando por la actividades finales, siguiendo todos los posibles caminos y el orden inverso en que están situadas las actividades y terminando en la actividad inicial, indicar si las actividades inmediatamente anteriores (las incluidas en la relación *Before*) de la actividad tratada siempre deben ser las mismas usando el atributo *Rigid Before*. Hay que tener en cuenta que si hay algún *Rigid After* situado en una actividad que es inicio de bifurcación entonces la actividad que es final de bifurcación también debe tener a verdadero el atributo *Rigid Before*. Además, si la actividad está incluida en la relación *Mandatory Activities* del workflow, solo debe indicarse este atributo a verdadero si todas las actividades afectadas también lo están.
 3. Para cada actividad en la ontología *OntoWF*, empezando por la actividad inicial, siguiendo todos los posibles caminos y el orden en que están situadas las actividades y terminando en las actividades inmediatamente anteriores a las actividades finales, incluir en la relación *Required After* de

- cada actividad aquellas actividades posteriores que, como mínimo, deben aparecer con posterioridad a la actividad en el workflow. Si la actividad está incluida en la relación *Mandatory Activities* del workflow, solo deben incluirse en esta relación a actividades que también lo estén.
4. Para cada actividad en la ontología *OntoWF*, empezando por la actividades finales, siguiendo todos los posibles caminos y el orden inverso en que están situadas las actividades y terminando en las actividades inmediatamente posteriores a la actividad inicial, incluir en la relación *Required Before* de cada actividad aquellas actividades anteriores que, como mínimo, deben aparecer con anterioridad a la actividad en el workflow. Si la actividad está incluida en la relación *Mandatory Activities* del workflow, solo deben incluirse en esta relación a actividades que también lo estén.
3. Por último, es necesario fijar los requisitos de adaptabilidad en cuanto a las características de cada actividad se refiere. Para ello hay que realizar para cada actividad en la ontología *OntoWF*, empezando por la actividad inicial y terminando en las actividades finales y siguiendo todos los posibles caminos y el orden en que están situadas las actividades, las siguientes acciones:
1. Indicar si los participantes disponibles para realizar la actividad siempre deben ser los indicados en la relación *Is Performed By* fijando el atributo *Rigid Participants* a verdadero. Si la actividad está incluida en la relación *Mandatory Activities* del workflow entonces debe incluirse a todos los participantes afectados en la relación *Mandatory Participants* si no lo estaban previamente.
 2. Dado que una de las características de un workflow correctamente especificado usando *OntoMetaWorkflow* es que todas las actividades deben incluir, al menos, un *Workflow Participant* en la relación *Is Performed By*, es necesario incluir alguno de los participantes indicados en la relación *Is Performed By* en la relación *Required Participants*. Además, si la actividad está incluida en la relación *Mandatory Activities* del workflow entonces debe incluirse a todos los participantes afectados en la relación *Mandatory Participants* si no lo estaban previamente.
 3. Indicar si los datos del dominio seleccionables y las propiedades del proceso modificables siempre deben ser los mismos usando el atributo *Rigid Updateable Data*. Si la actividad está incluida en la relación *Mandatory Activities* del workflow entonces debe incluirse a todos los datos y propiedades afectados en el atributo *Mandatory Data* si no lo estaban previamente.
 4. Dado que una de las características de un workflow correctamente especificado usando *OntoMetaWorkflow* es que todas las actividades deben incluir, al menos, una operación de selección sobre datos del dominio o una operación de modificación sobre propiedades del proceso, es necesario incluir alguno de los datos del dominio indicados en los atributos del dominio *Select Class Of Domain Data* o *Select Instance Of Domain Data* o a alguna de las propiedades del proceso indicadas en el atributo *Fill In Instance Attributes of Process* en el atributo *Required Updateable Data*. Además, si la actividad está incluida en la relación

Mandatory Activities del workflow entonces debe incluirse a todos los datos y propiedades afectados en el atributo *Mandatory Data* si no lo estaban previamente.

5. Indicar si los datos del dominio y las propiedades del proceso visualizables siempre deben ser los mismos usando el atributo *Rigid Viewable Data*. Este atributo debe tratarse con precaución porque podría darse el caso de que alguno de los elementos que se debe visualizar no esté en el workflow adaptado porque la actividad donde se seleccionaba o modificaba no está disponible en el workflow adaptado. Por este motivo, este atributo solo debe fijarse a verdadero si todos los datos que visualiza la actividad están incluidos en el atributo *Mandatory Data* y, además, están incluidos en el atributo *Required Updateable Data* de actividades obligatorias anteriores o son manipulados en actividades obligatorias anteriores que tengan fijado a verdadero el atributo *Rigid Updateable Data*.
6. Si alguno de los datos del dominio o de las propiedades del proceso siempre debe poder visualizarse en la actividad, incluir en el atributo *Required Viewable Data* dichos datos del dominio o propiedades del proceso. Al igual que en el paso anterior, este atributo también debe tratarse con precaución porque podría darse el caso de que alguno de los elementos que se debe visualizar no esté en el workflow adaptado porque la actividad donde se seleccionaba o modificaba no está disponible en el workflow adaptado. Por este motivo, en este atributo solo deben incluirse aquellos datos o propiedades que estén incluidos en el atributo *Mandatory Data* y, además, están incluidos en el atributo *Required Updateable Data* de actividades obligatorias anteriores o son manipulados en actividades obligatorias anteriores que tengan fijado a verdadero el atributo *Rigid Updateable Data*.
7. Indicar si el número máximo de días para realizar la actividad es fijo o puede reducirse usando el atributo *Rigid Days Time Frame*.
8. Indicar si es el número mínimo de días antes de comenzar la actividad es fijo o puede ampliarse usando el atributo *Rigid Days Before Beginning*.

Además, hay que indicar que, aunque los elementos de adaptación se especifican en la ontología *OntoWF*, también afectan a la ontología *OntoDD* de dos formas. Por un lado es obligatorio que los participantes afectados por la relación *Mandatory Participants* deban aparecer en la ontología *OntoDD* que use el workflow adaptado. Por otro lado, también es obligatorio que los *Domain Data* afectados por el atributo *Mandatory Data* también deban aparecer en la ontología *OntoDD*. Del mismo modo, las propiedades que el proceso adaptado tenga especificadas en la ontología *OntoWF* al menos deben ser las que están afectadas en el adaptable por el atributo *Mandatory Data*.

3.3. Tercera etapa: adaptación

Para esta etapa, el Método de Adaptación Jerárquica define un workflow adaptado como aquel correctamente especificado usando *OntoMetaWorkflow* que cumple una

serie de restricciones, implícitas y explícitas⁴, para la adaptación con respecto a un workflow adaptable dado.

Por un lado, las restricciones implícitas son aquellas que no dependen de los elementos de adaptación sino que dependen de la propia especificación del workflow en las ontologías *OntoDD* y *OntoWF*. Por otro lado, las restricciones explícitas son aquellas que están relacionadas con los valores que tenga fijado el workflow en los elementos de adaptación.

En resumen, un workflow será un workflow adaptado a partir de un workflow adaptable si:

- Es un workflow correctamente especificado usando *OntoMetaWorkflow*.
- Todos los datos, participantes y actividades obligatorias del workflow adaptable están incluidas en el workflow adaptado.
- Todas las actividades del workflow adaptable, obligatorias o no, incluidas en el workflow adaptado, cumplen las restricciones de inflexibilidad y requerimiento establecidas por sus elementos de adaptación.

Para especificar workflows adaptados correctos a partir de un workflow adaptable, el Método proporciona 20 operaciones para la adaptación jerárquica⁵ que pueden aplicarse sobre el workflow adaptable hasta conseguir el workflow adaptado deseado. Todas estas operaciones de adaptación jerárquica disponibles hacen uso de las operaciones de modificación de un workflow definidas para la etapa 1 pero añadiendo una serie de restricciones previas que vienen marcadas por los valores de los elementos de adaptación contenidos en el workflow adaptable y que deben cumplirse antes de poder aplicar cada operación.

Es decir, si por ejemplo un banco quiere adaptar el workflow genérico de petición de préstamos, basado en las normas del Banco Central a las características particulares de su entidad, las operaciones de esta etapa son las que puede aplicar sobre el workflow genérico para que el workflow adaptado pueda ser considerado correctamente adaptado con respecto al del Banco Central.

3.4. Cuarta etapa: propagación

Para esta última etapa, el Método de Adaptación Jerárquica proporciona 60 operaciones de propagación jerárquica⁶ de los cambios sufridos por un workflow adaptable a sus workflows adaptados. Estas operaciones habrá que aplicarlas si el proceso administrativo especificado en un workflow adaptable sufre uno o varios cambios. Esto va a implicar que, en primer lugar, el workflow adaptable debe ser modificado para contener los nuevos cambios y, en segundo lugar, que estos cambios deben propagarse en cascada a todos los workflow adaptados a partir del workflow adaptable.

⁴ El detalle de todas estas restricciones está disponible, entre las páginas 89 y 91, de esta Tesis Doctoral: <http://uex.be/tdaeprieto>

⁵ Los detalles de estas operaciones están disponibles, entre la página 94 y 106, de esta Tesis Doctoral: <http://uex.be/tdaeprieto>

⁶ Los detalles de estas operaciones están disponibles, entre la página 109 y 169, de esta Tesis Doctoral: <http://uex.be/tdaeprieto>

Estas operaciones han sido definidas para que, por un lado, el workflow adaptable siga manteniendo dicha condición tras su aplicación y además los workflows adaptados mantengan su condición tras su aplicación. Siguiendo con el ejemplo del banco de la sección anterior, las operaciones que aquí se presentan aplicadas en este ejemplo lograrían dos objetivos en el caso de que el Banco Central decidiera cambiar la normativa del proceso de petición de préstamos y, por tanto, cambiar el workflow genérico que lo representa. El primero de ellos es que el workflow genérico tras los cambios siga estando correctamente especificado y siga siendo un workflow adaptable. El segundo es que los bancos que adaptaron dicho workflow a su caso particular, puedan propagar los cambios producidos en el genérico de manera que su workflow adaptado siga manteniendo esta condición con respecto al workflow genérico de Banco Central.

Esto va a suponer que, si las acciones de cada operación jerárquica de propagación de cambios son aplicadas correctamente, tanto en el workflow adaptable como en los workflows adaptados, no será necesario realizar ninguna verificación posterior para asegurar que estos dos objetivos son cumplidos.

4. Validación de la Propuesta

El método ha sido validado en dos fases. En la primera fase fue validado por un grupo de once Ingenieros de Software y, una vez incluidas algunas de sus recomendaciones, fue validado en una segunda fase por un grupo de treinta Ingenieros de Software. En concreto, a ambos grupos se les pidió que tomaran el rol de un Ingeniero que acababa de ser contratado como responsable de los workflows para procesos administrativos en la zona de Extremadura por un banco con distintos departamentos en su sede central de Madrid y con una red de oficinas dividida en zonas geográficas por todo el territorio nacional⁷. Tras completar el caso, cada uno de ellos respondió a un cuestionario de opinión sobre el uso del Método de Adaptación Jerárquica. Aunque se disponía de herramientas software como soporte al método⁸, se decidió que ambos grupos realizaran el proceso analizando la descripción a bajo nivel de las operaciones.

Tras la primera fase de validación, el primer grupo vio como mejorable la claridad en la descripción de algunas operaciones, lo que fue corregido antes de la segunda fase, y puntuaron con una nota muy alta la capacidad de las operaciones para recoger todas las posibilidades y variantes de cambios.

Tras la segunda fase de validación, el segundo grupo destacó especialmente la división de la especificación de cada workflow en dos ontologías, los distintos tipos de operaciones disponibles y, en especial, las operaciones de propagación así como el potencial que tiene su aplicación en administraciones de gran tamaño. Además confirmaron lo avanzado por el primer grupo sobre que el método es completo ya que ambos grupos han estado de acuerdo en que las operaciones son precisas y tienen en cuenta todos los elementos que deben modificarse.

⁷ Los detalles de este caso están disponibles, entre las páginas 289 y 309, de esta Tesis Doctoral: <http://uex.be/tdaeprieto>. Para un caso más completo de aplicación se recomienda revisar el caso mostrado entre las páginas 171 a 264 de esta misma tesis.

⁸ <http://uex.be/wdesigner> y <http://uex.be/wmanager>

Se puede afirmar que, con las opiniones recogidas de los grupos de expertos, los métodos y operaciones que componen el Método permiten su aplicación en la resolución del Problema de Adaptación Jerárquica.

Actualmente se está trabajando en la aplicación de este Método a algunos procesos dentro de la Escuela Politécnica de la Universidad de Extremadura, como, por ejemplo, la adaptación del proceso de presentación de quejas por parte de los estudiantes a partir de la normativa genérica para todos los centros de la Universidad de Extremadura.

5. Conclusiones

Se ha presentado una propuesta para abordar el Problema de Adaptación Jerárquica de workflows para procesos administrativos que principalmente sucede en administraciones públicas y empresas privadas con una estructura jerárquica.

Esta propuesta, denominada Método de Adaptación Jerárquica, se apoya en las ventajas que ofrece la especificación de workflows en ontologías a partir de *OntoMetaWorkflow*. Este método ofrece una adaptación flexible, en la cual, las restricciones que deben satisfacer un workflow genérico y sus adaptaciones, vendrán marcadas por la normativa que regula el proceso administrativo gestionado, y no por una noción de adaptación rígida previamente establecida que no tenga en cuenta las características propias de cada proceso.

Además, el método ofrece el conjunto completo de operaciones de propagación que se deben aplicar si la normativa que regula el proceso es modificada y, por tanto, se necesita cambiar el workflow genérico y transmitir estos cambios en cascada a sus workflows adaptados. Es decir, gracias a estas operaciones se va a facilitar que tanto el workflow genérico como los workflows adaptados satisfagan las restricciones fijadas por la normativa que regula el proceso, a pesar de los cambios que puedan suceder.

Referencias bibliográficas

- Aalst, W. M. P. Van Der, & Basten, T. (2002). Inheritance of workflows: an approach to tackling problems related to change. *Theoretical Computer Science*, 270(1-2), 125–203. doi: 10.1016/S0304-3975(00)00321-2
- Alonso, G., Agrawal, D., Abbadi, A. E., & Mohan, C. (1997). Functionality and Limitations of Current Workflow Management Systems. *IEEE Expert Intelligent Systems And Their Applications*, 12(5), 1–25.
- Bernstein, A., & Grosz, B. N. (2003). Beyond Monotonic Inheritance: Towards Semantic Web Process Ontologies.
- Choppy, C., Desel, J., & Petrucci, L. (2011). Specialisation and Generalisation of Processes. In *Proceedings of the workshop on Petri Nets and Software Engineering (PNSE'11)*, Newcastle, UK (pp. 109–123). CEUR-WS.
- Feldman, M. S., & Khademian, A. M. (2000). Managing for inclusion: Balancing control and participation. *International Public Management Journal*, 3(2), 149–167. doi: 10.1016/S1096-7494(01)00035-6

- Ferndrigger, S., Bernstein, A., Dong, J. S., Feng, Y., Li, Y.-F., & Hunter, J. (2008). Enhancing Semantic Web Services with Inheritance. In *Proceedings of the 7th International Conference on The Semantic Web* (pp. 162–177). Berlin, Heidelberg: Springer-Verlag.
- Georgakopoulos, D., Hornick, M., & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2), doi: 119–153. 10.1007/BF01277643
- Hollingsworth, D. (1995). *The Workflow Reference Model Document Number TCOO-1003 Document Status - Issue 1.1* (p. 55). Winchester, UK.
- McReady, S. (1992). There is more than one kind of Workflow Software. *Computerworld*, November 2, 85–90.
- Moore, J., Stader, J., Macintosh, A., Casson-du Mont, A., & Chung, P. (1999). Intelligent task management support for new product development in the chemical process industries. In *6th International Product Development Management Conference (PDM 99)* (pp. 787–796). Cambridge, UK.
- Prieto, A. E., & Lozano-Tello, A. (2012). Defining Reusable Administrative Processes Using a Generic Ontology. *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 22(2), 243–264. doi: 10.1142/S0218194012400050
- Prieto, Á. E., & Lozano-Tello, A. (2009). Use of Ontologies as Representation Support of Workflows Oriented to Administrative Management. *Journal of Network and Systems Management*, 17(3), 309–325. doi: 10.1007/s10922-009-9132-6
- Wyner, G. M., & Lee, J. (1995). Applying specialization to process models. *Proceedings of Conference on Organizational Computing Systems.*, 290 – 301.
- Wyner, G. M., & Lee, J. (2005). Applying Specialization to Petri Nets: Implications for Workflow Design. In *Business Process Management Workshops. BPM 2005 International Workshops, Nancy, France, September 5, 2005. Revised Selected Papers* (Vol. 3812, pp. 432–443). Berlin, Heidelberg: Springer Berlin Heidelberg.