

# Hacia una estandarización de workflows científicos e industriales con BPMN2

**Corina Abdelahad, Daniel Riesco**  
*Departamento de Informática Universidad  
Nacional de San Luis, San Luis, Argentina*  
{cabdelah, driesco}@unsl.edu.ar

**Alessandro Carrara, Carlo Comin, Carlos Kavka**  
*Research and Development Department ESTECO  
SpA Trieste, Italy*  
{carrara, comin, kavka}@esteco.com

## Abstract

*La automatización de procesos de negocio es un componente clave para el crecimiento de las organizaciones. Existen muchos estándares en el ámbito de procesos de negocio pero no son directamente aplicables al campo de la ingeniería, ya que estos poseen requerimientos específicos. BPMN (Business Process Model and Notation) ampliamente utilizado para el modelado de procesos de negocio ha sido recientemente mejorado permitiendo extender su metamodelo. Por otro lado, uno de los principales obstáculos en el modelado de workflows ingenieriles es que no existen estándares para esto. Las transformaciones desempeñan un papel fundamental en la implementación de Arquitectura Dirigida por Modelos, proporcionando un mecanismo para expresar el refinamiento de modelos. Query / View / Transformation es el estándar de la OMG establecido para crear consultas, vistas y transformaciones de modelos. Sus relaciones permiten formalizar las transformaciones dirigidas por modelos. ESTECO es una compañía la cual ha desarrollado su propio modelador y motor workflow. Aunque su modelo propietario ha demostrado ser útil en el contexto de los procesos de ingeniería, la utilización de un estándar permitirá la construcción de modelos unificados y estandarizados. En este trabajo se presenta una transformación parcial utilizando QVT entre los metamodelos de ESTECO y BPMN2, donde los eventos y las tareas de ESTECO son transformados a eventos y tareas de BPMN, utilizando extensiones para soportar las propiedades específicas de ellas. La transformación completa permite la conversión de la mayoría de los workflows industriales de ESTECO, asegurando la portabilidad entre las herramientas que soportan el estándar BPMN2.*

**Palabras Clave** - BPMN 2.0; workflow de negocio; workflow industrial; transformación; QVT

## Introducción

La Arquitectura Dirigida por Modelos (Model Driven Architecture - MDA) [1] [2], implementación de la OMG (Object

Management Group) para la Ingeniería Dirigida por Modelos (Model-Driven Engineering- MDE) [3] [4], propone un proceso de desarrollo de software en el cual la clave son los modelos y las transformaciones entre ellos. En este proceso, el software es desarrollado construyendo uno o más modelos, y transformando estos en otros modelos o transformarlos hasta llegar a código ejecutable. Teniendo en mente este objetivo la OMG propone un lenguaje estándar para expresar consultas y definir transformaciones de modelos, este lenguaje es QVT (Query/View/ Transformation). Su especificación depende de otros dos estándares de la OMG como son MOF y OCL [5]. La utilización de QVT para especificar transformaciones, aporta reutilización de tecnologías que siguen estándares de la OMG. Con QVT es posible definir transformaciones genéricas entre metamodelos, de esta manera cualquier instancia de un metamodelo puede ser transformado en una instancia de otro metamodelo. La idea de QVT se basa principalmente en la definición de un lenguaje para las consultas (Queries) sobre los modelos MOF, la búsqueda de un estándar para generar vistas (Views) que muestran aspectos específicos de los sistemas modelados, y la definición de un lenguaje para la descripción de transformaciones (Transformations) de modelos MOF.

Por otro lado la OMG junto con BPMI (Business Process Modeling Initiative) han desarrollado la notación BPMN para el modelado de procesos de negocio [6] [7]. BPMN se considera la notación más

comprensible y utilizada para el modelado de procesos de negocio. Ésta es una plataforma independiente y define una notación para la definición de los procesos. BPMN también permite la colaboración entre procesos de negocios de distintas organizaciones, es decir permite la interrelación con otro proceso de negocio independiente que se ejecuta en otra empresa. BPMN2 es la última definición del estándar BPMN para modelar procesos de negocios. Éste ha sido desarrollado teniendo en cuenta algunas limitaciones y dándole mayor importancia a los objetos de datos persiguiendo el objetivo de poder interpretar y ejecutar directamente los modelos.

El objetivo principal de BPMN es proveer una notación realmente entendible para todos los usuarios del negocio, desde los analistas hasta las personas del negocio quienes manejarán los procesos. Por lo tanto, BPMN crea un puente estandarizado para la brecha entre el diseño de procesos de negocio y la implementación de estos. Un workflow industrial es un proceso de negocio automatizado generalmente utilizado para ejecutar tareas de procesamiento complejas que requieren muchas características que la mayoría de los modelos de procesos de negocio no soportan actualmente [8]. Este tipo de workflows son ampliamente utilizados en las ciencias naturales, las simulaciones computacionales, la medicina, las ciencias ambientales, la ingeniería, la geología, la astronomía, la industria automotriz y aeroespacial, entre otros campos.

Como se dijo anteriormente BPMN2 se ha desarrollado tomando como uno de sus objetivos la superación de algunas limitaciones. La definición de este nuevo estándar permite, por primera vez, extender el uso de workflows en el ámbito de procesos de negocio para el campo de la ingeniería [6][9]. Los workflows ingenieriles, comparten muchas características con los conocidos workflows científicos, muy utilizados en la industria hoy en día. Si bien estos workflows son

ampliamente utilizados, actualmente no existe ningún estándar para la definición de los mismos. Con el fin de construir modelos unificados y estandarizados, los workflows ingenieriles de la empresa ESTECO serán transformados al nuevo estándar permitiendo además la ejecución de los mismos.

### **Trabajos relacionados**

Varios trabajos en el campo de la ingeniería de software se relacionan con el concepto de transformaciones entre modelos, y muchos de ellos utilizan BPMN para modelar procesos de negocio. Hasta el momento no se ha encontrado un trabajo de investigación el cual considere BPMN2 como el modelo destino para la transformación en el contexto de workflows industriales. Persiguiendo ese objetivo, en [10] se ha presentado una transformación modelo a modelo utilizando QVT entre un workflow científico y BPMN 2.0. La misma permite convertir los datos de entrada y salida en el formato de destino. El trabajo fue validado experimentalmente en un entorno de ingeniería con el apoyo de una empresa en el campo de la optimización multi-objetivo.

En [11] los autores presentan una investigación sobre qué factores tienen un impacto en el rendimiento de la ejecución de la transformación. Este estudio estima el rendimiento de una transformación y permite elegir entre las implementaciones alternativas para lograr mayor rendimiento. Los resultados de este estudio pueden ser utilizados por los implementadores de los motores de transformación con el fin de mejorar el conjunto de herramientas disponibles en la actualidad.

En una misma línea, una transformación modelo a modelo entre PICTURE y BPMN2 se presenta en [12]. PICTURE es un lenguaje de modelado de procesos de dominio específico para el sector de la administración pública. La transformación permite modelar procesos administrativos en PICTURE y obtener modelos BPMN2

para estos automáticamente, ayudando al gobierno electrónico haciendo posible la implementación de procesos de soporte.

Los autores en [13], presentan tres conjuntos de relaciones QVT como un medio de transformaciones implementados por un método dirigido por modelos para desarrollo web. Un conjunto transforma un modelo de entrada de alto nivel a un modelo abstracto específico para web. Los otros dos conjuntos transforman el modelo abstracto para web a modelos específicos para plataformas web.

Un ejemplo de aplicación se presenta en [14] para mostrar una transformación automatizada de un modelo de proceso de negocio en un modelo de desempeño parametrizado, obteniendo así importantes ventajas sobre la fácil personalización y una mejor automatización.

En [15], se presenta la generación de componentes de plataforma de negocios Java EE6 desde procesos de negocio técnicos modelados con BPMN2. La generación se obtuvo mediante la realización de tres transformaciones en el contexto MDA, realizadas con QVT Relations y MOFScript. Este trabajo contribuye mejorando la productividad y la reducción de los errores de diseño.

## **Motivación**

Tanto los workflows científicos como los ingenieriles difieren en muchos aspectos con los workflows de negocio. Por ejemplo, los workflows de negocio suelen tratar con transacciones discretas, en cambio los workflows científicos o ingenieriles suelen trabajar con muchas herramientas de software interconectadas, grandes cantidades de datos con múltiples fuentes de datos y en múltiples formatos. Además, por lo general los servicios ingenieriles tienen una larga duración de ejecución y dependen del entorno de ejecución. Incluso si bien, los workflows ingenieriles se han utilizado con éxito desde hace muchos años, la mayoría de las herramientas que se

utilizan para definirlos y ejecutarlos no están basados tecnologías estándar.

Como hipótesis en este trabajo, se intenta demostrar que es posible que un modelo de workflow ingenieril sea modelado con la nueva versión del estándar de BPMN, ya que ésta ha sido desarrollada teniendo en cuenta algunas limitaciones y dándole mayor importancia a los objetos de datos persiguiendo el objetivo de poder interpretar y ejecutar directamente los modelos.

Hay que considerar que existe una gran base de workflows ingenieriles ya diseñados y utilizados actualmente por la industria que no pueden descartarse. Con el fin de proporcionar soporte para estos workflows, se ha propuesto un método para la transformación de los workflows propietarios existentes a workflows estándar BPMN2. Este enfoque proporcionará un incentivo adicional para que las empresas abandonen las herramientas de modelado propietarias y comiencen a utilizar tecnologías estándar que vienen de la mano de los procesos de negocio.

Sin embargo, las transformaciones no son directas. Hay actividades que poseen propiedades adicionales y requisitos que los procesos de negocios estándar no poseen. Afortunadamente, BPMN2 ha definido un mecanismo de extensión que permite añadir construcciones necesarias sin romper el cumplimiento del estándar.

Como parte del método, este trabajo presenta una transformación parcial con el fin de llevar a cabo una conversión de un workflow de ESTECO básico a BPMN2.

## **Arquitectura**

El objetivo de este trabajo es aplicar los últimos conceptos de procesos de negocio en el campo de los workflows ingenieriles. El uso de estándares en la industria es importante, ya que garantiza la portabilidad entre las herramientas que soportan BPMN2 y permite la construcción de modelos unificados y estandarizados.

Los workflows industriales de ESTECO [16] poseen una representación en XML, lo que permite el uso de herramientas como MediniQVT [17] para la transformación. No existe una correspondencia directa entre los componentes de los workflows de ESTECO y los constructores de BPMN2, no sólo porque los nodos de datos se manejan de manera diferente, sino que las actividades de los modelos de ESTECO poseen propiedades particulares que los modelos BPMN2 no poseen. Estas últimas sólo pueden ser manejadas a través del mecanismo de extensión que provee BPMN2.

Para especificar las reglas de transformación se definió una correspondencia entre las metaclasses del metamodelo ESTECO y las metaclasses del metamodelo BPMN2. Para definir estas reglas se utilizó QVT.

Las transformaciones QVT describen relaciones entre el metamodelo fuente y el metamodelo destino, ambos especificados en MOF. La transformación definida se aplica al modelo fuente, el cual es una instancia del metamodelo de ESTECO, para obtener un modelo destino, el cual será una instancia del metamodelo BPMN2, ver Figura 1.

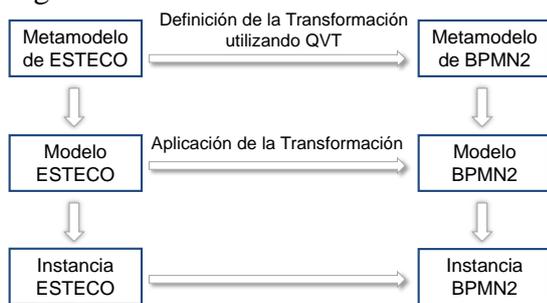


Figura 1: Arquitectura

Los metamodelos utilizados en la definición de la transformación se muestran en el nivel superior. Los modelos específicos a los cuales se les aplicará la transformación definida para obtener modelos BPMN2 se encuentran en el nivel intermedio. El nivel inferior representa las instancias de los modelos, los cuales serán ejecutados en el correspondiente motor workflow.

## Transformaciones y Metamodelos

Todo workflow de negocio o ingenieril deben contener eventos (de inicio y fin), actividades y flujos de ejecución. En este trabajo se mostrará cómo se transforman el nodo que representa el evento de inicio, los nodos que representan las actividades junto con sus propiedades y los conectores que representan los flujos de ejecución.

Una transformación especifica un grupo de relaciones que los elementos de los modelos involucrados deben satisfacer. Una transformación puede tener cualquier número de parámetros de entrada y salida llamados dominios. Los parámetros de entrada corresponden a los modelos a los cuales se les aplicará la transformación para generar los parámetros de salida. Para cada parámetro de salida, una nueva instancia de modelo será creada y/o modificada de acuerdo al metamodelo al que pertenece dicho modelo destino. Cada dominio identifica una correspondencia de un conjunto de elementos definidos llamados patrones.

Una relación en QVT define reglas de transformación, y esa relación implica la existencia de clases para cada uno de sus dominios. Las mismas pueden contener cláusulas *when* y *where*. Un dominio puede ser marcado como *checkonly* o *enforce*. Un patrón puede ser visto como un conjunto de variables y un conjunto de restricciones que los elementos de los modelos deben satisfacer. Un tipo de modelo está definido por un metamodelo y un conjunto opcional de expresiones de condición. El lector interesado puede consultar estos conceptos en la especificación QVT [18].

Para ilustrar cómo se van transformando los elementos, a continuación se mostrará las relaciones referidas a la transformación de un nodo de inicio, de una tarea y de un flujo de ejecución.

La transformación comienza definiendo el nombre de la misma junto con los modelos candidatos y sus metamodelos. La Figura 2 muestra la transformación de ESTECO a BPMN2, la cual toma como modelo origen

un modelo ESTECO, siendo una instancia del metamodelo ESTECO, y produce como modelo destino un modelo BPMN2, siendo una instancia del metamodelo BPMN2. Además, se presenta la relación *createProcess*, la cual constituye el punto de entrada para comenzar la transformación en QVT. La finalidad de esta relación es crear la etiqueta *Definitions*, etiqueta principal en cualquier modelo BPMN2 la cual contiene la etiqueta *Process*, además de invocar las otras relaciones encargadas de completar la transformación para la creación del modelo BPMN2.

```

transformation ESTECOToBPMN2(source: esteco_m,
                             target: bpmn2){
  top relation createProcess {
    checkonly domain source g:esteco_m::TGeometry{};
    enforce domain target d : bpmn2::Definitions{
      rootElements = r : bpmn2::Process{
        name = 'EstecoMainProcess',
        id = 'EstecoMainProcess',
        isExecutable = true,
        processType = bpmn2::ProcessType::Private
      }
    };
    ...
    where {
      createSequenceFlowStart2Activity(g, r);
      taskNode2taskBPMN2(g, r);
      ...
    }
  }
  ...
}

```

Figura 2: Definición de la transformación

El dominio definido como *enforce* corresponde al modelo BPMN2, lo cual significa que este modelo será una instancia del metamodelo BPMN2. La Figura 3 muestra parte de este metamodelo donde se observan los elementos utilizados en la Figura 2.

Una vez creado el proceso con sus etiquetas, la transformación se dirige hacia la cláusula *where* para chequear que la relación *createSequenceFlowStart2Activity* y la relación *taskNode2taskBPMN2* se satisfagan. Estas relaciones definen el mapeo entre los elementos del metamodelo ESTECO y los elementos del metamodelo BPMN2 relacionados con el evento de inicio y las tareas respectivamente.

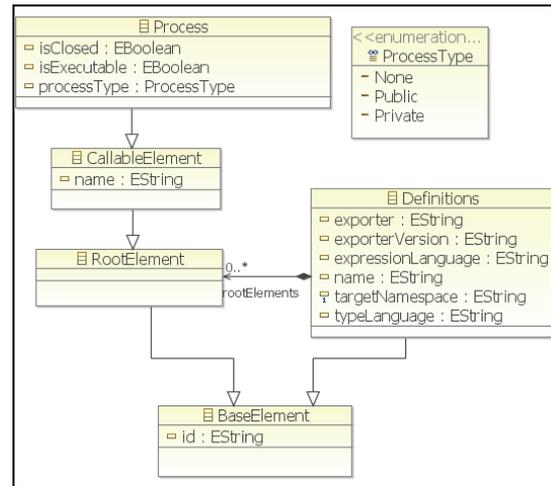


Figura 3: Elementos BPMN2 involucrados en la relación *createProcess*

La Figura 4 muestra la relación *createSequenceFlowStart2Activity*. Esta relación define la transformación de un evento de inicio de ESTECO hacia una tarea relacionados a través de un conector.

```

relation createSequenceFlowStart2Activity{
  e_id, e_name, id_end_connector :String;
  checkonly domain source g : esteco_m::TGeometry{
    taskNode = task : esteco_m::TTaskNode {
      startConnector =
        start_connector : esteco_m::TStartConnector {}
    },
    processEdge =
      process_edge : esteco_m::TProcessEdge {},
    startNode = s:esteco_m::TStartNode {
      id = e_id,
      name = e_name,
      endConnector = e : esteco_m::TEndConnector{
        id= id_end_connector }
    }
  };
  enforce domain target p:bpmn2::Process{
    flowElements = f : bpmn2::StartEvent{
      id= e_id,
      name= e_name },
    flowElements = flow : bpmn2::SequenceFlow {
      sourceRef = f,
      targetRef = task2 : bpmn2::Task {
        id = task.id }
    }
  };
  when { if (id_end_connector = process_edge.from and
            start_connector.id = process_edge.to )
    then true
    else false
  endif;
}
}

```

Figura 4: Relación para transformar un Evento de Inicio

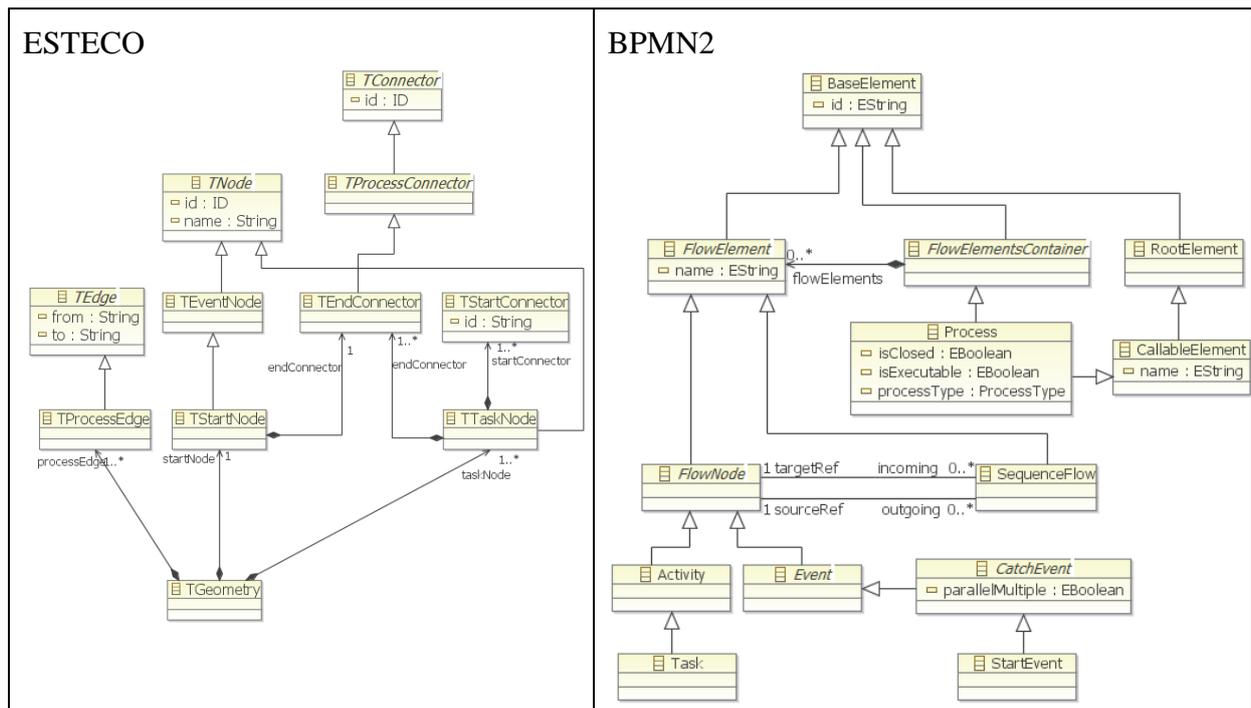


Figura 5: Elementos de los metamodelo involucrados en la relación *createSequenceFlowStart2Activity*

En esta relación puede observarse como un nodo de inicio (*startNode*) se transforma en un evento de inicio (*startEvent*). El nodo de inicio posee un conector de fin (*endConnector*) el cual ayudará a determinar hacia donde se dirige la ejecución. Un nodo de tarea (*taskNode*) posee un conector de inicio (*startConnector*) el cual también ayudará a determinar desde donde proviene la ejecución. El nodo *processEdge* es el encargado de relacionar el conector de fin (del nodo de inicio) y el conector de inicio (del nodo de tarea). Ambos conectores asisten al objeto *processEdge* para generar la conexión entre el nodo de inicio y el nodo de tarea. En conclusión se puede observar cómo el objeto *processEdge* corresponde a un flujo de secuencia de BPMN2.

En la Figura 5 se puede ver parte de los metamodelos involucrados en la relación antes vista.

De manera similar se genera la transformación del nodo de fin junto con su flujo de ejecución.

Continuando con la relación de la Figura 2, se puede observar que la cláusula *where*

también posee la relación *taskNode2taskBPMN2*. Esta relación define el mapeo entre las tareas de ESTECO y las tareas del BPMN2, ver Figura 6.

```

relation taskNode2taskBPMN2 {
  id_task, name_task, charset_task, type_task : String;
  valueType:String;
  checkonly domain source g:esteco_m: TGeometry {
    taskNode = t :esteco_m::TTaskNode{
      id= id_task,
      name= name_task,
      charset= charset_task,
      type= type_task
    }
  };
  enforce domain target p:bpmn2::Process {
    flowElements = f :bpmn2::Task {
      name = name_task,
      id = id_task,
    }
  };
  where{
    valueType = if (type_task =
it.esteco.integration.calculator.workflow.CalculatorNode')
then 'Calculator'
    ...
    createSequenceFlow(g.id_task, p);
    if (valueType='Calculator')then
    propertyCalculator(g, name_task,charset_task,f, id_task)
    ...
  }
}

```

Figura 6: Relación para transformar una Tarea

En esta relación se puede destacar 3 partes. La primera parte hace referencia al dominio ESTECO marcado con *checkonly*. Aquí se muestra el procedimiento para extraer los valores referidos a las tareas pertenecientes a la instancia del metamodelo ESTECO (modelo origen) tales como su identificador (*id*), su nombre (*name*), una propiedad específica (*charset*) y el tipo de la tarea (*type*). La segunda parte hace referencia al dominio BPMN2 marcado como *enforce*. Esto es, si la comprobación falla, se modificará el modelo destino, creando, borrando o modificando sus elementos para lograr satisfacer la relación. En este caso se crearán elementos en el modelo BPMN2 (modelo destino). La Figura 7 muestra los elementos de los metamodelos involucrados en esta relación.

La transformación de tareas implica la transformación y/o generación de otros elementos los cuales están involucrados en ellas.

ESTECO posee distintos tipos de tareas, cada una de estas posee propiedades específicas. Es decir, dependiendo del tipo de la tarea deben crearse propiedades particulares. La distinción de estos tipos pueden observarse en la clausula *where*, tercera parte de la relación. A través de una condición se le asignará a una variable el

tipo de tarea, en este caso un *Calculator*, para luego generar las propiedades específicas de él. Además se observa que aparece la relación *createSequenceFlow*. La misma se encargará del proceso de transformación relacionado a los flujos de ejecución y las tareas, en particular esta relación genera un flujo de secuencia entre la tarea que se está transformando y la tarea anteriormente ejecutada.

La cláusula *where* finaliza con una sentencia de condición la cual chequea el tipo de tarea. Dependiendo de ésta se chequeará una relación encargada de generar las propiedades específicas de la misma (*propertyCalculator*).

A continuación se explica las relaciones *createSequenceFlow* y *propertyCalculator*, respectivamente.

Los elementos de un modelo deben estar conectados para seguir la ejecución. Como los workflows de ESTECO pertenecen a un único pool, la conexión entre los elementos BPMN2 se realiza a través de flujos de secuencia. La relación *createSequenceFlow* define el mapeo entre un flujo de ejecución de ESTECO y uno de BPMN2 entre actividades, ya que el flujo de secuencia entre un evento y una actividad es tratada en la relación que mapea los eventos de

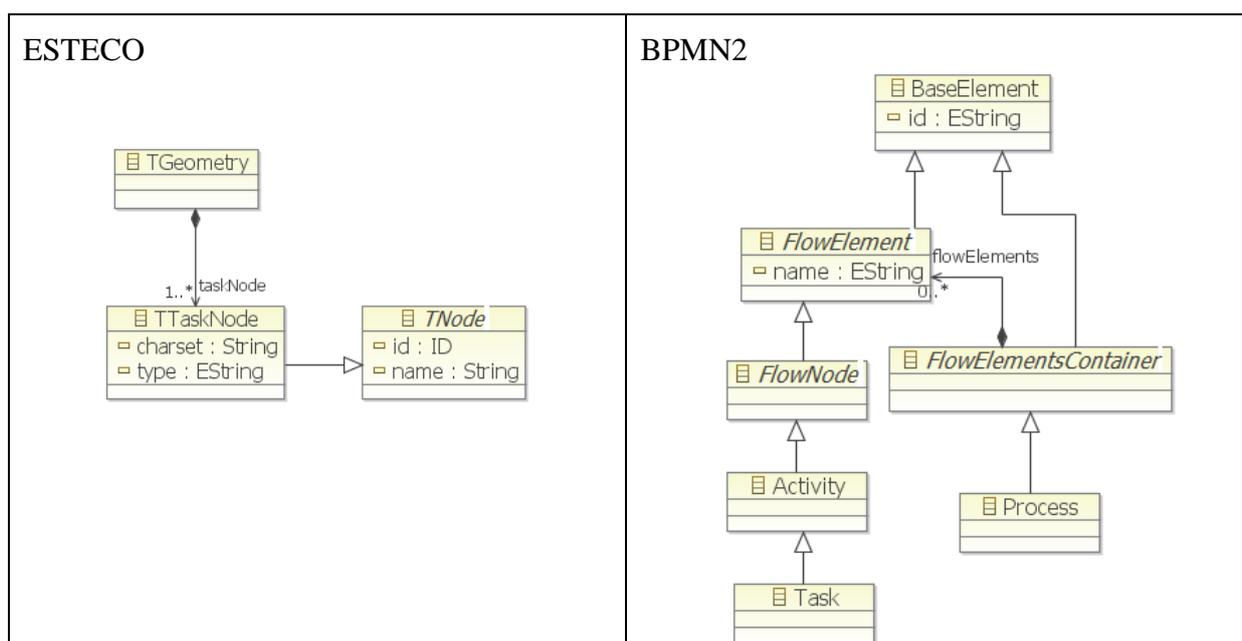


Figura 7: Elementos de los metamodelo involucrados en la relación *taskNode2taskBPMN2*

inicio y fin respectivamente. La Figura 8 muestra la relación antes mencionada.

```

relation createSequenceFlow{
  id_task_node, id_connector:String;
  checkonly domain source g:esteco_m::TGeometry{
    taskNode = task_node :esteco_m::TTaskNode{
      id = id_task_node,
      endConnector =
        end_connector : esteco_m::TEndConnector {
          id=id_connector
        }
    },
    processEdge =
      process_edge : esteco_m::TProcessEdge{
    };
  };
  primitive domain id_task:String;
  enforce domain target p :bpmn2::Process {
    flowElements = task : bpmn2::Task{
      id=id_task
    },
    flowElements = flow1 : bpmn2::SequenceFlow {
      sourceRef = task2 : bpmn2::Task {
        id= id_task_node
      },
    },
    targetRef = task
  }
};
when {
  if (id_connector = process_edge.from and
    process_edge.to = id_task + '_0') then
    true else
    false
  endif;
}
}

```

Figura 8: Relación para transformar un Flujo de Secuencia

En el dominio marcado como *checkonly* (modelo origen) se puede observar cómo se obtiene el identificador de la tarea (*id*) y como a través del atributo *endConnector* se obtiene el identificador el cual ayudará para saber desde donde proviene la ejecución. El elemento *processEdge* pertenece a la clase *TProcessEdge* la cual hereda los atributos de la clase *TEdge*, ver Figura 9, esta clase posee dos atributos: *from* y *to*, estos atributos poseen los identificadores los cuales son el origen y destino de un flujo de ejecución.

Si se observa, en la relación de la Figura 6 la relación *createSequenceFlow* posee tres parámetros, el primero hace referencia al elemento del tipo *TGeometry* (*g*), el segundo hace referencia al identificador de la tarea a la cual se está transformando y el tercero hace referencia al elemento del tipo *Process* (*p*).

En la relación de la Figura 8 se observa un dominio primitivo *String*, este es necesario para “capturar” el identificador que aparece como segundo parámetro en la relación y de este modo no “perder” el identificador de la tarea que se está transformando.

En el dominio marcado como *enforce* se puede observar cómo se creará el flujo de

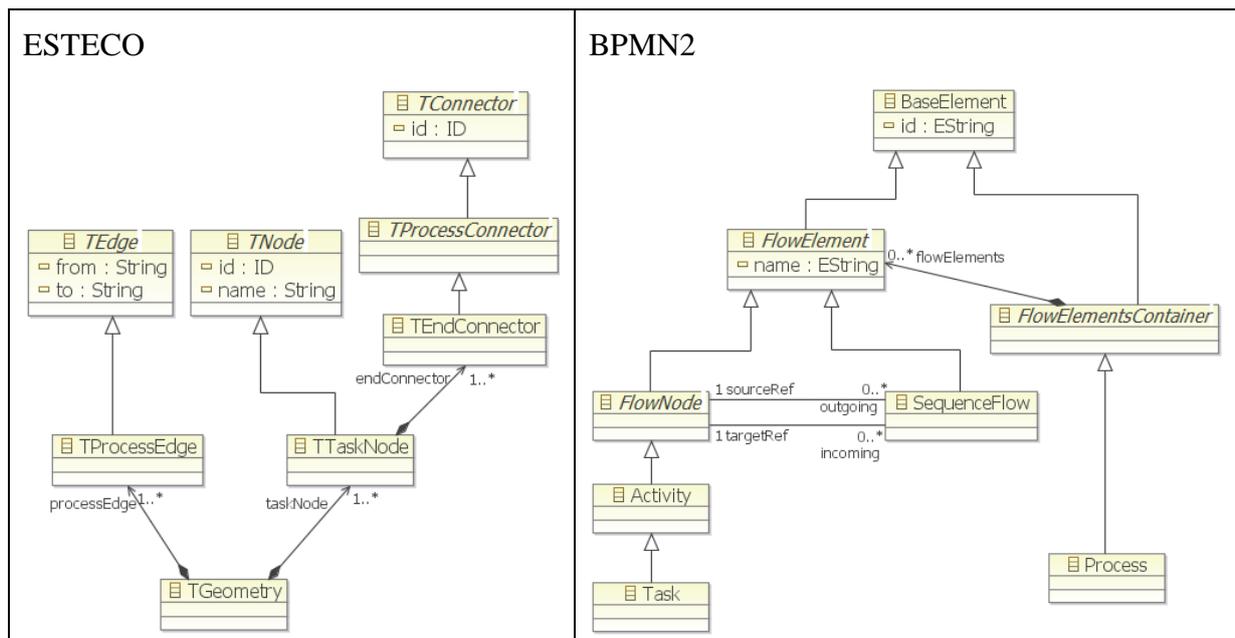


Figura 9: Elementos de los metamodelo involucrados en la relación *createSequenceFlow*

secuencia en el modelo destino. Como se ve en la Figura 9 la clase *SequenceFlow* posee dos atributos *sourceRef* y *targetRef*, estos atributos contendrán el origen y destino del flujo de secuencia. De éste modo se observa en la relación como se le asigna al atributo *sourceRef* una tarea y al atributo *targetRef* la tarea transformada en la relación antes vista. En la cláusula *when* se observa una sentencia condicional la cual controla que las tareas estén conectadas.

De esta manera, se puede observar cómo se construyó una relación la cual transforma un flujo de ejecución ESTECO a un flujo de secuencia BPMN2 entre tareas.

Como se mencionó anteriormente, la cláusula *where* de la Figura 6 tiene una condición que controla el tipo de la tarea. Dependiendo de esta condición deberán ser creadas las propiedades para cada tipo de tarea. La relación *propertyCalculator* es la encargada de crear las propiedades de la tarea *Calculator*, ver Figura 10. Por simplicidad esta Figura muestra una de las propiedades. Todos los dominios, a excepción del marcado como *enforce*, previos a la cláusula *where* son utilizados para mantener datos necesarios para crear las propiedades.

```

relation propertyCalculator{
  checkonly domain source g : esteco_m::TGeometry{};
  primitive domain name_task:String;
  primitive domain charset_task:String;
  enforce domain target t :bpmn2::Task { };
  primitive domain id_task:String;
  where {
    propertyCharset(g, charset_task, t);
  }
  ...
}

```

Figura 10: Relación para crear propiedades de la Tarea *Calculator*

Por una cuestión de claridad se explicará sólo una de las propiedades de la tarea *Calculator*.

Todas las propiedades de los tipos de las tareas de ESTECO poseen un tipo particular y valores los cuales, o vienen proporcionados por la configuración a la hora de generar el workflow o son valores

asignados por defecto. En la Figura 11, se puede observar cómo se genera la propiedad *charset* de *Calculator*. En la relación se puede observar como el segmento que enmarca el dominio *enforce* agrega una propiedad a la tarea, en este caso *charset*. Aquí se puede ver como se agrega el nombre de la propiedad junto con el tipo al cual hace referencia la misma (*itemSubjectRef*), en este caso hace referencia a un *String*. El estándar de BPMN2 no permite asignarles valores a las propiedades, para soportar esta carencia se utiliza el mecanismo de extensión que este provee. Para esto es necesario hacer referencia a la definición de los tipos en el metamodelo ESTECO. Parte de esta definición es presentada en la Figura 12, donde la clase *DocumentRoot*, posee una relación de generalización con la clase *ExtensionAttributeValue*, clase perteneciente a la definición de BPMN2. En la relación puede observarse como se generan los atributos para asignarle el valor a la propiedad a través del atributo *value*, este atributo pertenece a la clase *TSimpleValue* la cual es parte de la clase *TDefault* que a su vez es parte de la clase *DocumentRoot*, ver Figura 12.

```

relation propertyCharset {
  checkonly domain source g : esteco_m::TGeometry{};
  primitive domain charset_task:String;
  enforce domain target t :bpmn2::Task {
    properties = charset : bpmn2::Property {
      name = 'charset',
      itemSubjectRef = item : bpmn2::ItemDefinition{
        id = 'StringItemDefinition'
      },
      extensionValues = extension : esteco::DocumentRoot{
        default = default : esteco::TDefault {
          simpleValue = simple_value : esteco::TSimpleValue {
            value = 'calculator.charset.' + charset_task
          }
        }
      }
    }
  };
}

```

Figura 11: Relación para crear la propiedad *charset*

De modo similar se generan todas las otras propiedades de *Calculator*.

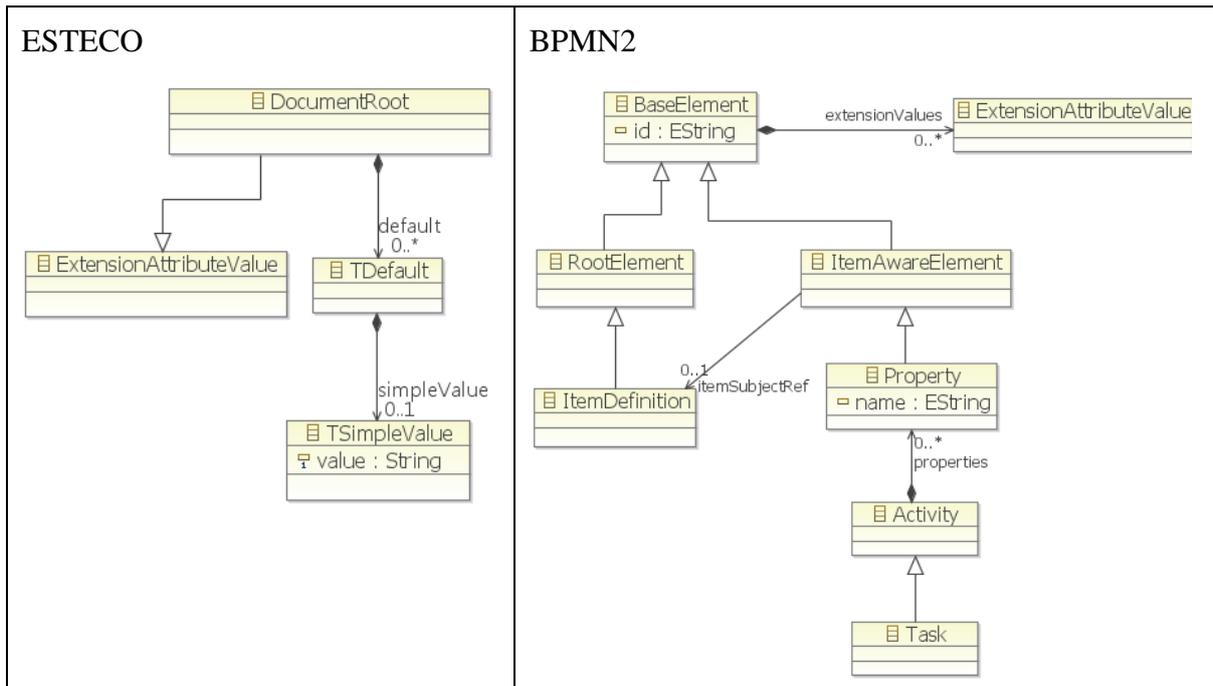


Figura 12: Elementos de los metamodelo involucrados en la relación *propertyCharset*

De esta manera se mostró como se transforma una tarea de ESTECO en BPMN2, junto con sus propiedades y el flujo de ejecución entre una tarea y otra. Con estas relaciones es posible transformar cualquier workflow básico de ESTECO a BPMN2.

### Un Experimento de transformación

En esta sección se presenta una correspondencia de una transformación utilizando el código QVT visto anteriormente. Las transformaciones QVT fueron definidas con el lenguaje Relations QVT en la herramienta MediniQVT, herramienta desarrollada por tecnologías IKV++ con una integración Eclipse [17]. El paquete *Process* contiene clases que se utilizan para modelar el flujo de actividades, eventos, compuertas, y flujos de secuencia dentro de un proceso. Un proceso debe estar contenido dentro de la etiqueta *Definitions* [6]. Un proceso es instanciado cuando ocurre un evento de inicio. El evento de fin indica dónde terminará el proceso. Los flujos de secuencia indican el orden de ejecución del workflow. Las actividades representan

tareas a realizar.

La Figura 13 presenta los resultados de la ejecución de una transformación aplicada a un workflow básico de ESTECO. Cada cuadro corresponde a un elemento XML, y la jerarquía entre los elementos es representada con la estructura de árbol.

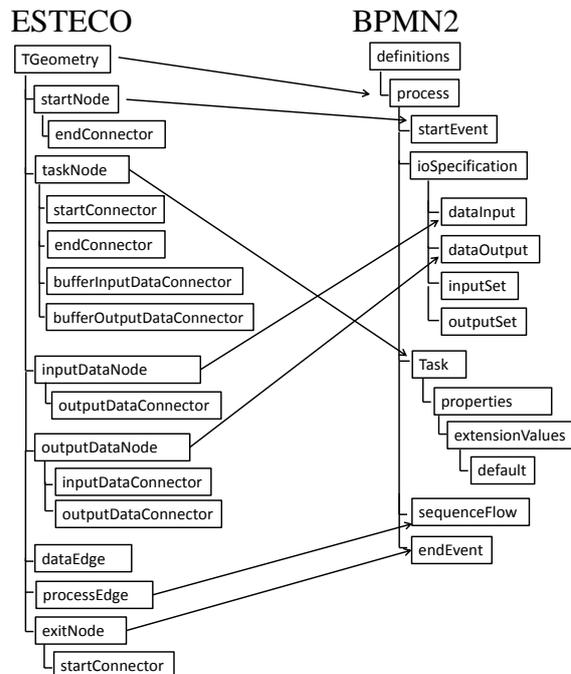


Figura 13: Correspondencia entre los elementos XML durante una transformación teniendo en cuenta una sola actividad

Se puede observar en la parte de ESTECO que el elemento principal es TGeometry. Este posee los nodos de inicio y fin, las tareas, los datos de entrada (*inputDataNode*), los datos de salida (*outputDataNode*) y los elementos que asisten para conectar los eventos, las tareas y los datos (*processEdge* y *dataEdge*). Además puede observarse en la Figura 13 que cada elemento de TGeometry posee conectores que asisten a los elementos *dataEdge* y *processEdge*.

En la parte de BPMN2 se observa que el elemento principal es *definitions* y con él el elemento *process* el cual contiene los eventos de inicio y fin, las tareas, flujos de secuencia y el objeto *ioSpecification* el cual contiene los datos de entrada y salida que participan en el proceso.

En la transformación de las tareas es necesario utilizar en más de una oportunidad el mecanismo de extensión de BPMN2 para satisfacer las necesidades específicas de ESTECO. Como se mencionó anteriormente las tareas poseen propiedades específicas dependiendo del tipo de tarea. Estas propiedades se representan con el elemento *properties* de BPMN2 junto con elementos propios de ESTECO.

## Conclusión

Hay que considerar que existe una gran base de workflows ingenieriles ya diseñados y utilizados actualmente por la industria que no pueden descartarse.

En este trabajo hemos propuesto un método para la transformación QVT de los workflows propietarios existentes a workflows estándar BPMN2. Este enfoque proporcionará un incentivo adicional para que las empresas abandonen los modeladores propietarios y comiencen a utilizar tecnologías estándares que vienen de la mano de los procesos de negocio.

Además se presentó un ejemplo en donde se transforman los elementos básicos utilizados en un workflow ingenieril. La propuesta ha sido validada

experimentalmente en un entorno de ingeniería soportado por la empresa especializada en optimización multi-objetivo. Este tipo de transformación permite la conversión de muchos workflows industriales de ESTECO a BPMN, permitiendo su ejecución en motores workflows BPMN con las adecuadas extensiones para soportarlos.

El objetivo de este trabajo es aplicar importantes conceptos de procesos de negocio en el campo industrial. Además, se pretende mostrar la importancia de la utilización de estándares en los sectores industriales con el fin de garantizar portabilidad entre las herramientas que soportan BPMN. En cuanto a un objetivo más general, se espera que el uso de un estándar para workflows científicos e ingenieriles facilite la colaboración entre científicos y diseñadores industriales, mejorando la interacción entre los diferentes campos, proporcionando así un vocabulario común en las publicaciones científicas e ingenieriles [19].

## Referencias

- [1] Object Management Group, MDA Guide, v1.0.1, <http://www.omg.org/cgi-bin/doc?omg/03-06-01> último acceso 26/07/2013
- [2] Anneke Kleppe, Jos Warmer, Wim Bast, "MDA Explained: The Model Driven Architecture™: Practice and Promise" Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
- [3] Stahl, T. and Völter, M. "Model-Driven Software Development". John Wiley & Sons, Ltd. (2006)
- [4] Favre Jean-Marie, Estublier Jacky, Blay Mireille. "Beyond MDA : Model Driven Engineering" (L'Ingénierie Dirigée par les Modèles : au-delà du MDA) Edition Hezmes-Lavoisier, ISBN 2-7462-1213-7. (2006).
- [5] Object Management Group (OMG), "Modeling and metadata specifications", <http://www.omg.org/spec>, último acceso 26/07/2013
- [6] OMG document number: Formal/2011-01-03 <http://www.omg.org/spec/BPMN/2.0/> último acceso 26/07/2013
- [7] Business Process Modeling Initiative <http://www.bpmi.org/> último acceso 26/07/2013
- [8] Li Hongbiao, Li Feng, and Yu Wanjun, "The

research of scientific workflow engine”, IEEE International Conference on Software Engineering and Service Sciences (ICSESS), 2010, pp. 412-414.

[9] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers, “Examining the challenges of scientific workflows”. IEEE Computer vol. 40, no. 12, 2007, pp. 24-32.

[10] Corina Abdelahad, Daniel Riesco, Carlo Comin, Alessandro Carrara, and Carlos Kavka, “Data Transformations using QVT between Industrial Workflows and Business Models in BPMN”, Proceedings of the Seventh International Conference on Software Engineering Advances ICSEA, 2012, IARIA.

[11] Marcel van Amstel, Steven Bosems, Ivan Kurtev, and Luís Ferreira Pires, “Performance in model transformations: experiments with ATL and QVT”, Lecture Notes in Computer Science, Volume 6707, Theory and Practice of Model Transformations, Springer, 2011, pp. 198-212.

[12] Henning Heitkoetter, “Transforming PICTURE to BPMN 2.0 as part of the model-driven development of electronic government systems”, 44th Hawaii International Conference on System Sciences (HICSS), 2011, pp. 1-10.

[13] Ali Fatollahi, Stéphane Somé, and Timothy Lethbridge, “Automated generation of abstract web models using QVT relations”, Technical Report TR-2010-06, School of Information Technology and Engineering, University of Ottawa, September 2010.

[14] Paolo Bocciarelli and Andrea D'Ambrogio, “A BPMN extension for modeling non functional properties of business processes”, TMS-DEVS'11 Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, Springer-Verlag, 2011, pp. 160-168.

[15] Narayan Debnath, Carlos Alejandro Martinez, Fabio Zorzan, Daniel Riesco, and German Montejano, "Transformation of business process models BPMN 2.0 into components of the Java business platform". Industrial Informatics (INDIN), 10th IEEE International Conference on Digital Objects, 2012, pp. 1035-1040, IEEE

[16] ESTECO S.p.A., “modeFRONTIER applications across industrial sectors involving advanced CAD/CAE packages”, [http://www.esteco.com/home/mode\\_frontier/by\\_industry](http://www.esteco.com/home/mode_frontier/by_industry), último acceso 24/07/2013

[17] IKV. The Medini QVT project. <http://projects.ikv.de/qvt>, último acceso 15/07/2013

[18] Object Management Group (OMG), “Meta object facility(MOF)2.0 query/view/transformation, V1.1”, <http://www.omg.org/spec/QVT/1.1> último acceso 8/07/2013

[19] Carlo Comin, Luka Onesti, and Carlos Kavka, “Towards a Standard Approach for Optimization in Science and Engineering”, Proceedings of the 8th International Conference on Software Engineering and Applications ICSOFT-EA, 2013, SciTePress.