

Propuesta de un framework para la comparación de diferentes lenguajes de modelado gráficos de procesos de negocio en términos de la representación de procesos inter-organizacionales

Diego Cocconi*, Marisa Pérez*, Juan Pablo Ferreyra*, Claudia Verino*,
Guido Melano*, Noelia Cocconi*, Andrea Biasco*

Departamento de Ingeniería en Sistemas de Información

Universidad Tecnológica Nacional (UTN) – Facultad Regional San Francisco

Av. de la Universidad 501 – San Francisco (2400) – Provincia de Córdoba – República
Argentina

*{dcocconi, mperez, jpferreyra, cverino, gmelano, ncocconi, abiasco}@sanfrancisco.utn.edu.ar

Resumen

Muchas organizaciones están aprovechando las ventajas de las tecnologías de Internet para colaborar entre ellas y participar de procesos de negocio inter-organizacionales o colaborativos. Desde hace varios años existe un estándar muy sólido para modelar procesos de negocio privados: BPMN (Business Process Model and Notation). Si bien desde la versión 2.0 cuenta con soporte para representar procesos de negocio colaborativos, al momento de iniciar determinado proyecto o implementación BPM (Business Process Management) en un ámbito inter-organizacional, es difícil tener la seguridad de que la documentación y los modelos que serán generados podrán representarse de acuerdo con las expectativas esperadas. Una vez comenzada la adopción de determinada notación o lenguaje, puede pasar un tiempo considerable desde el inicio del proyecto hasta percibir que no fue una buena decisión y tener que volver a modelar todo con un lenguaje diferente. En consecuencia, en el presente trabajo se realiza un análisis y comparación de las diferentes alternativas actuales para modelar procesos de negocio colaborativos, a fin de seleccionar la más apropiada para el modelado durante determinado proyecto. Entre las opciones disponibles actualmente para modelar procesos de negocio colaborativos, se consideraron las siguientes: (1) el lenguaje BPMN (versión 2.0); (2) las redes de Petri [9] o WorkFlow nets (WF-nets) [10], más específicamente; (3) los diagramas de actividad de UML (Unified Modelling Language) [11]; y (4) los diagramas de secuencia de UML. Sobre la base de dicho análisis, se propone un framework, cuya aplicación a proyectos concretos, partiendo de una especificación de sus principales características y/o requerimientos, permita obtener una recomendación acerca del lenguaje de modelado adecuado para llevarlos adelante.

1. Introducción

Muchas organizaciones están aprovechando las ventajas de las tecnologías de Internet para establecer *redes colaborativas* que les permitan participar de *procesos de negocio inter-organizacionales* o *colaborativos*. Una red colaborativa [1] consiste en un conjunto de organizaciones heterogéneas, autónomas y geográficamente distribuidas que deciden colaborar en conjunto para alcanzar objetivos comunes [2]. En este tipo de redes, la colaboración entre las organizaciones se materializa a través de la definición y ejecución de procesos de negocio colaborativos [3]. Un proceso de negocio colaborativo, que suele denominarse en la literatura *coreografía de procesos* [4] [5] o *Collaborative Business Process (CBP)*, según la caracterización descripta en [6], especifica la vista global de interacciones entre los roles que desempeñan las organizaciones para alcanzar sus objetivos comunes, sirviendo también como una base contractual para la colaboración establecida. Los procesos de negocio colaborativos se definen naturalmente como procesos abstractos, en el sentido en que no pueden ser directamente ejecutados por un motor de procesos centralizado [6]. En lugar de ello, la implementación y ejecución descentralizada de los procesos de negocio colaborativos requiere que cada organización implemente y ejecute sus propios *procesos privados* [4] de una manera sincronizada con el resto. Un proceso privado, o *proceso de orquestación* [5] según la literatura, referido también como *proceso de negocio de integración (IBP: Integration Business Process)* en [6], define las actividades públicas y privadas que cada organización debe llevar a cabo para posibilitar el intercambio de mensajes acordado en los procesos de negocio colaborativos.

Desde hace varios años existe un estándar muy sólido para modelar procesos de negocio privados: BPMN (Business Process Model and Notation), el cual cuenta con constructores con la semántica apropiada para modelar actividades, control de flujo y eventos propios de estos tipos

de procesos. En la versión actual (2.0) básicamente se pueden utilizar los *diagramas de orquestación* y de *colaboración* a tal fin [5]. Sin embargo, a la hora de realizar el modelado de procesos de negocio colaborativos surgen algunos cuestionamientos. Si bien desde la versión 2.0 BPMN cuenta con *diagramas de coreografía* para ello, al momento de iniciar determinado proyecto o implementación *BPM (Business Process Management)* en un ámbito inter-organizacional, es difícil tener la seguridad de que la documentación y los modelos que serán generados podrán representarse de acuerdo con las expectativas esperadas. Una vez comenzada la adopción de determinada notación o lenguaje, puede pasar un tiempo considerable desde el inicio del proyecto hasta percibir que no fue una buena decisión y tener que volver a modelar todo con un lenguaje diferente. Existen muchos trabajos que realizan comparaciones entre las diferentes alternativas para modelar procesos de negocio, como [7] [8] [9], que pueden tomarse como base para intentar responder interrogantes como las siguientes:

- ¿Permitirá el lenguaje/notación elegido describir procesos de negocio con la complejidad requerida de acuerdo con las situaciones que se deseen modelar?
- ¿Es posible representar los diferentes tipos de procesos de negocio de la misma forma?
- ¿Con cuál lenguaje/notación se sentirán más cómodos la mayoría de los involucrados en determinado proyecto, si no tuvieran conocimiento sobre procesos de negocio?

La respuesta a estas preguntas es decisiva a la hora de elegir el lenguaje apropiado para modelar y generar documentación estable en el tiempo durante implementaciones BPM reales, especialmente de procesos de negocio colaborativos. Lamentablemente, la mayoría de los trabajos al respecto, según el conocimiento de los presentes autores a la fecha, no se centran en analizar los procesos de negocio considerando una perspectiva inter-organizacional. Entre las opciones disponibles actualmente para modelar procesos de negocio colaborativos, se destacan cuatro que tienen gran influencia: (1) el lenguaje BPMN (versión 2.0); (2) las *redes de Petri* [10] o *WorkFlow nets (WF-nets)* [11], más específicamente; (3) los *diagramas de actividad* de UML (*Unified Modelling Language*) [12]; y (4) los *diagramas de secuencia* de UML.

En consecuencia, en el presente trabajo se realiza un análisis y comparación de las diferentes alternativas actuales para modelar procesos de negocio colaborativos y sus procesos de negocio privados implicados (si corresponde), a fin de seleccionar la más apropiada para el modelado durante determinado proyecto, logrando alcanzar las expectativas esperadas. De las diferentes alternativas mencionadas anteriormente, son destacados los distintos aspectos que las distinguen, así como sus ventajas y desventajas, mediante una revisión de la literatura relevante. Sobre la base de dicho análisis, se propone un *framework*, cuya aplicación a proyectos concretos, partiendo de una especificación de sus principales características y/o requerimientos, permita obtener una

recomendación acerca del lenguaje de modelado adecuado para llevarlos adelante.

El resto del artículo se organiza de la siguiente manera. Primero, en la Sección 2, se brinda un breve panorama sobre procesos de negocio colaborativos, así como una descripción de las diferentes alternativas de modelado. A continuación, en la Sección 3, se realiza la comparación de las mismas mediante una revisión de la literatura relevante, con el objetivo de obtener también criterios para poder caracterizar y evaluar cada una; el resultado de la comparación permite implementar el *framework*. La Sección 4 describe brevemente dos escenarios posibles de aplicación del *framework*. Finalmente, las Secciones 5 y 6 presentan las conclusiones y los próximos pasos a seguir, respectivamente.

2. Procesos de negocio colaborativos

Para llevar a cabo colaboraciones entre varias organizaciones, las mismas deben integrar sus diferentes procesos de negocio privados, acordar ciertos objetivos comunes, coordinar sus acciones e intercambiar información por medio de la definición y ejecución de procesos de negocio colaborativos [13]. La implementación de dichas colaboraciones también implica que las organizaciones sean capaces de aplicar las fases del *ciclo de vida BPM* considerando los procesos colaborativos involucrados [4] [14] [15].

Durante las etapas de análisis y diseño de dicho ciclo de vida, las organizaciones deben definir los procesos de negocio colaborativos, así como los procesos de negocio privados que le corresponden a cada una, los cuales van a modelar el comportamiento público y privado que van a necesitar para soportar las interacciones y roles a desempeñar en las colaboraciones. Cabe recordar que los procesos de negocio colaborativos no pueden ser ejecutados centralizadamente por el motor de procesos de un único *Sistema de Información Orientado a Procesos (SIOP)*, del inglés *PAIS: Process-Aware Information System* o *BPMS (Business Process Management System)*. En lugar de ello, deberán ser “ejecutados” por medio de la ejecución indirecta de los procesos de negocio privados de cada organización, ya sí por parte de sus respectivos SIOPs o BPMSs [6]. Es aquí donde cobra fundamental importancia utilizar un lenguaje de modelado apropiado, que permita la definición de modelos colaborativos (y privados) de una manera agnóstica, independiente de la tecnología de implementación, logrando un adecuado nivel de abstracción.

La fase de implementación implica el desarrollo, configuración y despliegue de los SIOPs requeridos por cada organización para ejecutar los procesos de negocio privados que darán soporte al proceso de negocio colaborativo. La ejecución de este último dependerá de las posibilidades con que cuenten las organizaciones de disponer de SIOPs capaces de interoperar entre ellos para

concretar el intercambio de mensajes acordado en el proceso de negocio colaborativo [6] [16].

Finalmente, la etapa de ejecución consiste en la instanciación de los procesos de negocio colaborativos por medio de la ejecución real de las instancias de los procesos de negocio privados de cada organización a través de sus respectivos SIOPs, para llevar a cabo las actividades tanto privadas como públicas requeridas, permitiendo el intercambio de mensajes entre ellos.

De lo descrito hasta el momento, se cuenta con dos diferentes perspectivas básicas para definir colaboraciones [16]: (1) considerando una vista global del flujo de control de las interacciones entre las organizaciones (*perspectiva global*) y (2) teniendo en cuenta una vista local del flujo de control de actividades tanto privadas como públicas de cada organización, con los puntos de interacción para comunicarse entre ellas (*perspectiva local*). La correspondencia entre ambas perspectivas, los tipos de procesos de negocio involucrados, y los lenguajes y sus representaciones aptas para cada una de ellas se muestran en la Figura 1.

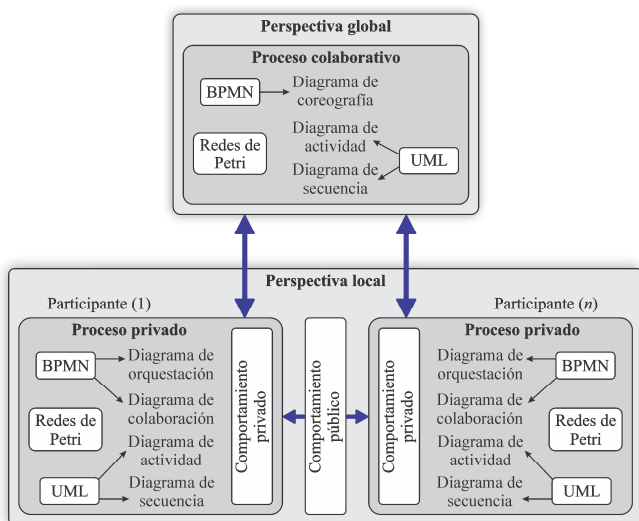


Figura 1: Perspectivas global y local involucradas en las colaboraciones entre organizaciones (adaptación de [13]).

La perspectiva global describe el comportamiento público de una colaboración y las responsabilidades de cada organización participante [3]. Dicho comportamiento se representa mediante procesos de negocio colaborativos, los cuales describen el flujo de control y en qué orden se van realizando los intercambios de mensajes entre los participantes. La perspectiva local describe el comportamiento de la colaboración desde el punto de vista de cada participante. Según esta perspectiva, la colaboración se define como un conjunto de procesos de negocio privados, uno para cada organización participante, y las interacciones que tienen lugar entre ellas. Cada participante define su propio flujo de control, mostrando hacia los demás los puntos de interacción (envío/recepción

de mensajes). Las organizaciones disponen principalmente de dos opciones a la hora de definir procesos de negocio según esta perspectiva: (1) *procesos de interfaz* (sólo comportamiento público) y (2) *procesos privados completos* (con comportamiento público y privado). Los procesos de negocio de interfaz sólo exponen aquellas actividades externamente visibles para el resto de los participantes, fundamentalmente las que tienen que ver con el envío/recepción de mensajes [3] [4] [17]. Un proceso de negocio privado completo (denominado *proceso de integración* en [6]) describe tanto el comportamiento público como el privado para concretar la colaboración. El comportamiento privado implica aquellas actividades y eventos necesarios para representar la lógica de negocio interna de cada participante.

En las siguientes Subsecciones se describen los lenguajes mencionados previamente en la Sección 1 y cómo permiten implementar cada una de las perspectivas anteriores.

2.1. BPMN

El lenguaje BPMN fue desarrollado bajo la coordinación de la *OMG (Object Management Group)* con la finalidad de convertirse en un estándar internacional, identificando las mejores prácticas y posibilidades para representar procesos de negocio (mediante grafos, redes de Petri, etc.), procurando un lenguaje que sea ampliamente adoptado. Este lenguaje pretende abarcar un amplio rango de abstracción, desde modelado del negocio hasta implementación de los procesos de negocio mediante alguna tecnología, ideal para ser comprendido por todos los involucrados en una implementación BPM [4].

BPMN define varios tipos de diagramas para representar las perspectivas global y local. La perspectiva global se puede caracterizar mediante diagramas de coreografía. La Figura 2.a muestra un ejemplo de diagrama de coreografía para una colaboración que consiste en acordar la reposición de determinada línea de productos en un supermercado por parte de un proveedor. El proceso de negocio colaborativo inicia con el supermercado solicitando el pedido de reposición al proveedor. Luego el control se le pasa al proveedor, quien decide si aceptar o rechazar la solicitud de reposición. En ambos casos actúa en consecuencia: (1) si acepta realizar la reposición, le confirma el pedido al supermercado, con una fecha estimada de entrega, por ejemplo; (2) si rechaza realizar la reposición, informa el motivo del rechazo al supermercado (por ejemplo, que ya no trabaja con esa línea de productos). En un diagrama de coreografía, las interacciones entre participantes se representan explícitamente mediante *tareas de coreografía*. El flujo de control y los *gateways* determinan la secuencia de las diferentes tareas [5].

La perspectiva local también se puede representar sin problemas en BPMN. Para ello, se dispone de los diagramas de orquestación y colaboración. Los diagramas de orquestación proporcionan una vista detallada de las actividades del proceso de negocio y sus relaciones, el cual es llevado a cabo en el entorno de una sola organización [4].

Pueden usarse para definir los procesos en los dos niveles de esta perspectiva. Si la intención es realizar un proceso de interfaz, el diagrama sólo debe incluir actividades y eventos públicos, como aquellos referidos al envío/recepción de mensajes. Por el contrario, si la intención es modelar un proceso de negocio privado completo, deben definirse todas las actividades y eventos involucrados para llevar a cabo el mismo. Un diagrama aún más útil para esta perspectiva que proporciona BPMN es el diagrama de colaboración. Este tipo de diagrama (Figura 2.b) generalmente tiene dos o más *pools*, uno para cada participante de la colaboración. El intercambio de mensajes entre ellos se muestra explícitamente como un flujo de mensajes que conectan los *pools* [5]. Aquí se presentan varias posibilidades para representar diferentes niveles de abstracción [4], incluso para cada participante independientemente; lo que no se puede dejar de representar son los mensajes que se intercambian [5]. Básicamente, las opciones incluyen: (1) un *pool* que se puede dejar en blanco, si no se conoce nada acerca del participante; (2) un *pool* que puede representar en su interior un proceso de interfaz, mostrando solamente la “cara” pública del participante (el proveedor en el diagrama de colaboración de la Figura 2.b); y (3) un *pool* que puede mostrar todo el proceso de negocio completo del participante (el supermercado en la Figura 2.b).

2.2. Redes de Petri

Las redes de Petri constituyen una opción de modelado importante, ya que han demostrado su aplicación para representar y analizar todo tipo de procesos, desde protocolos, hardware, sistemas embebidos, hasta sistemas de manufactura, logística, y procesos de negocio. Las redes de Petri clásicas se construyen mediante un grafo dirigido con dos tipos de nodos: *places* (representados por círculos) y *transitions* (representadas por rectángulos). La red permite el movimiento de *tokens* (representados por puntos negros) entre los diferentes *places*, los cuales permiten definir el estado actual de la red. En realidad, son las *transitions* y sus reglas las que permiten a los *tokens* moverse entre *places*, precisamente mediante el “disparo” de tales *transitions* [10].

A diferencia de BPMN, las redes de Petri no fueron concebidas para representar naturalmente las perspectivas global y local. Para la perspectiva local y el modelado involucrando una sola organización, no hay mayores inconvenientes. Sin embargo, cuando entra en juego más de un participante, no son directamente aplicables. En [10] se propone la definición de un *workflow* inter-organizacional que utiliza *communication places* para representar elementos de comunicación asincrónicos correspondientes a intercambios de mensajes entre dos o más redes, una para cada participante. Con esta consideración, es posible representar todas las perspectivas.

En la Figura 3.a se muestra la perspectiva global para la colaboración entre el supermercado y el proveedor de la Figura 2, la cual se representa por medio de dos redes de Petri que solamente cuentan con *places* y *transitions* para permitir el intercambio de mensajes mediante los *communication places*. Ningún comportamiento privado se muestra en las dos redes.

La perspectiva local también se puede representar sin problemas mediante la propuesta de [10], en los dos niveles considerados. Si la intención es realizar un proceso de interfaz, la red sólo debe incluir *places* y *transitions* que den soporte para que los *communication places* puedan trabajar, exactamente del mismo modo en que se representó para la perspectiva global (el proveedor en la colaboración de la Figura 3.b). Por otro lado, si la intención es proveer un proceso de negocio privado completo, deben definirse los *places* y *transitions* que den soporte a todas las tareas y eventos involucrados para llevar a cabo el proceso de negocio (el supermercado en la Figura 3.b).

Además de las redes de Petri propiamente dichas, en el mundo de los procesos de negocio se utilizan ampliamente las *Workflow nets (WF-nets)*. En una *WF-net*, las actividades o tareas se modelan a través de *transitions* y sus dependencias casuales, mediante *places*. Además, se han definido varios constructores útiles para representar patrones de *workflows* de una manera bastante compacta: *AND-split/AND-join* y *OR-split/OR-join*. Una restricción importante que agrega este tipo de redes es que no puede haber más de un *input place* y un *output place*, porque la idea es que cuando el *workflow* inicia mediante un sistema de gestión de *workflows* (el cual le entrega un *token* al *input*

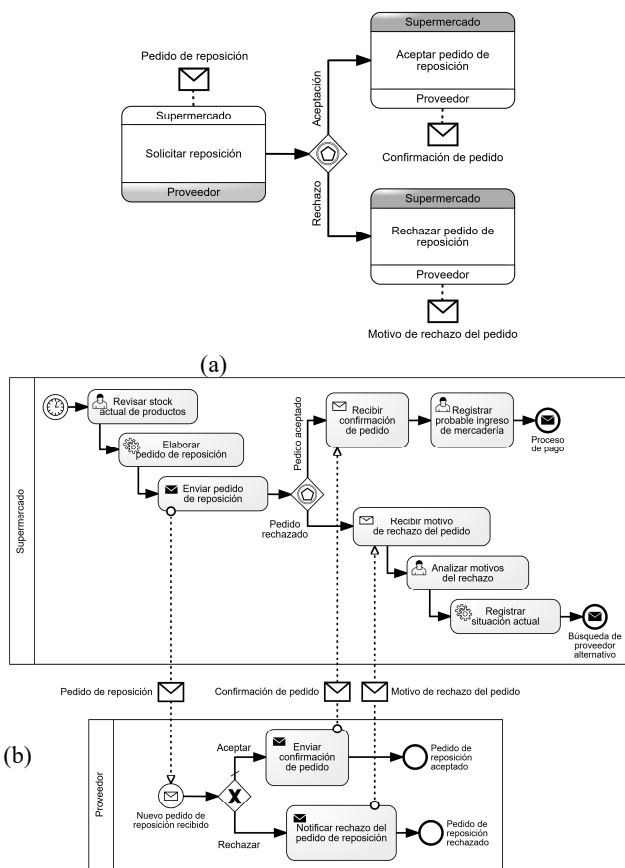


Figura 2: Colaboración de ejemplo para acordar la reposición de determinada línea de productos en un supermercado por parte de un proveedor (en BPMN). (a) Perspectiva global representada mediante un diagrama de coreografía. (b) Perspectiva local representada mediante un diagrama de colaboración.

place), debe ser completamente terminado al acabar su ejecución, no quedando ningún *token* circulando en la estructura (el sistema espera que el único *token* que quede termine llegando al *output place*) [4] [10].

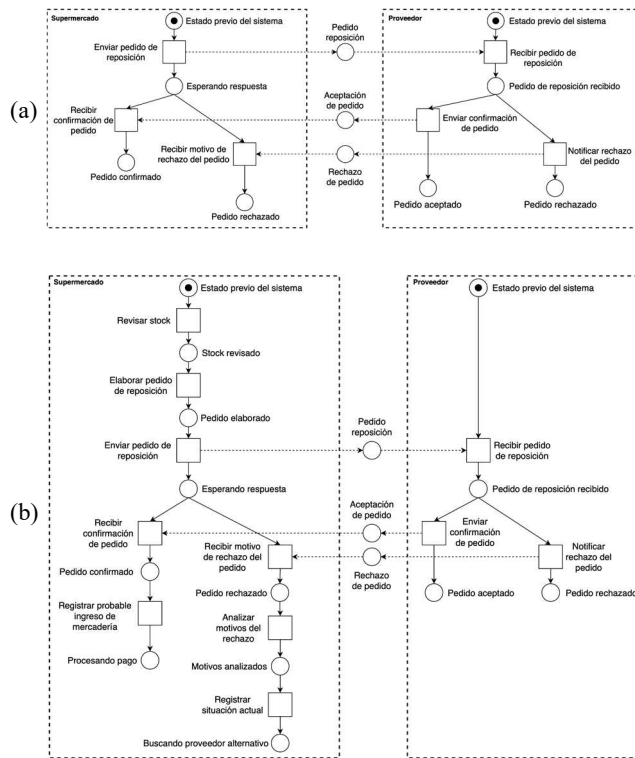


Figura 3: Colaboración de ejemplo de la Figura 2, en términos de redes de Petri, según la propuesta de [10]. (a) Perspectiva global. (b) Perspectiva local.

Adicionalmente, en [18] los autores describen un método para modelar *workflows* inter-organizacionales, en el que se propone partir de una red de Petri completa en la que intervienen todas las organizaciones participantes, hasta llegar a una partición para determinar los *workflows* privados de cada una. En ese mismo trabajo, proponen también una notación propia para el intercambio de mensajes entre bloques que representan las particiones, ocultando su comportamiento privado.

2.3. UML

UML es un lenguaje de modelado visual para sistemas de software, que se diseñó para incorporar las mejores prácticas de las técnicas de modelado e ingeniería de software existentes. La premisa básica de UML consiste en poder modelar software y otros sistemas como colecciones de objetos que interactúan, lo cual encaja perfectamente en el desarrollo de software utilizando lenguajes de programación orientados a objetos, pero también funciona muy bien para procesos de negocio y otras aplicaciones. UML permite representar dos aspectos fundamentales de los sistemas: (1) su estructura estática (los objetos que son importantes en el modelo y cómo se relacionan) y (2) su comportamiento dinámico (los ciclos de vida de los objetos

y cómo interactúan entre sí para entregar la funcionalidad requerida del sistema). Uno de los bloques constitutivos importantes de UML son los diversos diagramas para representar esos dos aspectos. Dichos diagramas conforman las diferentes vistas del modelo. Entre los trece diagramas que dispone la versión 2 de UML, uno de ellos se ha utilizado ampliamente para representar procesos de negocio: los *diagramas de actividad*. Sin embargo, en el ámbito colaborativo cobra importancia otro diagrama ampliamente conocido: el *diagrama de secuencia* [19].

2.3.1. Diagramas de actividad

Los diagramas de actividad proporcionan un mecanismo de carácter general para modelar comportamiento y pueden emplearse en diversas etapas del proceso de desarrollo de software [19], pero además, se pueden usar en el modelado organizacional para la ingeniería de procesos de negocio y *workflows* [19] [20]. Los diagramas de actividad especifican una secuencia de unidades subordinadas (*actividades, eventos de señal*), utilizando un modelo de control y flujo de datos. Los comportamientos subordinados pueden iniciarse porque otros comportamientos en el modelo terminan de ejecutarse, porque los objetos y datos están disponibles o porque los eventos ocurren externamente al flujo. Un diagrama de actividad se especifica como un gráfico de nodos interconectados por flechas [20].

Estos diagramas permiten la representación de la perspectiva global mediante el uso de *calles*, una para cada participante, los cuales pueden enviarse mensajes como objetos de datos intercambiados (Figura 4.a) a través de eventos de señal. Para la perspectiva local, puede usarse el mismo criterio en caso de tratarse de procesos de interfaz (el proveedor en la Figura 4.b). Si se necesita representar un proceso de negocio completo, el participante que lo desee puede agregar todas las actividades privadas que considere (el supermercado en la Figura 4.b).

En [21] los autores toman este tipo de diagrama como punto de partida para definir un perfil UML que les permite representar las perspectivas global y local de manera más cercana al dominio de los procesos de negocio. En dicho trabajo también utilizan un método para generar modelos ejecutables a partir de las coreografías y orquestaciones definidas mediante ese perfil.

2.3.2. Diagramas de secuencia

Los diagramas de secuencia también permiten representar comportamiento, pero enfatizando la secuencia de envío de mensajes ordenada en el tiempo entre denominadas *líneas de vida*, que corresponden a los objetos participantes de una interacción [19]. Suelen emplearse mucho en la literatura para mostrar interacciones entre actores, interfaces, componentes de sistemas, clases, etc., por lo que por naturaleza se adecuan para representar colaboraciones.

Desde el punto de vista de la perspectiva global, cada participante se puede mostrar en el diagrama como una línea de vida que interactúa con los demás mediante el envío de

mensajes asincrónicos (Figura 5.a). Para representar la perspectiva local, en lo que respecta a los procesos de interfaz, el diagrama no cambia (el proveedor en la Figura 5.b). Sin embargo, para representar procesos de negocio completos, es necesario exhibir actividades privadas. Esto último se puede lograr de dos maneras: (1) mediante una *autodelegación*, es decir, el envío de un mensaje de una línea de vida a sí misma, indicando la llamada a determinada actividad privada; o (2) por medio del envío de mensajes a otras líneas de vida visibles solamente por el participante, encargadas de llevar a cabo las actividades (como la llamada al sistema de información por parte del supermercado en la Figura 5.b).

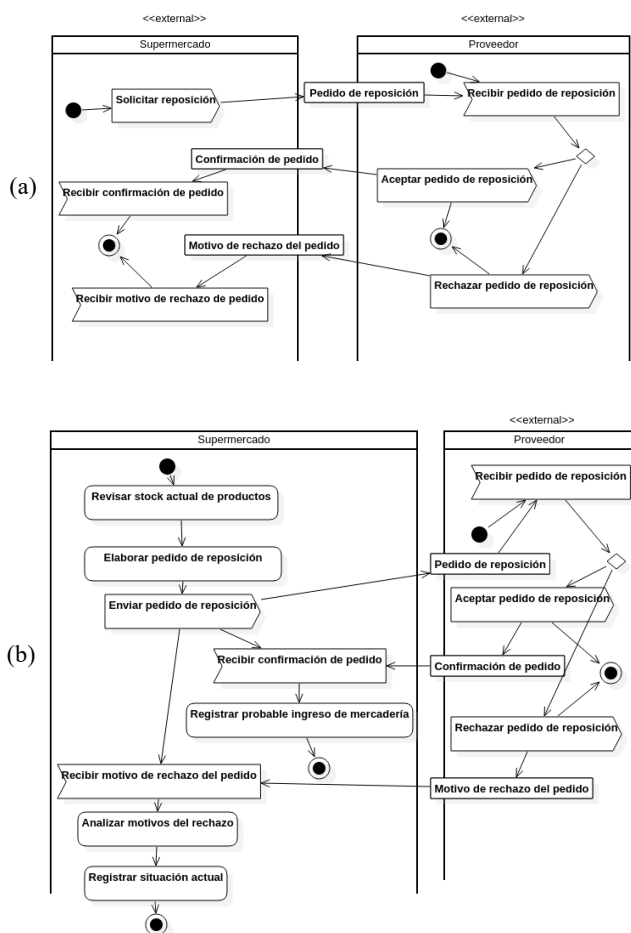


Figura 4: Colaboración de ejemplo de la Figura 2, en términos de diagramas de actividad UML. (a) Perspectiva global. (b) Perspectiva local.

Tal como ocurre con los diagramas de actividad, la semántica de los diagramas de secuencia se puede ampliar mediante la definición de perfiles UML. Por ejemplo, los autores de [22] proponen un método basado en *MDA (Model Driven Architecture)* para generar modelos de procesos de negocio ejecutables a partir de la definición de modelos de procesos colaborativos, independientes de una implementación tecnológica específica. Para ello, utilizan un lenguaje denominado *UP-ColBPIP*, el cual define varias

vistas, entre las cuales se encuentra una llamada *interaction protocols*, basada en la especialización de diagramas de secuencia para representar colaboraciones.

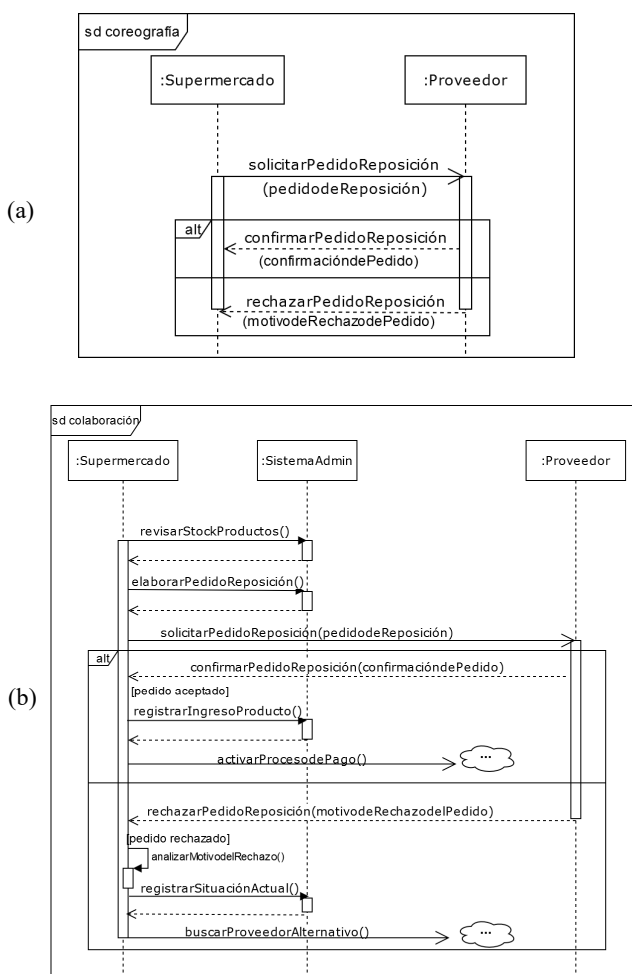


Figura 5: Colaboración de ejemplo de la Figura 2, en términos de diagramas de secuencia UML. (a) Perspectiva global. (b) Perspectiva local.

3. Comparación de diferentes lenguajes de modelado para representar colaboraciones

Considerando las diferentes alternativas actuales para modelar procesos de negocio colaborativos y sus procesos de negocio privados implicados (si corresponde), a fin de seleccionar la más apropiada para el modelado durante determinado proyecto/implementación BPM, se propone un *framework* basado en el análisis y comparación de las mismas, teniendo en cuenta los distintos aspectos que las distinguen, así como sus ventajas y desventajas. De este modo, el *framework* permite contar con una referencia acerca de cuál escoger a la hora de iniciar un proyecto concreto. Para ello, primero se ha realizado una revisión de la literatura relevante sobre modelado de procesos de negocio colaborativos, con el objetivo de obtener criterios

para poder caracterizar y evaluar cada lenguaje, luego se realizó la comparación con los resultados de dicha revisión y la experiencia acumulada acerca del tema en implementaciones previas (como las descritas en [16], [15] y [23]), y con los resultados obtenidos, se procede a la implementación del *framework*.

3.1. Revisión de la literatura

La revisión de la literatura se realizó de acuerdo con el método denominado “*snowballing*” por [24]. Este método sugiere comenzar con un grupo inicial de trabajos manualmente identificados de acuerdo con cierto criterio. Luego, se agregan más trabajos hacia atrás y hacia adelante en el tiempo, utilizando las referencias de los trabajos relacionados (*backward snowballing*) y las citas que se mencionen en las librerías electrónicas (*forward snowballing*). La búsqueda manual se realizó en librerías electrónicas relevantes (*Springer*¹, *Elsevier*², *IEEE Xplore*³) y en libros acerca de los que se tiene conocimiento, primero buscando criterios para evaluar lenguajes de modelado de procesos de negocio y luego, en una segunda pasada, con varios criterios en mente, se refinó la búsqueda en términos de: “lenguaje + procesos colaborativos + criterio”. Luego se procedió a realizar la segunda etapa del método (*backward snowballing* y *forward snowballing*).

Acerca de los criterios, de acuerdo con los autores de [25], se han utilizado muchos para evaluar lenguajes de modelado de procesos de negocio (en términos de la representación de procesos privados), algunos con más peso que otros en la literatura. Varios de estos criterios, que los autores han logrado unificar, constituyen un importante punto de partida para extrapolarlos al ámbito colaborativo. Entre ellos, se rescatan los siguientes:

- **Expresividad:** capacidad del lenguaje para representar múltiples situaciones organizacionales, en términos de perspectivas funcionales, estructurales y de datos requeridas por BPM.
- **Facilidad de entendimiento:** grado de facilidad/dificultad de los involucrados o *stakeholders* para entender e interpretar los modelos generados. Según este criterio, influye el diseño de los constructores del lenguaje a nivel gráfico, en el sentido de resultar o no intuitivos (*user-friendly*).
- **Formalidad:** rigor de la semántica, en su carácter de permitir o no ambigüedades a la hora de interpretar (o incluso ejecutar) los modelos.
- **Versatilidad:** grado de aplicabilidad del lenguaje para representar modelos a lo largo de todo el ciclo de vida BPM (análisis, documentación, simulación, ejecución, etc.).
- **Soporte de herramientas:** grado de mayor/menor disponibilidad de herramientas para realizar el modelado utilizando el lenguaje.
- **Concisión:** grado de mayor/menor posibilidad del lenguaje para expresar diferentes situaciones

utilizando la menor cantidad posible de constructores.

- **Facilidad de aprendizaje:** grado de mayor/menor esfuerzo requerido por los involucrados o *stakeholders* para dominar el lenguaje.

Por su parte, los autores de [26] hacen énfasis en la evaluación de diferentes herramientas de modelado BPMN, lo cual permite agregar el siguiente criterio, relacionado con el “soporte de herramientas” de [25], pero considerando licenciamiento gratuito:

- **Disponibilidad de herramientas gratuitas:** grado de mayor/menor disponibilidad de herramientas gratuitas/*open-source* para realizar el modelado utilizando el lenguaje.

En [27] también se realiza un análisis de los lenguajes de modelado para dar soporte a colaboraciones, haciendo énfasis en diferentes niveles de abstracción. Gran parte de los criterios que utilizan ya fueron descritos anteriormente, pero se incorpora el siguiente, relacionado con la “versatilidad” de [25]:

- **Ejecución:** grado de mayor/menor capacidad de obtener modelos que sean ejecutables por un SIOP o BPMS directamente.

Finalmente, sobre la base de la experiencia de los autores del presente trabajo ([16] [15] [23]), se cree conveniente tener en cuenta tres aspectos adicionales más:

- **Extensibilidad:** grado de facilidad/complejidad para extender el lenguaje por medio de mecanismos previstos en situaciones especiales.
- **Dominio de aplicación:** grado en que el lenguaje se relaciona con el ámbito del dominio del proyecto/implementación BPM.
- **Tipos de diagramas requeridos:** posibilidad de poder expresar mediante el lenguaje las diferentes perspectivas (global y local) haciendo uso de la menor cantidad posible de tipos de diagramas.

Sobre la base de estos criterios se han clasificado los trabajos restantes producto de la revisión de la literatura, cuyas consideraciones y/u observaciones son tomadas en cuenta durante la evaluación de los lenguajes en la Subsección siguiente. La clasificación de dichos trabajos se muestra en la Tabla 1, indicando la referencia que los autores hacen respecto de los criterios definidos anteriormente.

3.2. Análisis comparativo

Con los criterios de evaluación definidos y los trabajos clasificados según los mismos, se realiza el análisis comparativo de los lenguajes de modelado. Para ello, se intenta responder a la pregunta “¿cómo se valora lo más objetivamente posible el lenguaje *X* respecto del criterio *Y*”

¹ <https://www.springer.com/la>

² <https://www.elsevier.es/es>

³ <https://ieeexplore.ieee.org/Xplore/home.jsp>

para representar procesos de negocio colaborativos (considerando ambas perspectivas, la global y la local)?”.

Tabla 1. Clasificación de los trabajos producto de la revisión de la literatura según los criterios para analizar y evaluar lenguajes de modelado para representar colaboraciones.

Lenguajes/ Criterios	BPMN	Redes de Petri	UML	
			Diag. de actividad	Diag. de secuencia
<i>Expresividad</i>	[28] [29]	[10] [30] [4]	[20]	[31]
<i>Formalidad</i>	[32] [33]	[10] [4]	[19]	[31]
<i>Concisión</i>	[25]	[33]		
<i>Versatilidad</i>	[34] [35]		[19]	[19]
<i>Ejecución</i>	[16]	[4]	[21]	[31]
<i>Facilidad de entendimiento</i>	[28] [36] [37]	[4]	[38]	[19]
<i>Facilidad de aprendizaje</i>	[39]		[38]	
<i>Soporte de herramientas</i>	[40] [41]			
<i>Disponibilidad de herramientas gratuitas</i>				
<i>Extensibilidad</i>	[42]	[18]	[21]	[22]
<i>Dominio de aplicación</i>	[4] [43]	[4] [43]	[43]	
<i>Tipos de diagramas requeridos</i>	[4]	[4]	[21]	[19] [22]

Como en [25], se utiliza una escala del 0 al 5 (el lenguaje no cumple con el criterio o lo hace de manera satisfactoria, respectivamente) para evaluar los lenguajes. Seguidamente, se resume el análisis/evaluación respecto de cada criterio:

- **Expresividad:** los lenguajes que presentan la mayor expresividad para representar colaboraciones son BPMN y UML (diagramas de actividad), considerando las principales perspectivas de los procesos de negocio según BPM y las perspectivas global y local de las colaboraciones. Las redes de Petri también son expresivas, pero debe aplicarse la propuesta descrita en [10], por ejemplo, para poder representar procesos de negocio colaborativos. Finalmente, los diagramas de secuencia de UML son los menos expresivos, pues si bien resultan naturales para las colaboraciones, no permiten representar explícitamente actividades, sino llamadas o intercambios de mensajes; por lo tanto, es difícil de expresar la perspectiva local. En [22] se especializan solamente para la perspectiva global.
- **Formalidad:** este es el punto fuerte de las redes de Petri; los demás lenguajes presentan ambigüedades subsanables en mayor o menor medida. BPMN y los diagramas de actividad UML tienen un comportamiento basado en redes de *tokens*, pero carecen de la formalidad matemática propia de las redes de Petri. Es más, muchos autores han

representado sus semánticas en términos de redes de Petri (como en [33]), con la finalidad de reducir la brecha entre los modelos y sus implementaciones ejecutables.

- **Concisión:** todos los lenguajes de modelado evaluados están pensados para ser utilizados en diferentes contextos. BPMN es el que cuenta con más constructores específicos para modelar diferentes situaciones propias de los procesos de negocio, incluyendo los colaborativos, por lo que permite representar de forma clara y concisa diferentes situaciones. Las redes de Petri, por su parte, son de aplicación muy general, y como cuentan con muy pocos constructores básicos (*places* y *transitions*), generalmente se requieren varios de ellos para representar patrones que se suelen simbolizar mediante un solo constructor en BPMN. Finalmente, UML está pensado para modelar en el ámbito de la ingeniería de software, por lo que carece de constructores específicos para procesos de negocio.
- **Versatilidad:** en este sentido, los lenguajes que expresamente definen diagramas (BPMN y UML) presentan una elevada versatilidad a la hora de utilizar los propios modelos como artefactos de análisis, diseño, documentación y comunicación, incluso de ejecución. Por otro lado, las redes de Petri, cuyo énfasis es la definición formal del modelo, puede prescindir incluso de representación gráfica, o la misma puede presentar una elevada complejidad, lo cual la hace impráctica para una documentación de alto nivel, por ejemplo.
- **Ejecución:** aquí la formalidad juega un papel importante, porque si el modelo está libre de ambigüedades, puede ser directamente ejecutado o simulado. Los lenguajes evaluados, con excepción de las redes de Petri, no son estrictamente formales, lo cual genera inconvenientes a la hora de realizar la implementación de un proceso de negocio ejecutable. Por lo tanto, suelen emplearse lenguajes de especificación en su lugar, por ejemplo, *WS-BPEL*⁴ (*Web Services Business Process Execution Language*), *WS-CDL*⁵ (*Web Services Choreography Description Language*) y *XPDL*⁶ (*XML Process Definition Language*). En la literatura existen muchos ejemplos de transformaciones o generaciones de modelos a partir de representaciones en lenguajes de modelado como BPMN [21] o UML [3].
- **Facilidad de entendimiento:** en este punto existen disparidades, incluso considerando un mismo lenguaje. Los diagramas de actividad UML destacan por ser altamente intuitivos, debido a su remembranza con diagramas de flujo tradicionales. BPMN, por su parte, presenta dos inconvenientes: (1) una elevada disponibilidad/configuración de constructores, algunos de ellos no tan intuitivos; y

⁴ <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

⁵ <https://www.w3.org/TR/ws-cdl-10/>

⁶ <https://wfmc.org/standards/xpdl>

(2) agrega un diagrama que se aleja bastante de la clásica remembranza del diagrama de flujo (el diagrama de coreografía). Las redes de Petri requieren un sustento teórico importante por detrás para representar correctamente los modelos. Finalmente, los diagramas de secuencia UML resultan poco intuitivos, pues están orientados al mundo del software; mediante la especificidad de su semántica, como en [22], se pueden volver más intuitivos para las colaboraciones.

- **Facilidad de aprendizaje:** el aspecto anterior y el dominio de aplicación influyen ampliamente sobre este criterio. Para dominar con propiedad el modelado en redes de Petri se requieren conocimientos acerca de sus fundamentos matemáticos, relacionados con su aspecto altamente formal. BPMN es intuitivo en lo que respecta a la perspectiva local (aunque cuenta con muchos constructores y configuraciones por aprender), pero las coreografías implican dificultades adicionales, ya que hay que tener ciertas consideraciones acerca de cuándo un flujo es correcto. Los diagramas de actividad UML son muy intuitivos, y debido al acotado número de constructores con los que cuentan, pueden conseguirse avances muy rápidamente al modelar. Los diagramas de secuencia UML, por su parte, son equiparables a los diagramas de coreografía BPMN, aunque las interacciones se pueden percibir con mayor naturalidad en el tiempo; igualmente, es necesario contar con cierto conocimiento sobre los aspectos estructural y de comportamiento de UML para comprenderlos adecuadamente.
- **Soporte de herramientas:** todos los lenguajes cuentan con soporte para diversas herramientas software. Particularmente, para BPMN es amplia la oferta de suites BPMS (*Bizagi*⁷, *Signavio*⁸, *IBM Business Process Manager*⁹, *Oracle Business Process Management Suite*¹⁰, *Camunda BPM*¹¹), así como también existen modeladores integrados a IDE (*Integrated Development Environments*) multilenguaje (*Eclipse*¹²); sin embargo, para coreografías, la oferta es mucho más acotada. UML cuenta con numerosos IDE multilenguaje (*Eclipse*, *MS Visual Studio*¹³), herramientas de modelado específicas (*StarUML*¹⁴, *Modelio*¹⁵) e incluso software de uso general que le dan soporte (*diagrams.net*, *MS Visio*¹⁶). Las redes de Petri también cuentan con una numerosa oferta de herramientas, principalmente de carácter académico, siendo posible emplear software de diseño de propósito general para modelar.
- **Disponibilidad de herramientas gratuitas:** el soporte de herramientas software por sí sólo no suele ser un indicador de peso en algunas situaciones. Si las organizaciones no están dispuestas a incurrir en los costos de una suite BPMS o determinado IDE de modelado, este aspecto cobra importancia y puede cambiar el rumbo del proyecto. BPMN cuenta con varios proyectos de modeladores gratuitos, pero con soporte para coreografías, la oferta es muy reducida (por ejemplo, *chor-js*¹⁷ o *Signavio*, pero este último no está disponible para el público en general, sino como una iniciativa educativa). Las redes de Petri disponen de muchas herramientas académicas o de diseño de propósito general, como se comentó anteriormente. UML, por su parte, presenta abundante oferta de herramientas gratuitas, incluso integrables a IDE (como *Eclipse*).
- **Extensibilidad:** los mecanismos de extensibilidad varían ampliamente para los lenguajes considerados. BPMN cuenta con un mecanismo que pone a disposición cuatro meta-modelos a tal fin, pero son pocas las extensiones que cumplen estrictamente el estándar (la mayoría son *ad hoc*) [42], por simplicidad. En comparación, la semántica de los elementos estándar de UML se puede especializar mediante un *perfil*. Por su parte, las redes de Petri son extensibles *per se*, ya que, para agregar alguna característica, deben realizarse nuevas restricciones en términos matemáticos/ formales.
- **Dominio de aplicación:** el dominio de aplicación natural de los tres lenguajes también difiere considerablemente. El más específico para el ámbito de los procesos de negocio y que involucra conceptos relacionados con profesionales de esa área es BPMN. Las redes de Petri son lo suficientemente generales, pero tienen una carga matemática muy elevada. UML, por su parte, está pensado para profesionales de la ingeniería de software y los sistemas de información.
- **Tipos de diagramas requeridos:** la variedad de representaciones gráficas que deben conocerse para modelar los procesos de negocio según las perspectivas global y local dependen del lenguaje elegido. Con BPMN se puede usar un único diagrama para representar colaboraciones (diagrama de colaboración), aunque dispone de uno exclusivo para procesos de negocio colaborativos y su perspectiva global (diagrama de coreografía). Las redes de Petri utilizan siempre una representación mediante grafos y lo que puede cambiar es el agregado de algún nuevo constructor con alguna semántica definida formalmente (involucrando

⁷ <https://www.bizagi.com/?lang=es>

⁸ <https://www.signavio.com/>

⁹ <https://www.ibm.com/products/process-management-software-in-the-cloud>

¹⁰ <https://www.oracle.com/middleware/technologies/bpm.html>

¹¹ <https://camunda.com/>

¹² <https://www.eclipse.org/bpmn2-modeler/>

¹³ <https://visualstudio.microsoft.com/es/>

¹⁴ <http://staruml.io/>

¹⁵ <https://www.modelio.org/>

¹⁶ <https://www.microsoft.com/es-es/microsoft-365/visio/flowchart-software>

¹⁷ <https://github.com/bptlab/chor-js>

places y/o transitions). Los diagramas de actividad y de secuencia UML no cambian para representar las diferentes perspectivas.

Sobre la base de este análisis, se conforma una tabla de resultados (Tabla 2), que sirve para la implementación del *framework*.

Tabla 2. Resultados de la evaluación de los lenguajes de modelado para representar colaboraciones según los criterios definidos.

Lenguajes/ Criterios	BPMN	Redes de Petri	UML	
			Diag. de actividad	Diag. de secuencia
<i>Expresividad</i>	5	3	4	2
<i>Formalidad</i>	4	5	4	3
<i>Concisión</i>	5	2	3	3
<i>Versatilidad</i>	5	3	4	4
<i>Ejecución</i>	4	5	3	3
<i>Facilidad de entendimiento</i>	3	2	5	3
<i>Facilidad de aprendizaje</i>	3	2	5	4
<i>Soporte de herramientas</i>	4	4	5	5
<i>Disponibilidad de herramientas gratuitas</i>	3	4	5	5
<i>Extensibilidad</i>	3	2	5	5
<i>Dominio de aplicación</i>	5	3	4	3
<i>Tipos de diagramas requeridos</i>	3	5	4	4

3.3. Conclusiones de la evaluación

Los resultados de la Tabla 2 muestran que ningún lenguaje es predominante según todos los criterios simultáneamente. Dependiendo del grupo de criterios al que se le asigne mayor relevancia, habrá un lenguaje que se destacará de los demás, por las ventajas que ofrezca. Si dicha tabla se utiliza como base para el desarrollo de un *framework*, a partir de la identificación del grupo de criterios que requiere o que sean importantes de acuerdo con las características de un nuevo proyecto/implementación BPM implicando procesos colaborativos (considerando todos los involucrados/*stakeholders*), sería posible aproximar el lenguaje más apropiado, por las ventajas que aportaría.

3.4. Implementación del *framework*

Actualmente, se está trabajando en la implementación del *framework* mediante un prototipo, el cual se encuentra en etapa de definición de requerimientos. Entre los más importantes, se destacan: (1) poder actualizar los valores de la Tabla 2 en la medida que se amplíe la revisión de la literatura o surjan nuevos trabajos al respecto; (2) poder agregar nuevos criterios/lenguajes; (3) poder elegir las características de los proyectos, asignando diferentes pesos según los criterios importantes; (4) permitir recomendar la mejor opción de lenguaje para los proyectos; (5) realizar el

desarrollo como una plataforma Web debido a la disponibilidad de recursos para ello.

4. Potenciales casos de estudio

Para probar la utilidad del *framework*, se cuenta con dos situaciones que requieren lidiar con la selección de un lenguaje apropiado para modelar colaboraciones.

En el primero de los casos, se está trabajando junto a dos organizaciones interesadas en optimizar sus operaciones mediante un modelo colaborativo, ya que presentan un elevado grado de interacción entre ellas. En este contexto, la empresa *Melkon S.A.* se compone de dos unidades de negocio (organizaciones): *Nitron* y *Métron* (los nombres han sido cambiados y sus procesos de negocio serán descritos con el suficiente nivel de abstracción, a modo de preservar sus identidades y la privacidad de sus procesos internos). La primera produce maquinaria agrícola y la segunda, partes y componentes para dichas maquinarias. Entre ambas unidades de negocio existe una fuerte relación que afecta principalmente las áreas de compras, calidad y producción. Para ejemplificar, cuando *Métron* planifica la producción de maquinarias agrícolas involucra a personal de producción de *Nitron*, quienes deben coordinar y adaptar sus planes de producción para cumplir con la planificación de *Métron*. A su vez, *Métron* se compone de varias plantas productivas, ubicadas físicamente en diferentes lugares. Esto plantea un flujo permanente de físicos, tanto de materias primas como de piezas, entre las distintas plantas y con proveedores de servicios de mecanizado denominados “procesos de terceros”.

El segundo caso corresponde al *Centro Auditor* dependiente de la Secretaría de Extensión Universitaria del Rectorado de la Universidad Tecnológica Nacional. La tarea fundamental del mismo es validar ante la Secretaría de Energía de la Nación las auditorías que realicen auditores presentes en diferentes Facultades, las cuales proporcionan este servicio a operadores de estaciones de servicio de combustibles, almacenaje de combustibles, etc. Por lo tanto, hay una interacción biunívoca entre el *Centro Auditor* y las diferentes Facultades, que puede ser abordada mediante un modelo apropiado de colaboración.

5. Conclusiones

Al momento de iniciar determinado proyecto o implementación BPM en un ámbito inter-organizacional, es difícil tener la seguridad de que la documentación y los modelos que serán generados podrán representarse de acuerdo con las expectativas esperadas. Una vez comenzada la adopción de determinada notación o lenguaje, puede pasar un tiempo considerable desde el inicio del proyecto hasta percibir que no fue una buena decisión y tener que volver a modelar todo con un lenguaje diferente.

Por lo tanto, en el presente trabajo se ha descrito un análisis y comparación de diferentes alternativas actuales

para modelar procesos de negocio colaborativos, a fin de seleccionar la más apropiada para el modelado durante determinado proyecto. Se han considerado las siguientes opciones: (1) el lenguaje BPMN (versión 2.0); (2) las redes de Petri o *WorkFlow nets (WF-nets)*; (3) los diagramas de actividad de UML; y (4) los diagramas de secuencia de UML. Producto de dicho análisis se ha concluido que dependiendo del grupo de criterios al que se le asigne mayor relevancia en el proyecto, habrá un lenguaje que se destacará de los demás, por las ventajas que ofrezca.

Finalmente, sobre la base del análisis, se propuso un *framework*, cuya aplicación a proyectos concretos, partiendo de una especificación de sus principales características y/o requerimientos, permita obtener una recomendación acerca del lenguaje de modelado adecuado para llevarlos adelante por parte de todos los interesados/*stakeholders*.

6. Trabajos futuros

El trabajo futuro se centrará en perfeccionar los resultados del análisis y comparación de las diferentes alternativas para modelar procesos colaborativos, ampliando la revisión de la literatura y teniendo en cuenta si es conveniente agregar criterios adicionales.

También se terminará la implementación del *framework* para poder aplicarlo a dos casos de estudio actuales, los cuales involucran procesos de negocio colaborativos. En uno de ellos los interesados no conocen nada acerca de procesos de negocio ni sistemas de información; en el otro, parte del personal conoce de sistemas, y algo de procesos de negocio y cadenas de valor. Por lo tanto, será muy valioso conocer el resultado de aplicar el *framework* a ambos casos antes de comenzar a realizar la implementación BPM, específicamente en lo que respecta a la documentación de los modelos, para que sea accesible y mantenible por los interesados en el futuro.

Referencias

- [1] C.-M. Chituc, A. Azevedo y C. Toscano, «A framework proposal for seamless interoperability in a collaborative networked environment,» *Computers in industry*, vol. 60, n° 5, pp. 317-338, 2009.
- [2] L. M. Camarinha-Matos, H. Afsarmanesh, N. Galeano y A. Molina, «Collaborative networked organizations—Concepts and practice in manufacturing enterprises,» *Computers & Industrial Engineering*, vol. 57, n° 1, pp. 46-60, 2009.
- [3] P. D. Villarreal, E. Salomone y O. Chiotti, «Modeling and Specification of Collaborative Business Processes with a MDS Approach and a UML Profile,» *Enterprise modeling and computing with UML*, pp. 13-44, 2007.
- [4] M. Weske, *Business process management: concepts, languages, architectures*, 2da ed., Springer Publishing Company, Incorporated, 2012.
- [5] Object Management Group (OMG), «Business Process Model and Notation (BPMN) version 2.0. Specification “formal/2011-01-03”. Technical Report.,» 2011. [En línea]. Available: <http://www.omg.org/spec/BPMN/2.0/PDF>.
- [6] I. M. Lazarte, L. H. Thom, C. Iochpe, O. Chiotti y P. D. Villarreal, «A distributed repository for managing business process models in cross-organizational collaborations,» *Computers in Industry*, vol. 64, n° 3, pp. 252-267, 2013.
- [7] R. S. Aguilar-Savén, «Business process modelling: Review and framework,» *International Journal of production economics*, vol. 90, n° 2, pp. 129-149, 2004.
- [8] M. B. Araújo y R. F. Gonçalves, «Selecting a notation to modeling business process: a systematic literature review of technics and tools,» de *IFIP International Conference on Advances in Production Management Systems*, 2016.
- [9] G. Reggio y M. Leotta, «A method-wise approach for selecting the most suitable business process modelling notation,» de *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019.
- [10] W. van der Aalst, «Modeling and analyzing interorganizational workflows,» de *Proceedings 1998 International Conference on Application of Concurrency to System Design*, 1998.
- [11] W. van der Aalst, «Verification of workflow nets,» de *International Conference on Application and Theory of Petri Nets*, 1997.
- [12] Object Management Group (OMG), «Unified Modeling Language (UML) version 2.5.1. Specification “formal/17-12-05”,» 2017. [En línea]. Available: <https://www.omg.org/spec/UML/2.5.1/PDF>.
- [13] P. D. Villarreal, E. Salomone y O. Chiotti, «A MDA-based development process for collaborative business processes,» *Milestones, Models and Mappings for Model-Driven Architecture*, vol. 17, 2006.
- [14] M. Dumas, M. La Rosa, J. Mendling y H. A. Reijers, *Fundamentals of Business Process Management*, Springer-Verlag Berlin Heidelberg, 2013.
- [15] D. Cocconi, J. P. Ferreyra, C. Verino y M. Pérez, «Optimización organizacional basada en la aplicación del ciclo de vida BPM completo para la mejora continua de los procesos de negocio,» de *6to Congreso Nacional de Ingeniería Informática – Sistemas de Información, CoNaISI 2018*, Mar del Plata, Argentina, 2018.
- [16] D. A. Cocconi, J. M. Roa y P. D. Villarreal, «Collaborative Business Process Management Through a Platform Based on Cloud Computing,» *CLEI Electronic Journal*, vol. 21, n° 2, pp. 6:1-6:26, 2018.
- [17] I. M. Lazarte, E. Tello-Leal, J. Roa, O. Chiotti y P. D. Villarreal, «Model-driven development methodology

- for B2B collaborations,» de *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, 2010.
- [18] W. M. P. van der Aalst y M. Weske, «The P2P approach to interorganizational workflows,» de *International conference on advanced information systems engineering*, 2001.
- [19] J. Arlow y I. Neustadt, UML 2, ANAYA, 2005.
- [20] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede y P. Wohed, «On the suitability of UML 2.0 activity diagrams for business process modelling,» de *APCCM'06 Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, 2006.
- [21] B. Hofreiter y C. Huemer, «A model-driven top-down approach to inter-organizational systems: From global choreography models to executable BPEL,» de *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, 2008.
- [22] P. D. Villarreal, I. Lazarte, J. Roa y O. Chiotti, «A modeling approach for collaborative business processes based on the up-colbip language,» de *International conference on business process management*, 2009.
- [23] D. Cocconi, M. Pérez, J. P. Ferreyra, C. Verino, G. Melano, N. Cocconi y A. Biasco, «Simulación de procesos de negocio utilizando técnicas de minería de procesos para estimar recursos computacionales necesarios para la apropiada implementación de un sistema BPMS/SIOP,» de *7mo. Congreso Nacional de Ingeniería Informática – Sistemas de Información, CoNalISI 2019 (San Justo, Buenos Aires, Argentina)*, 2019.
- [24] J. Webster y R. T. Watson, «Analyzing the past to prepare for the future: Writing a literature review,» *MIS quarterly*, pp. xiii-xxiii, 2002.
- [25] J. L. Pereira y D. Silva, «Business process modeling languages: A comparative framework,» de *New Advances in Information Systems and Technologies*, Springer, Cham, 2016, pp. 619-628.
- [26] Z. Yan, H. A. Reijers y R. M. Dijkman, «An evaluation of BPMN modeling tools,» de *International Workshop on Business Process Modeling Notation*, 2010.
- [27] S. H. Malekan y H. Afsarmanesh, «Overview of business process modeling languages supporting enterprise collaboration,» de *International Symposium on Business Modeling and Software Design*, 2013.
- [28] M. Cortes Cornax, S. Dupuy-Chessa, D. Rieu y M. Dumas, «Evaluating choreographies in BPMN 2.0 using an extended quality framework,» de *International Workshop on Business Process Modeling Notation*, 2011.
- [29] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede y N. Russell, «On the suitability of BPMN for business process modelling,» de *International conference on business process management*, 2006.
- [30] W. M. P. van der Aalst y A. H. M. ter Hofstede, «Workflow patterns: On the expressive power of (petri-net-based) workflow languages,» de *Proceedings of the fourth workshop on the practical use of coloured petri nets and CPN Tools (CPN 2002)*, 2002.
- [31] H. Shen, A. Virani y J. Niu, «Formalize UML 2 sequence diagrams,» de *2008 11th IEEE High Assurance Systems Engineering Symposium*, 2008.
- [32] O. Kopp, F. Leymann y S. Wagner, «Modeling choreographies: BPMN 2.0 versus BPEL-based approaches,» de *Enterprise modelling and information systems architectures (EMISA 2011)*, 2011.
- [33] R. M. Dijkman, M. Dumas y C. Ouyang, «Formal semantics and analysis of BPMN process models using Petri nets,» 2007.
- [34] L. M. Daniele, L. Ferreira Pires y M. van Sinderen, «Process-Oriented Behavior Generation Using Interaction Patterns,» de *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, 2010.
- [35] F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi y A. Vandin, «BProVe: a formal verification framework for business process models,» de *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2017.
- [36] M. Chinosi y A. Trombetta, «BPMN: An introduction to the standard,» *Computer Standards & Interfaces*, vol. 34, n° 1, pp. 124-134, 2012.
- [37] N. Genon, P. Heymans y D. Amyot, «Analysing the cognitive effectiveness of the BPMN 2.0 visual notation,» de *International conference on software language engineering*, 2010.
- [38] L. Li, X. Li, T. He y J. Xiong, «Extensics-based test case generation for UML activity diagram,» *Procedia Computer Science*, vol. 17, pp. 1186-1193, 2013.
- [39] D. Ritter, J. Ackermann, A. Bhatt y F. O. Hoffmann, «Building a business graph system and network integration model based on BPMN,» de *International Workshop on Business Process Modeling Notation*, 2011.
- [40] F. Corradini, A. Morichetta, A. Polini, B. Re y F. Tiezzi, «Collaboration vs. choreography conformance in BPMN 2.0: from theory to practice,» de *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, 2018.
- [41] M. Geiger y G. Wirtz, «BPMN 2.0 serialization-standard compliance issues and evaluation of modeling tools,» *Enterprise Modelling and Information Systems Architectures (EMISA 2013)*, 2013.
- [42] R. Braun, «BPMN Extension Profiles--Adapting the Profile Mechanism for Integrated BPMN Extensibility,» de *2015 IEEE 17th Conference on Business Informatics*, 2015.
- [43] B. List y B. Korherr, «An evaluation of conceptual business process modelling languages,» de *Proceedings*

of the 2006 ACM symposium on Applied computing, 2006.

- [44] A. Schönberger, «Visualizing b2bi choreographies,» de *2011 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2011.
- [45] D. Birkmeier y S. Overhage, «Is BPMN really first choice in joint architecture development? an empirical study on the usability of BPMN and UML activity diagrams for business users,» de *International Conference on the Quality of Software Architectures*, 2010.