

# Software Agents for Management Dynamic Inter-Organizational Collaborations

E. T. Leal, O. Chiotti, and P. D. Villarreal

**Abstract**— The globalization, modern markets, as well as new organizational management philosophies and advances in Information and Communications Technologies, encourage organizations to establish collaboration networks or inter-organizational collaborations. In this paper we propose a technology solution based on software agents which allows supporting the management of collaborative business processes in environments dynamic inter-organizational collaborations. First, we propose a software agent platform that integrates in agent specification's the notions of Belief-Desire-Intention agent architecture with functionalities of process-aware information systems. The platform enables organizations to negotiate collaborations agreements in electronic format to establish dynamic inter-organizational collaborations and define the collaborative processes to be executed. Second, we propose a methodology that includes methods based on Model-Driven Development, which enable the generation of executable process models and the code of process-oriented agents, derived from conceptual models of collaborative processes. This methodology and methods are implemented and automated by software agents that enable the generations of these implementation artifacts, at run-time of the platform. Therefore, the platform enables the automatic generation of the technology solution that requires each organization to execute the agreed collaborative processes, where the generated artifacts are built and initialized in the platform, allowing the implementation and execution of these processes. In this way, the proposed agent-based platform allows to establish collaboration among heterogeneous and autonomous organizations focusing in the process-oriented integration.

**Keywords**— Software Agents, Dynamic Inter-Organizational Collaborations, Model-Driven Development, Collaborative Business Process.

## I. INTRODUCCIÓN

LA GLOBALIZACIÓN, los mercados modernos, las nuevas filosofías de gestión de organizaciones y los avances en las Tecnologías de Información y Comunicaciones (TICs), alientan a las organizaciones a establecer relaciones estrechas de integración, cooperación y colaboraciones entre ellas, dando lugar a las colaboraciones inter-organizacionales o redes de colaboración [1]. Este tipo de colaboraciones han demostrado traer beneficios para sus participantes, tales como disminución de costos, mejoras de rendimiento en las organizaciones, manejo eficiente de información en tiempo real, etc. Una colaboración inter-organizacional implica una integración orientada a procesos entre organizaciones heterogéneas y autónomas, que debe ser alcanzada tanto a *nivel de negocio* como a *nivel tecnológico* [2][3].

A *nivel de negocio*, la solución consiste en aplicar un modelo de gestión que defina las reglas generales que gobiernan la colaboración entre organizaciones. En este nivel, la integración y colaboración entre organizaciones se alcanza a través de la definición y ejecución de los procesos de negocio colaborativos. Un *proceso de negocio colaborativo*, también llamado *coreografía de proceso* [3], define el comportamiento de las interacciones entre las organizaciones y sus roles desde un punto de vista global, esto es, cómo coordinan sus acciones e intercambian información, con el propósito de tomar decisiones en forma conjunta para alcanzar una meta de negocio común [2][4].

Una gestión descentralizada de los procesos colaborativos es requerida para mantener la autonomía de las organizaciones [2], de tal manera que cada organización gobierne y gestione en forma independiente sus procesos privados y la información que produce para dar soporte a los procesos colaborativos, la cual se puede alcanzar a través de la ejecución de los procesos de integración de las organizaciones involucradas [5]. Un *proceso de integración* define tanto el comportamiento público como el comportamiento privado que soporta el rol que una organización cumple en un proceso colaborativo. Este proceso contiene la lógica pública y privada requerida para procesar o generar el intercambio de información entre las organizaciones [5].

A *nivel tecnológico*, la solución se centra en la funcionalidad y la interoperabilidad de los sistemas de información inter-organizacionales para automatizar y ejecutar los procesos colaborativos. Esto implica desarrollar e implementar por las organizaciones sistemas de información autónomos que sean interoperables y lleven a cabo la ejecución de los procesos de integración, posibilitando el intercambio de información y las interacciones entre dichos sistemas, para dar soporte a la gestión descentralizada de los procesos colaborativos [2]. Esta interoperabilidad de sistemas puede ser alcanzada a través de la definición de modelos o especificaciones ejecutables, tanto de las interfaces de los sistemas, así como de los procesos de integración de las organizaciones, basadas en estándares de-facto o definidos *ad-hoc*, que luego puedan ser interpretadas por *sistemas de información orientados a procesos* (SIOPs, o *process-aware information systems*) [6]. Un SIOP, es aquel que ejecuta procesos, en los que están involucradas personas, aplicaciones y fuentes de información, sobre la base de modelos o especificaciones ejecutables de procesos que son interpretadas por un motor de procesos [7].

El enfoque tradicional de gestionar colaboraciones inter-organizacionales basadas en acuerdos estáticos que predefinen los procesos colaborativos a ejecutar, conlleva costos y tiempos de desarrollo elevados para la implementación y/o adaptación de sistemas de información inter-organizacionales

E. T. Leal, Universidad Autónoma de Tamaulipas, Victoria, Tamaulipas, México, etello@uat.edu.mx

O. Chiotti, INGAR, Consejo Nacional de Investigaciones Científicas y Técnicas, Santa Fe, Argentina, chiotti@santafe-conicet.gov.ar

P. D. Villarreal, Universidad Tecnológica Nacional, Santa Fe, Argentina. pvillarr@frsf.utn.edu.ar

ante cambios de requerimientos de negocio en los procesos colaborativos [8].

Por lo tanto, para responder a las nuevas oportunidades de negocio o adaptarse a los frecuentes cambios del entorno, las organizaciones requieren entablar *colaboraciones inter-organizacionales dinámicas*, lo cual implica que, a través del uso de Tecnologías de Información, dos o más organizaciones puedan acordar en forma electrónica llevar a cabo una colaboración inter-organizacional definiendo los procesos colaborativos a ser ejecutados [9]-[12]. Ante una nueva colaboración que entablan o una colaboración ya establecida, las organizaciones acuerdan ejecutar nuevos procesos o nuevas versiones de los procesos existentes para adaptarlos a los nuevos requerimientos de negocio. Para ello, el acuerdo de cuáles procesos colaborativos ejecutar, se realiza en forma electrónica a través de un proceso de negociación en donde, modelos predefinidos de dichos procesos, son intercambiados y acordados entre las organizaciones. Estos modelos pueden ser obtenidos de repositorios públicos o privados de modelos de procesos colaborativos o bien definidos *ad-hoc* en forma conjunta por las organizaciones. Cuáles procesos ejecutar se decide sobre la marcha de la colaboración, a través de negociaciones para acordar la gestión de los mismos [8],[10].

Los cambios a los procesos o nuevos procesos requieren que se modifiquen los sistemas que los soportan, esto es, que se realice un nuevo desarrollo para adaptar el sistema a los cambios efectuados en los procesos. Este nuevo desarrollo implica una inversión importante en recursos humanos, tiempo y costos por parte de las organizaciones, para el rediseño, recodificación, re-compilación y redespigue de los sistemas de información inter-organizacionales. Por lo tanto, el desarrollo de estos sistemas requiere de nuevas metodologías y métodos de desarrollo de software, arquitecturas de software y de la infraestructura o plataforma adecuada para dar soporte a los diferentes aspectos y requerimientos de las colaboraciones inter-organizacionales dinámicas.

Con esta finalidad, en este trabajo se presenta una solución tecnológica basada en agentes de software para dar soporte a la gestión de procesos colaborativos en ambientes de colaboración inter-organizacionales dinámicos. Por un lado, se propone una plataforma de agentes de software que posibilita que una organización pueda acordar en forma electrónica con otras organizaciones la implementación y ejecución de procesos colaborativos, a partir de modelos de procesos que son intercambiados entre las organizaciones. La plataforma está integrada por agentes estáticos, cuyo comportamiento está predefinido en la plataforma, y agentes dinámicos, cuyo comportamiento es generado en tiempo de ejecución del sistema. Por otro lado, con el propósito de guiar el proceso de desarrollo para generar soluciones tecnológicas (agentes orientados a procesos dinámicos), a partir de modelos conceptuales de procesos colaborativos, se propone una metodología que sigue un enfoque *top-down* basada en los principios del desarrollo dirigido por modelos (MDD, del inglés *Model-Driven Development*) [13],[14]. En esta metodología se incluyen métodos basados en MDD que permiten generar, para cada organización involucrada en un proceso colaborativo, un modelo de proceso ejecutable (que implementa un proceso de integración) y el código del agente dinámico orientado a procesos. Estos, en conjunto, representan

los artefactos de implementación requeridos por una organización para ejecutar un proceso colaborativo. Adicionalmente, para satisfacer los requerimientos de las colaboraciones inter-organizacionales dinámicas, la metodología y los métodos son implementados y automatizados mediante un agente estático provisto en la plataforma, el cual genera en forma automática los artefactos de implementación, posterior a establecer un acuerdo de colaboración en forma electrónica a través de la plataforma. De esta manera, en tiempo de ejecución de la plataforma, se posibilita generar en forma automática la solución tecnológica basada en agentes de software que permite ejecutar procesos colaborativos.

## II. METODOLOGÍA

La metodología propuesta tiene como objetivo guiar el proceso de desarrollo de soluciones tecnológicas basadas en agentes de software, utilizando como base el marco conceptual presentado en [5], el cual proporciona los diferentes niveles de abstracción en los que se debe enfocar el desarrollo de colaboraciones inter-organizacionales. Adicionalmente, la metodología se basa en la propuesta metodológica presentada en [15], la cual es aplicada a un caso concreto del área de servicios electrónicos de atención a la salud (*e-healthcare*) con un enfoque de colaboración inter-organizacional tradicional o estático, que se caracteriza por negociaciones y acuerdos de colaboración que se realizan cara a cara entre las organizaciones participantes para definir los procesos colaborativos a ejecutar. Sin embargo, para cumplir con los requerimientos de las colaboraciones inter-organizacionales dinámicas se propone una adaptación de dicha metodología, permitiendo dar soporte a la parte dinámica de las colaboraciones inter-organizacionales.

La metodología propuesta apunta al uso de modelos conceptuales desde diferentes puntos de vista, niveles de abstracción y de granularidad. Los artefactos de salida de las fases de esta metodología se representan por medio de modelos (de procesos y/o sistemas), generados mediante la aplicación de los principios del MDD [13],[16]-[18], tal como se muestra en la Fig. 1. El resultado final son los artefactos de implementación, estos están representados por los modelos de procesos de integración ejecutable y el código de los agentes de software que representan los sistemas de información inter-organizacionales de cada participante en la colaboración.

### A. Fases de la metodología

La metodología para el desarrollo de soluciones tecnológicas basadas en agentes de software en ambientes de colaboraciones inter-organizacionales dinámicas se compone por la fase de *diseño de solución de negocio* y por la fase de *generación de la solución tecnológica* mediante las cuales se generan la solución de negocio y la solución tecnológica respectivamente.

La fase de *diseño de la solución de negocio* está conformada por la actividad de *diseño del proceso de integración*. La actividad *diseño del proceso de integración* consiste en llevar a cabo la definición y derivación de un modelo de proceso de integración a partir de un modelo de proceso colaborativo. Esto implica que esta actividad se realiza varias veces en paralelo por las diferentes

organizaciones (Fig. 1). Por cada proceso colaborativo, cada una de las organizaciones involucradas debe realizar esta actividad para generar su correspondiente modelo de proceso de integración.

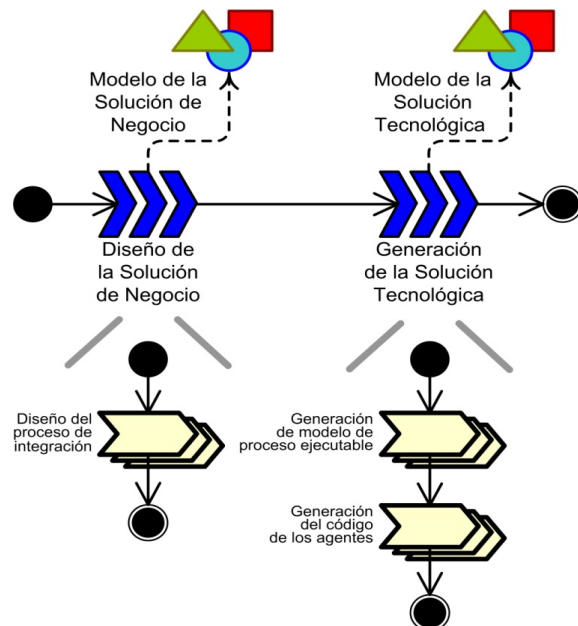


Figura 1. Fases de la metodología propuesta para generar soluciones tecnológicas en colaboraciones inter-organizacionales dinámicas.

El *diseño de un proceso de integración* implica la definición, desde el punto de vista de una organización, tanto de las actividades públicas que soportan el envío y recepción de mensajes (que transportan documentos de negocio) con otras organizaciones, como de las actividades privadas que procesan los documentos de negocio recibidos y aquellas que generan los documentos de negocio que serán intercambiados a través de los mensajes. Para representar los modelos de procesos de integración se utiliza el lenguaje de Notación para el Modelado de Procesos de Negocio (BPMN, del inglés *Business Process Model and Notation*) [19]. Estos modelos son derivados mediante la aplicación de un método MDD propuesto en [20], el cual permite generar un modelo BPMN del proceso de integración de una organización, a partir de un modelo de proceso colaborativo definido con el lenguaje de Modelado de Procesos de Negocio Colaborativos basados en Protocolos de Interacción (UP-ColBPIP, del inglés *UML Profile for Collaborative Business Processes based on Interaction Protocols*) [21]. Esto es, un proceso colaborativo representado como un protocolo de interacción, que es utilizado como modelo de entrada para generar un diagrama de proceso BPMN que representa un proceso de integración de una organización involucrada en el proceso colaborativo. El lenguaje UP-ColBPIP [21] permite definir formalmente el comportamiento de los procesos colaborativos mediante protocolos de interacción mediante el uso de actos de comunicación (*speech acts*).

Las reglas de transformación definidas en el método [20] agregan las actividades públicas del proceso de integración a partir de los mensajes contenidos en el protocolo de

interacción. Las actividades privadas del proceso, que soportan el intercambio de mensajes, son generadas utilizando reglas de transformación basadas en la teoría de patrones de actividades [22].

La fase *generación de la solución tecnológica* está conformada por las actividades: *generación de modelo de proceso ejecutable* y *generación del código de los agentes*, como se muestra en la Fig. 1. La solución tecnológica apunta a posibilitar la ejecución descentralizada de los procesos colaborativos, mediante la implementación, por cada organización, de sus procesos de integración que son ejecutados a través de agentes orientados a procesos. Estos agentes representan los sistemas de información inter-organizacionales. Un modelo de proceso ejecutable se refiere a la implementación de un proceso de integración. Un agente orientado a procesos está compuesto de un motor de procesos embebido, a través del cual interpreta un modelo de proceso ejecutable para realizar la ejecución de un proceso de integración del rol de la organización que representa. Los modelos de esta fase son definidos usando conceptos de una plataforma de implementación específica.

Esta fase y sus actividades son realizadas en paralelo por cada organización involucrada en los procesos colaborativos definidos, las cuales tienen que estar de acuerdo en la tecnología o plataforma que se utilizará para la implementación. El resultado de esta fase son los artefactos de implementación: los modelos de procesos ejecutables y el código que representa la estructura y comportamiento de los agentes orientados a procesos. Mediante estos artefactos, los agentes pueden ser instanciados en la plataforma de agentes de software propuesta en la Sección IV, y llevar a cabo la ejecución de los procesos colaborativos.

La actividad *generación de modelo de proceso ejecutable* consiste en la definición de un modelo de proceso basado en el lenguaje BPMN [19], conteniendo los parámetros de configuración y la semántica necesaria para su ejecución. Para cada proceso colaborativo, esta actividad debe ser realizada en paralelo por cada organización que participa en el mismo. Por cada organización, se toma como entrada un modelo conceptual (definido en un nivel independiente de la plataforma) de un proceso de integración, el cual representa el comportamiento del rol que la organización desempeña en un proceso colaborativo, y se obtiene como salida un modelo de proceso ejecutable, definido en un nivel específico de la plataforma. En este trabajo de investigación se propone que ambos modelos sean especificados utilizando el lenguaje BPMN. Entonces, la *generación de modelo de proceso ejecutable* implica la definición de un proceso de negocio ejecutable en una plataforma específica de implementación, en donde el modelo contiene anotaciones y detalles de implementación en cada uno de sus elementos, los cuales son requeridos para habilitar su ejecución. De esta manera, el modelo generado almacena la información necesaria para ser ejecutado mediante un motor de procesos.

La *generación del código de los agentes* consiste en llevar a cabo la definición y derivación de los artefactos de implementación finales de los agentes de software, a partir de un modelo conceptual de proceso de integración definido con el lenguaje BPMN. Esta actividad también es realizada en forma paralela por las diferentes organizaciones. El código

generado representa la estructura y el comportamiento de un agente orientado a procesos que permite la ejecución de un modelo de proceso ejecutable. El rol que desempeña un participante (organización) en una colaboración se utiliza para identificar el rol del agente. Para cada participante, un rol es definido con su respectivo comportamiento, el cual es derivado del modelo de proceso de integración correspondiente al participante. Este comportamiento es implementado por un modelo de proceso ejecutable y ejecutado a través de un plan definido en el agente, el cual desempeña el rol correspondiente. Dicho plan consiste en ejecutar el modelo de proceso ejecutable mediante el motor de procesos embebido en el agente. De esta manera se expresa la relación entre el rol y su comportamiento en el agente. Los mecanismos de interacción son agregados en el código del agente a partir de eventos de mensajes definidos en el modelo de proceso de integración.

### B. Métodos basados en el desarrollo dirigido por modelos

Los métodos basados en MDD posibilitan generar soluciones tecnológicas basadas en agentes de software a partir de modelos de procesos colaborativos intercambiados entre las organizaciones. Estos métodos dan soporte a cada una de las fases y actividades de la metodología propuesta para generar soluciones tecnológicas en ambientes de colaboración inter-organizacionales dinámicas. Entonces, a partir de un modelo de proceso colaborativo descrito como un protocolo de interacción, cada organización se centra en generar el modelo de proceso de integración del rol que esta organización desempeña en el modelo del proceso colaborativo. Luego, en forma paralela, un modelo de proceso ejecutable y un modelo de agente son construidos a partir del modelo de proceso de integración. Este procedimiento es llevado a cabo en forma independiente por cada organización participante en la colaboración. A continuación, el modelo del agente es transformado en un documento que contiene el código del agente, el cual representa el artefacto final de implementación del mismo.

En la Fig. 2 se muestran los métodos MDD [16]-[17], detallando los niveles de abstracción de modelos, que van desde un nivel PIM hasta el código. También se detallan los artefactos (modelos y código) que se generan en cada fase. Para cada uno de ellos se indica la transformación que se realiza, señalando el modelo y nivel del cual se parte, el modelo o código resultante de la transformación, y el nivel al que corresponde. Mediante la transformación **T1** se genera un modelo conceptual de proceso de integración de una organización, a partir de un modelo de proceso colaborativo, mediante una transformación horizontal PIM-a-PIM, utilizando el método propuesto en [20]. La transformación modelo-a-modelo **T2** apunta a generar un modelo de proceso ejecutable usando como entrada un modelo del proceso de integración. Por un lado, el modelo del proceso de integración conforma un modelo conceptual en un nivel independiente de la plataforma PIM con alto nivel de abstracción. Por otro lado, el modelo de proceso ejecutable se define en un nivel PSM utilizando conceptos específicos de la plataforma de ejecución del proceso que se seleccione. El modelo generado es replicado a nivel de código porque es usado también como

artefacto de implementación para realizar la ejecución del mismo, tal como se muestra en la Fig. 2.

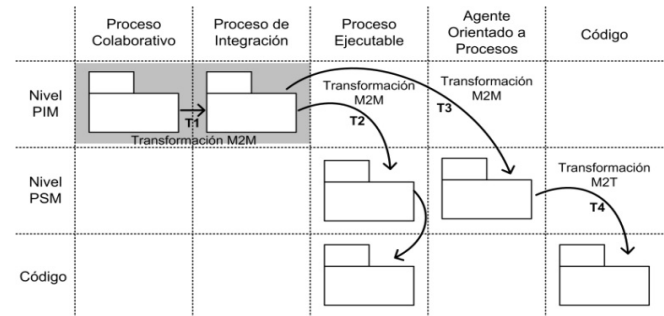


Figura 2. Métodos MDD para colaboraciones inter organizacionales dinámicas.

El código de un agente orientado a procesos es generado mediante dos métodos dirigidos por modelos. El primer método aplica la transformación modelo-a-modelo **T3** (Fig. 2), generando un modelo de salida basado en la plataforma de implementación de agentes que se seleccione, utilizando un modelo de proceso de integración como entrada. El modelo generado es definido en un nivel específico de la plataforma PSM utilizando conceptos de la plataforma de implementación de agentes. Las reglas de transformación definidas en el método agregan: un agente a partir del participante contenido en el modelo de proceso de integración; un plan, que implementará el agente derivado del nombre del proceso que acordaron las organizaciones ejecutar; y los eventos de mensaje del agente, que son generados a partir de las tareas de enviar o recibir contenidas en el proceso de integración. Los eventos de mensaje representan la lógica de interacción del agente y habilitan los mecanismos de comunicación con los agentes que desempeñan otros roles en el proceso colaborativo. Estos eventos de mensaje están basados en el lenguaje de comunicación de agentes FIPA-ACL (*Foundation for Intelligent Physical Agents - Agent Communication Language*) [23]. Por último, la meta definida entre los participantes del proceso se utiliza para generar la meta que debe alcanzar el agente mediante la ejecución de sus planes y eventos de mensaje.

El segundo método para generar el código de un agente orientado a procesos se lleva a cabo mediante la transformación directa modelo-a-código **T4** (Fig. 2), que consiste en la generación de un documento con el código fuente que representa la estructura y comportamiento de un agente orientado a procesos en el formato de implementación de la plataforma de agentes de software seleccionada. Este método utiliza como entrada al modelo del agente, generado en la transformación previa, y se obtiene un documento ejecutable con el código que posibilita la implementación del agente.

## III. IMPLEMENTACIÓN DE LA METODOLOGÍA Y LOS MÉTODOS MDD

Un conjunto de herramientas se desarrollaron con el fin de probar la funcionalidad de la metodología propuesta y su futura aplicación por parte de las organizaciones. Estas herramientas están construidas sobre la plataforma Eclipse

[24], con el fin de aprovechar las ventajas que ofrece este ambiente de desarrollo y de los mecanismos de extensión que proporciona. Las herramientas se implementan como un motor de transformación dentro de la plataforma de modelado de Eclipse, utilizando transformaciones modelo-a-modelo (M2M) y modelo-a-texto (M2T) siguiendo los principios del enfoque MDD. La Fig. 3 muestra una vista de la arquitectura del IDE basado en Eclipse que da soporte a los métodos MDD. Las reglas de transformación contenidas en los métodos MDD (tanto para M2M como para M2T) son especificadas con el lenguaje estándar para la transformación de modelos ATL (*ATLAS Transformation Language*) [25]-[26]. Todos los meta-modelos utilizados en IDE son implementados mediante EMF (*Eclipse Modeling Framework*) [27].

El primer motor de transformación interpreta y ejecuta la definición de la transformación para generar un modelo de proceso de integración derivado de un modelo de proceso colaborativo, utilizando el meta-modelo del lenguaje UP-ColBPIP como meta-modelo de entrada y el meta-modelo del lenguaje BPMN como meta-modelo destino. El segundo motor de transformación implementa reglas de transformación para derivar un modelo de procesos de negocio ejecutable definido como un *workflow* BPMN de Jadex [28], a partir de un modelo conceptual del proceso de integración, usando el meta-modelo del lenguaje BPMN como meta-modelo de entrada y el meta-modelo del BPMN del *framework* Jadex como meta-modelo de salida.

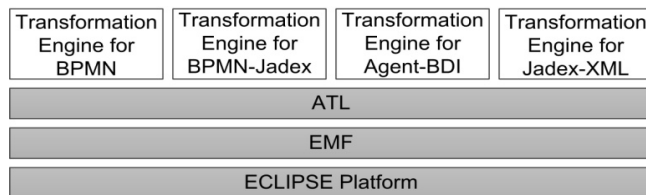


Figura 3. Arquitectura del IDE basado en Eclipse para soportar los métodos.

El tercer motor de transformación implementa un método que contiene reglas de transformación para generar un modelo de un agente de software BDI (*Belief-Desire-Intention*) [29], utilizando el modelo conceptual del proceso de integración como entrada al método, usando el meta-modelo del lenguaje BPMN como meta-modelo de entrada. El último motor de transformación permite generar un artefacto de implementación que contiene la estructura de un agente de software BDI en formato XML basado en la plataforma de implementación de agentes de software Jadex, usando un meta-modelo basado en el enfoque de agentes BDI como meta-modelo de origen y el meta-modelo del lenguaje XML como meta-modelo destino. El documento generado contiene el código de implementación del agente de software que implementará el proceso colaborativo derivado del modelo de un agente de software BDI generado previamente.

#### IV. PLATAFORMA BASADA EN AGENTES DE SOFTWARE

La arquitectura global de la plataforma está representada por sistemas multi-agentes que son desplegados en cada una de las organizaciones involucradas en una colaboración inter-organizacional dinámica, con la meta de establecer un acuerdo

de colaboración y la ejecución de los procesos colaborativos. Por consiguiente, las principales funcionalidades a las que da soporte la plataforma de agentes de software propuesta son:

- Ejecutar procesos de negociación en forma electrónica que permitan establecer acuerdos de colaboración inter-organizacionales dinámicas, posibilitando el intercambio de modelos de procesos colaborativos entre las organizaciones y la selección de cuáles procesos colaborativos ejecutar,
- Ejecutar en forma descentralizada procesos colaborativos a través de la ejecución distribuida de los procesos de integración de las organizaciones, para lo cual se deben interpretar modelos de procesos de integración ejecutables y gestionar instancias de modelos de procesos de integración mediante un motor de procesos.
- Gestionar para cada organización un repositorio local de modelos de procesos colaborativos, así como los modelos de procesos y artefactos de implementación generados en forma automática por la plataforma.
- Generar, en forma automática en la plataforma, los artefactos de implementación que permitan ejecutar un proceso colaborativo, para lo cual se deben generar modelos de procesos de integración ejecutables para cada organización involucrada en un proceso colaborativo y generar el código de implementación de agentes orientados a procesos, los cuales ejecutarán los modelos de procesos de integración ejecutables.
- Intermediar el intercambio de información requerido en la ejecución de los procesos colaborativos con los sistemas internos de la organización, para lo cual se requiere gestionar la generación y procesamiento de documentos de negocio, que representan la información a intercambiar en los procesos colaborativos.

Los agentes que integran la plataforma están basados en el modelo BDI [29]. La arquitectura se compone de cinco agentes, cuyas responsabilidades se describen a continuación:

- *Agente Administrador de Colaboraciones (AC)*. Este agente representa a una organización y es responsable de establecer comunicaciones con agentes AC de otras organizaciones, con el objetivo de establecer un acuerdo de colaboración que permita la ejecución de procesos colaborativos. Además, es responsable de crear instancias de un agente *Administrador de Proceso (AP)* de una organización.
- *Agente Administrador de Proceso (AP)*. Es responsable de desempeñar el rol que una organización cumple en la ejecución de un proceso colaborativo acordado por las organizaciones involucradas en una colaboración inter-organizacional. Para desempeñar este rol interpreta un modelo de proceso de integración ejecutable mediante un motor de procesos embebido en el agente. Durante el tiempo en que el acuerdo de colaboración está vigente, una instancia de un agente AP en cada organización está disponible para ejecutar una nueva instancia del proceso colaborativo acordado.
- *Agente Administrador de Modelos (AM)*. Es responsable de gestionar (almacenar y recuperar) los modelos de procesos de negocio de las colaboraciones en las que

participa una organización, y almacenarlos en repositorios locales de esa organización.

- **Agente Generador de Implementaciones (GI).** Es responsable de realizar las transformaciones de modelos de procesos y generar los artefactos de implementación, esto es, los modelos de procesos ejecutables y el código de los agentes AP que los interpretan. Para ello, este agente automatiza los métodos MDD de la metodología propuesta mediante una máquina de transformación embebida en este agente.
- **Agente Integrador de Sistemas (IS).** Es responsable de la integración entre los sistemas internos de la organización y la plataforma de agentes de software. Esta integración permite recuperar la información (representada mediante documentos de negocio) intercambiada en la ejecución de los procesos colaborativos.

Los agentes definidos en la plataforma se clasifican en dos tipos de agentes: *estáticos* y *dinámicos*. Los *agentes estáticos* se refieren a agentes cuyo comportamiento es definido en tiempo de diseño e implementación. Estos agentes son provistos en la instalación de la plataforma. Las funcionalidades del comportamiento de estos agentes se mantienen sin cambios en tiempo de ejecución de la plataforma, y sólo sus creencias son modificadas durante la ejecución del agente.

Los *agentes dinámicos* se refieren a un nuevo tipo de agente propuesto en este trabajo de investigación, cuyas funcionalidades, comportamiento, implementación e instancias se generan en tiempo de ejecución de la plataforma. Las funcionalidades de estos agentes no se conocen de antemano en tiempo de diseño. La implementación de estos agentes puede ser generada en tiempo de ejecución a través de la automatización de métodos MDD que soporten transformaciones de modelos que generan los artefactos y código de implementación de los agentes. En la plataforma propuesta, el tipo de agente AP es un agente dinámico, el cual a su vez es orientado a procesos, ya que tiene un motor de procesos embebido. Estos agentes dinámicos orientados a procesos dan soporte a la ejecución de los procesos colaborativos a través de la ejecución de los procesos de integración. La lógica y comportamiento de estos agentes se define en un modelo de proceso ejecutable mediante el cual se gobierna la ejecución de actividades y el intercambio de mensajes, permitiendo cumplir con la meta de ejecutar un modelo de proceso de integración.

Las interacciones entre los agentes son modeladas como protocolos de interacción utilizando el lenguaje UP-ColBPIP [21]. Mediante diagramas de interacción se describe el comportamiento de las interacciones entre los agentes, definiendo una secuencia permitida de mensajes entre los agentes participantes. Esto permite construir parte de la estructura de cada agente y da lugar a las especificaciones que describen cómo cada agente se comporta con el fin de ejecutar un plan o satisfacer una meta.

El comportamiento predefinido de las interacciones entre los agentes involucrados para satisfacer la meta de establecer un acuerdo de colaboración se muestra en el protocolo de interacción presentado en la Fig. 4. Dicho protocolo inicia cuando el agente AC de la Organización-A, desempeñando el

rol de *iniciador*, envía un mensaje con un *acto de comunicación* de tipo *request* al agente AC de la Organización-B, desempeñando el rol de *receptor*. Este mensaje contiene un documento que describe los términos de un acuerdo de colaboración (objetivo de la colaboración, alcances del acuerdo, duración de la colaboración, descripción y responsabilidades de cada rol, roles asignados a cada organización, meta de negocio) y el nombre de los procesos colaborativos que se requieren ejecutar para alcanzar una meta de negocio común.

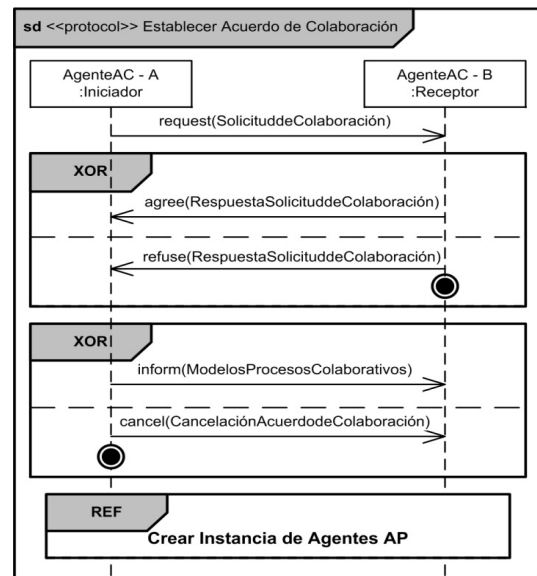


Figura 4. Protocolo de interacción Establecer Acuerdo de Colaboración.

El agente *receptor* procesa el mensaje, realizando una evaluación de la solicitud mediante una actividad interna del agente. El agente AC *receptor* puede responder mediante un mensaje *refuse*, con lo cual deniega establecer la colaboración y el protocolo finaliza. De lo contrario, puede responder con un mensaje *agree*, aceptando las condiciones contenidas en el documento propuesto. Cuando el agente *iniciador* procesa el mensaje *agree* recibido, inmediatamente ejecuta un protocolo de interacción con el agente AM de su organización, para obtener los modelos a enviar de los procesos colaborativos que aceptó ejecutar el agente AC *receptor*. Si dicho protocolo finaliza correctamente, es decir, el agente AM pudo recuperar y enviar al agente AC los modelos de procesos colaborativos solicitados, el agente AC *iniciador* de inmediato retransmite estos modelos al agente AC *receptor* mediante un mensaje de tipo *inform*. De lo contrario, sino se pudieron recuperar todos los modelos de procesos colaborativos solicitados, el agente AC *iniciador* envía un mensaje *cancel* indicando la cancelación del acuerdo de colaboración y la finalización de las interacciones entre los agentes AC.

Finalmente, posterior al intercambio de los modelos de procesos colaborativos, los agentes AC de las organizaciones ejecutarán un sub-protocolo *Crear Instancia de Agentes AP* a través del cual se realizará la generación de la solución tecnológica y la creación de las instancias de los agentes AP de cada una de las partes.

El protocolo de interacción detallado anteriormente permite cumplir con los requerimientos de las colaboraciones inter-organizacionales dinámicas, en lo que se refiere a establecer una colaboración definiendo los procesos colaborativos a ejecutar e intercambiando los modelos que contienen dichos procesos (Fig. 4). Sin embargo, en las colaboraciones inter-organizacionales dinámicas también se requiere que todas las partes involucradas puedan proponer modificaciones cuando se establece el acuerdo de colaboración y en los procesos colaborativos acordados.

Por tal motivo, se ha definido una extensión del protocolo de interacción *Establecer Acuerdo de Colaboración* (Fig. 4) que permite proponer cambios al documento de la solicitud de colaboración. Además, se ha definido un protocolo de interacción, que complementa al protocolo presentado (Fig. 4), el cual habilita a las organizaciones a proponer cambios en los modelos de procesos colaborativos que se han acordado ejecutar, es decir, negociar nuevas versiones de modelos de dichos procesos, o proponer nuevos modelos de procesos colaborativos (por cuestiones de espacio no se muestran las figuras de estos protocolos de interacción).

El protocolo de interacción que se muestra en la Fig. 5 permite que los agentes AC cumplan la meta *generar instancia de agente AP*, que consiste en crear un agente AP, incorporarlo a la plataforma y vincularlo con el agente AP de la otra organización participante en el proceso colaborativo. Un agente AP es creado e inicializado en forma dinámica por un agente AC, basado en los artefactos de implementación que genera un agente GI.

Posterior a establecer un acuerdo de colaboración inter-organizacional mediante la plataforma de agentes y que se han intercambiado electrónicamente los modelos de procesos colaborativos acordados entre los agentes AC, el agente AC *receptor* procederá internamente a ejecutar las interacciones con el agente GI de su organización para generar la solución tecnológica y obtener los artefactos de implementación de cada uno de los procesos colaborativos acordados. Cuando el agente AC *receptor* recibe los artefactos de implementación, desde un agente GI, envía un mensaje *inform* para notificar a su contraparte (el agente AC *iniciador*), que la transformación finalizó en forma correcta; o envía un mensaje *failure* cuando la transformación no finalizó correctamente y la colaboración es cancelada (Fig. 5). Cuando el agente AC *receptor* envía un mensaje *inform*, permanecerá en espera de un mensaje que le autorice iniciar el procedimiento para crear una instancia del agente AP.

De la misma manera, cuando el agente AC *iniciador* recibe un mensaje *inform*, internamente ejecuta un protocolo de interacción con un agente GI, a través del cual el agente GI realizará un proceso de transformaciones de modelos y le enviará al agente AC el resultado de las transformaciones. En forma similar al agente AC *receptor*, el agente AC *iniciador* notificará el resultado de las transformaciones mediante mensajes de tipo *inform* o *failure*, según sea el caso.

A continuación, tomando los artefactos de implementación generados para un modelo de proceso colaborativo, el agente AC *iniciador* internamente crea una instancia de un agente AP por cada proceso colaborativo acordado. Luego, ejecuta un procedimiento que permite capturar los datos de identificación de los agentes AP instanciados, mediante estos datos de

identificación se permite establecer la comunicación entre los agentes AP. Entonces, los datos de identificación de los agentes AP son agregados al contenido de un mensaje *inform* que es enviado al agente AC *receptor*. En el caso de que se presente alguna falla al momento de crear alguno de los agentes AP, el agente AC *iniciador* enviará un mensaje de tipo *failure*, notificando la cancelación de la colaboración.

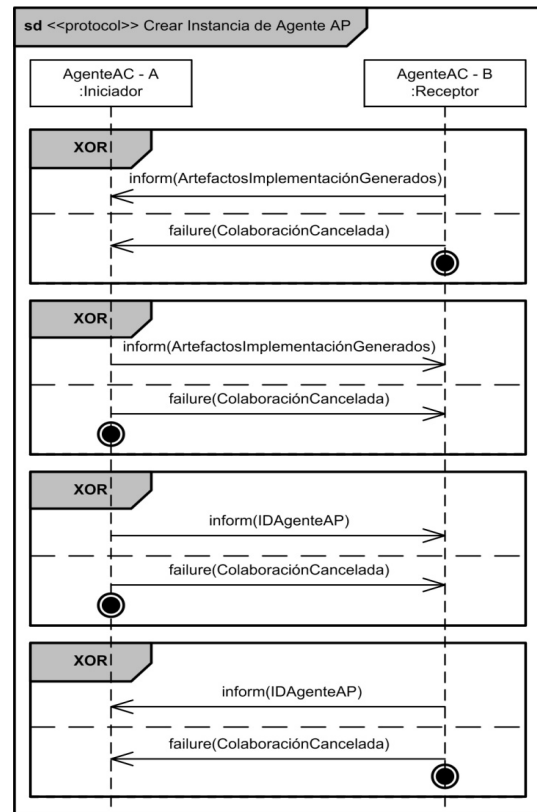


Figura 5. Protocolo de interacción Crear Instancia de Agente AP.

El agente AC *receptor* ejecuta un procedimiento similar de creación de agentes AP, capturando luego los datos de identificación de estos agentes instanciados, los cuales son enviados mediante un mensaje tipo *inform* al agente AC *iniciador*. En el caso de presentarse alguna falla en la creación de las instancias de los agentes AP, el agente AC *receptor* genera y envía un mensaje *failure* notificando la falla al agente AC *iniciador* y el proceso de colaboración se cancela. De esta manera, luego de crearse una instancia de los agentes AP de cada una de las partes, queda habilitada la ejecución de los procesos colaborativos acordados.

La implementación de la plataforma A4IOC (*Agents for Inter-Organizational Collaborations*) se llevó a cabo con el uso del *framework* y plataforma de agentes Jadex [30]-[31], la cual permite construir aplicaciones con componentes heterogéneos, tales como agentes y BPMN *workflows* (automatización y ejecución de modelos de procesos basados en BPMN). Por lo tanto, la especificación de los agentes de la plataforma sigue un enfoque de agentes BDI orientados a procesos, ya que el comportamiento de estos agentes se implementó y definió mediante modelos de procesos (BPMN *workflows*). Este modelo de procesos BPMN es implementado



como el plan principal de cada agente de software que conforman la plataforma, guiando y controlando su comportamiento.

Los elementos de tipo *tarea* contenidos en el modelo de procesos BPMN representan los sub-planes que el agente ejecutará. Estos elementos de tipo *tarea* son implementados mediante clases Java, que realizan acciones específicas como generar un documento, invocar un componente de la plataforma o guardar un archivo. Los sub-planes, en conjunto con las metas y eventos intermedios de mensaje, representan el comportamiento de un agente. Los elementos de evento intermedio de mensaje (que representan el envío o recepción de un mensaje) contenidos en la especificación de un modelo de procesos BPMN mantienen una relación con los mensajes definidos en la especificación del agente, debido a que cuando se ejecuta un evento intermedio de mensaje en el modelo de procesos BPMN invoca al agente para que ejecute los mecanismos de interacción, lo que habilita al agente a establecer la comunicación con otros agentes de la plataforma, tal como se muestra en la Fig. 6.

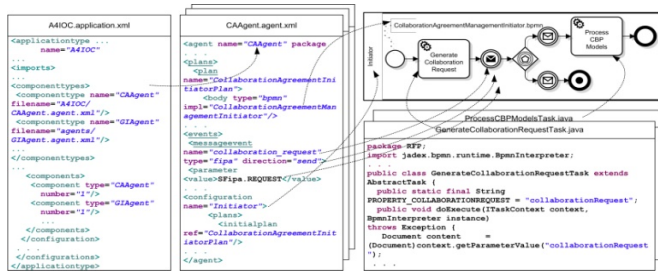


Figura 6. Componentes que integran la plataforma basada en agentes de software A4IOC.

En la Fig. 7 se presenta el comportamiento definido en el agente AC mediante un modelo de procesos BPMN, en donde las interacciones especificadas en los protocolos de interacción (Fig. 4 y Fig. 5) son definidas como evento intermedios de mensaje. En este modelo de procesos se integran las tareas y eventos que debe realizar el agente en forma coordinada para llevar adelante estos protocolos. Los eventos intermedios de mensaje definidos en el modelo BPMN representan las comunicaciones e interacciones de los protocolos de interacción que realiza un agente AC con otro agente AC, y entre el agente AC con sus agentes AM, GI, y AP que integran el sistema multi-agente. Estas interacciones son realizadas a través del intercambio de mensajes definidos con el lenguaje FIPA ACL [23]. La información intercambiada mediante estos eventos intermedios de mensaje es gestionada por actividades de tipo tarea que permiten al agente generar o almacenar la información contenida en los mensajes, y/o invocar sistemas internos u otros agentes de la plataforma.

## V. CASO DE ESTUDIO

En esta sección se describe la implementación de una colaboración inter-organizacional dinámica para un caso de estudio del dominio de la industria de las telecomunicaciones mediante la plataforma presentada. El caso de estudio consiste en la implementación de la plataforma de agentes de software propuesta en los servidores (conectados vía Internet) de las empresas MotoRepair y MotoParts, para que puedan

establecer colaboraciones inter-organizacionales dinámicas y ejecutar procesos colaborativos relacionados con la gestión de órdenes de compras y la provisión de productos de MotoParts a MotoRepair.

La empresa MotoRepair es el centro de reclamo de garantías de los productos fabricados por MotoBCS, en el cual se realizan la mayor cantidad de reparaciones de los productos MotoBCS a nivel mundial. La empresa MotoParts es una división de MotoBCS dedicada a la fabricación de componentes exclusivos para los productos fabricados por MotoBCS, los cuales son utilizados en los procesos de reparación en la empresa MotoRepair.

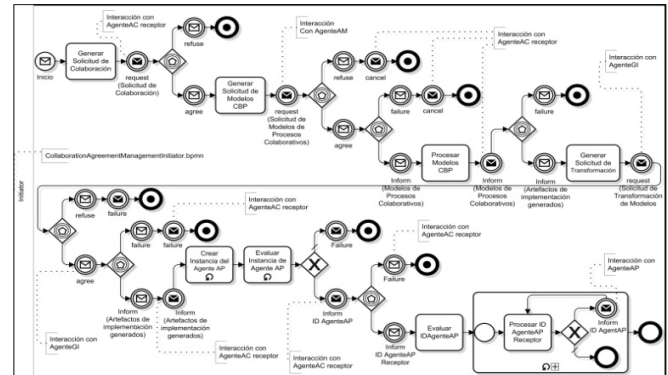


Figura 7. Comportamiento definido para el agente AC mediante un modelo de procesos BPMN.

El comportamiento definido en los agentes de software estáticos de la plataforma posibilita a las organizaciones establecer una colaboración inter-organizacional dinámica. En la Fig. 8 se muestran las interacciones ejecutadas entre los agentes de los MASs de la plataforma para implementar una colaboración inter-organizacional. Estas interacciones son el resultado de ejecutar los caminos de los protocolos de interacción entre estos agentes, lo que permite realizar el proceso de negociación entre las partes para establecer la colaboración y acordar los procesos colaborativos a ejecutar (interacciones del 1 al 6), generar en forma automática los modelos de procesos ejecutables que implementan los procesos de integración y el código de los agentes AP que ejecutan los procesos de integración (interacciones del 7 al 16), así como inicializar e incorporar dichos agentes AP en los MAS de las organizaciones (interacciones 17 y 18), para que ejecuten en forma descentralizada y coordinada los procesos colaborativos. En la Fig. 8, los agentes con el sufijo @A4IOC.server\_MX representan los agentes instanciados en el MAS de la organización MotoRepair. Los agentes con el sufijo @A4IOC.server\_BCS son aquellos instanciados en el MAS de la organización MotoParts.

Los resultados generados por la implementación de los métodos MDD mediante el agente GI de cada organización se presentan en la Fig. 8. En las interacciones 7 y 8 se muestra el proceso de negociación para generar los artefactos de implementación entre el agente AC y el agente GI del MAS de la empresa MotoParts. En la interacción 9 se muestra cómo el agente GI envía un documento de negocio conteniendo el código de los artefactos de implementación. En las interacciones 11, 12 y 13 se presenta un proceso de



negociación y generación de artefactos similar para el MAS de la empresa MotoRepair.

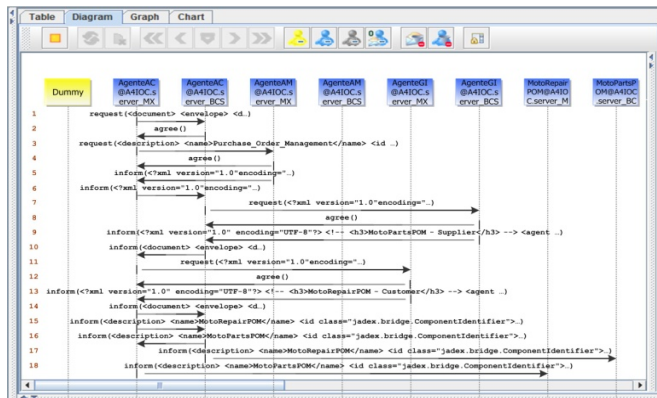


Figura 8. Interacciones entre los agentes para establecer una colaboración inter-organizacional dinámica.

Los artefactos de implementación son generados del proceso colaborativo intercambiado entre los agentes AC. Entonces, mediante el establecimiento de una colaboración inter-organizacional dinámica, MotoRepair le propone a MotoParts colaborar a través de la ejecución del proceso colaborativo "Gestión de Orden de Compra" (*Purchase Order Management*), cuyo comportamiento se muestra en el protocolo de interacción de la Fig. 9. En este proceso colaborativo, la empresa MotoRepair desempeña el rol de cliente y la empresa MotoParts desempeña el rol de proveedor de componentes. El proceso colaborativo tiene como meta de negocio reducir los tiempos en la gestión de la adquisición de componentes y acelerar el proceso de compra en MotoRepair mediante la automatización de las decisiones de confirmación en forma electrónica por parte del proveedor. El proceso permite a las partes negociar los tiempos de entrega de los componentes y proponer cambios en la orden de compra.

La Fig. 10 muestra un extracto del código generado para una actividad de tipo *enviar* representada como un *evento intermedio de mensaje* para un modelo de proceso ejecutable (nivel PSM) derivado de un modelo de proceso de integración (nivel PIM). Este código detalla las anotaciones requeridas para la ejecución de la actividad, tales como el *nombre* de la actividad, el participante *receptor* del mensaje, el *acto de comunicación* (*speech act*) utilizado en el mensaje, y el documento *contenido* en el mensaje, el cual contiene la información que se requiere enviar al receptor del mensaje.

En la ejecución del proceso colaborativo "Gestión de Orden de Compra" mediante los agentes AP MotoRepairPOM y MotoPartsPOM de los respectivos MAS implementados con la plataforma, se pueden presentar seis escenarios con diferentes comportamientos, determinados por las negociaciones que se llevan a cabo en dicho proceso. A continuación, se describe uno de estos escenarios.

La Fig. 11 muestra un escenario que se puede presentar en la ejecución del proceso colaborativo. En este caso, cuando el agente MotoRepairPOM (que representa un tipo de agente AP) propone una orden de compra mediante un mensaje *propose* (interacción 19), el agente MotoPartsPOM (que representa también un tipo de agente AP) responde con una

contrapropuesta de la orden de compra. Para lo cual, genera un documento de negocio *PurchaseOrderChange* que contiene las condiciones en que puede cumplir con los requerimientos de compra. Este documento es enviado mediante un mensaje *propose* (interacción 20), con lo cual se inicia una nueva negociación entre los agentes. El agente MotoRepairPOM, evalúa el documento propuesto y puede responder aceptando, rechazando o proponiendo cambios a la orden de compra recibida. En este caso, el agente MotoRepairPOM responde con un mensaje *accept-proposal* (interacción 21), con lo cual se aceptan los cambios propuestos en la orden de compra.

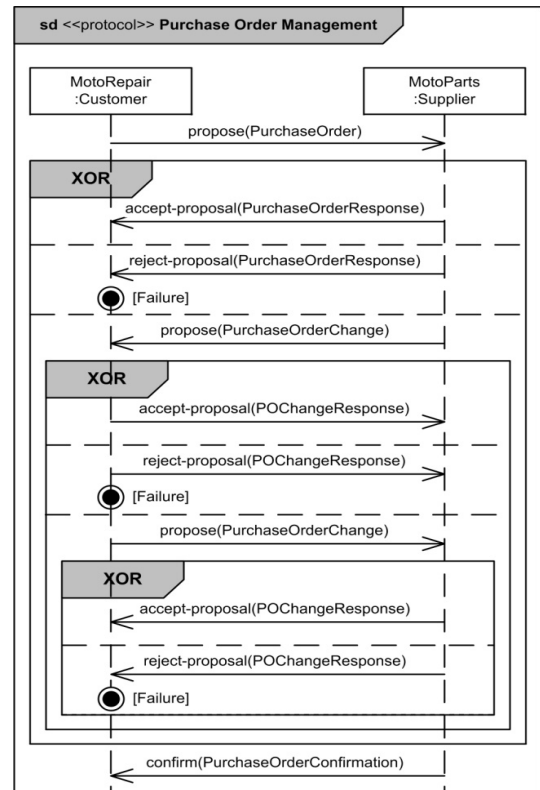


Figura 9. Proceso colaborativo "Gestión de Orden de Compra" definido con el lenguaje UP-ColBPIP mediante la vista de protocolos de interacción.

Cuando el agente MotoPartsPOM recibe el mensaje de aceptación, lo procesa y genera un documento de negocio confirmando la aceptación de la orden de compra propuesta. Este documento de negocio *PurchaseOrderConfirmation* es enviado como contenido de un mensaje *confirm* (interacción 22), con lo cual la ejecución del proceso colaborativo finaliza.

## VI. TRABAJOS RELACIONADOS

El desarrollo de soluciones tecnológicas para la ejecución de procesos colaborativos mediante plataformas basadas en agentes de software, utilizando en algunos casos los principios de MDD [32-34], se han estudiado previamente en algunos trabajos de investigación. En [32] se describe un enfoque para el diseño y ejecución de procesos colaborativos basados en una arquitectura de agentes de software, utilizando los principios de MDD. Los procesos privados son modelados en el nivel de negocio y en un nivel técnico se introduce un nivel

de abstracción adicional entre los procesos privados y los procesos colaborativos, llamado vista de procesos. Esta vista de procesos representa una interface para la interacción con otras organizaciones, describiendo las interacciones de uno o más procesos privados desde la perspectiva de una organización, en comparación con los procesos colaborativos que describen las interacciones desde una perspectiva neutral. En el nivel de ejecución los procesos colaborativos se extienden con información de la plataforma específica. El modelo del agente se construye a partir de un modelo de procesos generado en el nivel técnico que está basado en la arquitectura orientada a servicios (SOA, del inglés *Service-Oriented Architecture*), el cual contiene sólo la información necesaria para la interacción con uno o más partes de la colaboración. En esta propuesta el modelo del agente generado no tiene una orientación a procesos y los planes del agente son especificados en una estructura secuencial y estática.

```
<vertices xmlns:type="BPMN:Activity" id="sendTask_bml1" outgoingEdges="successorSequenceFlow1"
incomingEdges="sequenceFlow_bml1" name="Send Propose PurchaseOrder" activityType="EventIntermediateMessage">
  <eAnnotations id="sendTask_bml1annotation1" source="isThrowing">
    <details id="sendTask_bml1annotation1" key="isThrowing" value="true"/>
  </eAnnotations>
  <eAnnotations id="sendTask_bml1annotation2" source="jadx_parameters_table">
    <details id="sendTask_bml1annotation2" key="dimension" value="4:2"/>
    <details id="sendTask_bml1annotation2" key="uniqueColumnIndex" value="0"/>
    <details id="sendTask_bml1annotation2" key="complexColumns" value="false:false"/>
    <details id="sendTask_bml1annotation2" key="0:0" value="type"/>
    <details id="sendTask_bml1annotation2" key="0:1" value="Propose PurchaseOrder"/>
    <details id="sendTask_bml1annotation2" key="1:0" value="receivers"/>
    <details id="sendTask_bml1annotation2" key="1:1" value="MotoParts"/>
    <details id="sendTask_bml1annotation2" key="2:0" value="performative"/>
    <details id="sendTask_bml1annotation2" key="2:1" value="PROPOSE"/>
    <details id="sendTask_bml1annotation2" key="3:0" value="content"/>
    <details id="sendTask_bml1annotation2" key="3:1" value="PurchaseOrder"/>
  </eAnnotations>
</vertices>
```

Figura 10. Código generado para la actividad "Send Propose PurchaseOrder" del proceso ejecutable a partir de un proceso de integración.

En [33] se propone un método basado en la MDD para la generación de modelos de agentes de software con capacidad de ejecutar procesos colaborativos. Se generan las especificaciones de un modelo de agente ejecutable en una plataforma específica, que utiliza capacidades (*capabilities*) definidas como módulos que encapsulan una funcionalidad. Estas son definidas mediante el lenguaje WSDL (del inglés *Web Services Description Language*) para especificar invocaciones a sistemas internos de la organización. Estas capacidades son definidas en los planes del agente. El modelo del agente sólo contiene las interacciones de mensajes generadas a partir del proceso colaborativo y los planes del agente son especificados con invocaciones a otros servicios, delegando la ejecución de la tareas. Además, la etapa de la generación código del agente requiere de la intervención del diseñador de sistemas para asignar las capacidades requeridas por el agente para invocar a un servicio específico.

En [34] se describe una propuesta para el diseño de protocolos de interacción basado en un lenguaje de modelado independiente de la plataforma para el dominio de sistemas multi-agente, llamada DSML4MAS [35], y hace uso de los principios de MDD para la generación de código. En primer lugar, en una vista de interacción se define el protocolo de

interacción, utilizado para generar el comportamiento del agente. La vista de interacción permite definir las interacciones de los actores involucrados. Luego, el diseñador del sistema puede realizar un refinamiento a la descripción del comportamiento mediante una vista de comportamiento, agregando instancias del proceso privado e información adicional. Esta especificación del comportamiento se genera utilizando el meta-modelo DSML4MAS. Finalmente, a partir del comportamiento generado se realiza una transformación a código, siguiendo la estructura de un agente de la plataforma JACK [36]. En esta etapa, el código puede ser modificado manualmente, agregando el código necesario, para que el agente pueda ser ejecutado.

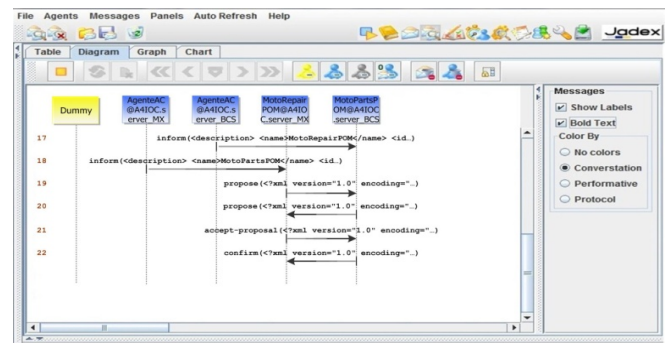


Figura 11. Ejecución del proceso colaborativo "Gestión de Orden de Compra" mediante la plataforma de agentes de software.

En [37] se propone una plataforma de agentes de software para la gestión de procesos inter-organizacionales utilizando Petri-Nets [38]. Los procesos son modelados mediante una vista de procesos, y posteriormente, son mapeados mediante reglas a Petri-Nets. La plataforma está compuesta de agentes locales en cada organización involucrada y un agente global, el cual tiene la función de mediador entre los agentes locales, así como de ejecutar el proceso inter-organizacional. Este enfoque requiere de un agente con una función de mediador que centraliza las interacciones entre los agentes responsables de ejecutar los procesos, limitando la autonomía de las organizaciones en ambientes de colaboraciones inter-organizacionales. En [39] se presenta una metodología que permite modelar procesos de negocio y generar las especificaciones para su ejecución en una plataforma de agentes. La metodología utiliza un enfoque basado en MDD que permite realizar las transformaciones necesarias mediante el Framework JIAC V [40]. El desarrollo del proceso inicia con el análisis mediante diagramas de casos de uso. Luego, para cada caso de uso se crea un diagrama de proceso basado en el lenguaje BPMN [19]. A partir de los diagramas de procesos y los diagramas de casos de uso, se derivan el rol de cada participante, el comportamiento y sus capacidades. En la etapa de diseño se puede realizar un refinamiento de los modelos mediante los editores de la herramienta. Finalmente, los modelos de organización (roles y agentes) de cada agente y el comportamiento de los agentes (planes, reglas y servicios) son integrados, lo cual permite generar el código del agente.

Las propuestas antes mencionadas ofrecen ventajas en el desarrollo de soluciones tecnológicas. Sin embargo, tienen como característica un enfoque tradicional en el diseño de los

agentes de software, en donde el comportamiento del agente que representa una organización en la ejecución de un proceso colaborativo, es definido en tiempo de diseño y no puede ser creado o modificado en tiempo de ejecución de la solución tecnológica. Esto implica que los procesos colaborativos y de integración a los que dan soporte los agentes, son acordados y definidos también en tiempo de diseño a través de acuerdos estáticos de colaboración. Por lo tanto, los agentes no pueden adaptarse en tiempo de ejecución a cambios en dichos procesos o ejecutar nuevos procesos. Lo anterior, representa la principal desventaja que limita la implementación de dichas propuestas en ambientes de colaboración inter-organizacionales dinámicos. En estos ambientes, se requiere una plataforma que permita adicionar nuevos comportamientos a los agentes o crear nuevos agentes para que puedan gestionar procesos colaborativos nuevos o rediseñados que las organizaciones acuerdan en forma dinámica, y comunicarse con otros agentes que representan a las organizaciones involucradas. Esto requiere que la generación de estos agentes se realice en tiempo de ejecución a través de la plataforma que los contiene.

## VII. CONCLUSIÓN

En este trabajo se propuso una solución tecnológica basada en agentes de software que permite dar soporte a la gestión de procesos colaborativos en ambientes de colaboración inter-organizacionales dinámicos. Por un lado, se propuso una plataforma de agentes de software que habilita a las organizaciones a: negociar acuerdos de colaboración en forma electrónica de los procesos colaborativos a ejecutar o nuevas versiones de los procesos acordados previamente; ejecutar en forma descentralizada los procesos colaborativos acordados; y generar los artefactos de implementación que permiten ejecutar los procesos colaborativos. En la plataforma se propusieron y definieron diferentes tipos de agentes, los cuales se clasifican en agentes estáticos y dinámicos. Los agentes estáticos posibilitan a las organizaciones entablar y gestionar colaboraciones dinámicas en forma electrónica. Los agentes dinámicos dan soporte a la gestión de procesos colaborativos que se acuerdan y definen ejecutar durante el transcurso de la colaboración. Estos agentes dinámicos son generados en tiempo de ejecución del acuerdo de colaboración a partir de los modelos conceptuales de procesos colaborativos acordados.

Por otro lado, se propuso una metodología de desarrollo, basada en los principios del desarrollo dirigido por modelos (MDD), que posibilita generar los artefactos de implementación (modelo de proceso ejecutable y código de agentes orientados a procesos) a partir de modelos conceptuales de procesos colaborativos y de integración. Esta metodología y sus métodos son implementados y automatizados por agentes de software provistos en la plataforma, los que habilitan la generación de dichos artefactos en tiempo de ejecución de la plataforma, posibilitando de esta manera que la plataforma soporte el aspecto dinámico de las colaboraciones inter-organizacionales, y permitiendo que los sistemas de las organizaciones se adapten rápidamente en tiempo de ejecución para gestionar nuevos procesos colaborativos o nuevas versiones de dichos procesos.

La generación automática (en tiempo de ejecución de la plataforma) de los artefactos de implementación requeridos por las organizaciones, para dar soporte a la ejecución de procesos colaborativos, conlleva varios beneficios: 1) disminución de los tiempos y costos de desarrollo, comparado con el desarrollo tradicional de sistemas; 2) permite prevenir errores en la etapa de diseño y de desarrollo de los artefactos de implementación; 3) posibilita que los sistemas (agentes) de las organizaciones, funcionen de acuerdo a lo especificado en los modelos conceptuales de procesos colaborativos, definidos en un nivel de negocio, garantizando la alineación de la solución de negocio con la solución tecnológica; 4) posibilita que los sistemas de las organizaciones sean flexibles y se adapten rápidamente a cambios en los requerimientos organizacionales o de negocio de las colaboraciones inter-organizacionales, que implican ejecutar nuevos procesos colaborativos o nuevas versiones de los procesos acordados.

La solución tecnológica propuesta es consistente con el enfoque de integración entre organizaciones orientado a procesos, en donde la colaboración (incluyendo la coordinación de actividades para la toma de decisiones en forma conjunta y el intercambio de información entre organizaciones) es definida, ejecutada y gestionada en términos de procesos colaborativos.

## REFERENCIAS

- [1] C.M. Chituc, A. Azevedo, and C. Toscano. A framework proposal for seamless interoperability in a collaborative networked environment. *Computers in Industry*, 60(5), pp. 317-338, 2009.
- [2] P.D. Villarreal, E. Salomone, and O. Chiotti, "Modeling and Specifications of Collaborative Business Processes using MDA Approach and a UML Profile", in *Enterprise Modeling and Computing with UML*, P. Rittgen, Ed. USA: Idea Group Inc., 2007, pp. 13-45.
- [3] M. Weske. *Business Process Management. Concepts, Languages, Architectures*. Berlin: Springer, 2007.
- [4] S. Roser, and B. Bauer, "A Categorization of Collaborative Business Process Modelling Techniques", in *Proceedings of the 7th IEEE International Conference on E-Commerce Technology Workshops*, IEEE Computer Society, pp. 43-54, 2005.
- [5] I.M. Lazarte, E. Tello-Leal, J. Roa, O. Chiotti, and P.D. Villarreal, "Model-Driven Development Methodology for B2B Collaborations", in *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW-2010)*, IEEE Computer Society, pp. 69-78, 2010.
- [6] M. Dumas, W. van der Aalst, and A. ter Hofstede, *Process-Aware Information Systems: Bridging People and Software through Process Technology*, New Jersey USA: John Wiley & Sons, Inc., 2005.
- [7] F. A. Zorzan, and D. Riesco. Transformation in QVT of Software Development Process based on SPEM to Workflows. *IEEE Latin America Transactions*, 6(7), pp. 655-660, 2008.
- [8] L.M. Camarinha-Matos, H. Afsarmanesh, N. Galeano, and A. Molina. Collaborative Network Organizations - Concepts Practice in Manufacturing Enterprises. *Computers and Industrial Engineering*, 57(1), pp. 46-60, 2009.
- [9] E. Tello-Leal, O. Chiotti, and P.D. Villarreal, "An Agent-based B2B Collaboration Platform for Executing Collaborative Business Process", in W. Cellary and E. Estevez (Eds.), *Software Services for e-World*, Berlin, Germany: Springer Heidelberg, pp. 40-50, 2010.
- [10] N. Mehandjiev, and P. Grefen. *Dynamic Business Process Formation for Instant Virtual Enterprises*. London: Springer, 2010.
- [11] P. Grefen, N. Mehandjiev, G. Kouvas, G. Weichhart, and R. Eshuis. Dynamic Business Network Process Management in Instant Virtual Enterprises. *Computers in Industry*, 60(2), pp. 86-103, 2009.
- [12] N. Meyer, T. Feiner, M. Radmayr, D. Blei, and A. Fleischmann, "Dynamic Catenation and Execution of Cross Organisational Business Processes - The jCPEX! Approach", in A. Fleischmann, W. Schmidt, R. Singer, and D. Seese (Eds.), *Subject-Oriented Business Process Management*, Berlin, Germany: Springer, pp. 84-105, 2011.



- [13] B. Selic, The Pragmatics of Model-Driven Development, *IEEE Software*, 20(5), pp. 19-25, 2003.
- [14] T. Stahl, and M. Volter, *Model-Driven Software Development: Technology, Engineering, Management*. USA: Wiley, 2006.
- [15] E. Tello-Leal, O. Chiotti, and P.D. Villarreal. Process-Oriented Integration and Coordination of Healthcare Services across Organizational Boundaries, *Journal of Medical Systems*, 36(6), pp. 3713-3724, 2012.
- [16] OMG-MDA, "Model Driven Architecture - MDA Guide Version 1.0.1", Specification omg/2003-06-01, Object Management Group (OMG), 2003. Available: <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- [17] A.W. Brown, J. Conallen, and D. Tropeano, "Introduction: Models, Modeling, and Model-Driven Architecture (MDA)", in S. Beydeda, M. Book, and V. Gruhn (Eds.), *Model-Driven Software Development*, Berlin, Germany: Springer Heidelberg, pp. 1-16, 2005.
- [18] J. Duarte, M. González, L. Cernuzzi, and O. Pastor. IDEAS03: Evaluation of software development through an MDA tool: a case study, *IEEE Latin America Transactions*, 6(3), pp. 252-259, 2008.
- [19] OMG-BPMN, "Business Process Model and Notation version 2.0", Specification formal/2011-01-03, Object Management Group (OMG), 2011. Available: <http://www.omg.org/spec/BPMN/2.0/PDF/>
- [20] I.M. Lazarte, O. Chiotti, P.D. Villarreal, L.E. Thom, and C. Iochpe, "An MDA-based Method for Designing Integration Process Models in B2B Collaborations", in *Proceedings of the 13th International Conference on Enterprise Information Systems (ICEIS 2011)*, SciTePress, Beijing, China, pp. 55-65, 2011.
- [21] P.D. Villarreal, I. Lazarte, J. Roa, and O. Chiotti, "A Modeling Approach for Collaborative Business Processes based on the UP-ColBPIP", in S. Rinderle-Ma, S. Sadiq, and F. Leymann (Eds.), *Business Process Management Workshops*, Berlin, Germany: Springer Berlin Heidelberg, pp. 318-329, 2010.
- [22] L.H. Thom, M. Reichert, and C. Iochpe, Activity patterns in process-aware information systems: basic concepts and empirical evidence, *International Journal of Business Process Integration and Management*, 4(2), pp. 93-110, 2009.
- [23] FIPA, "Agent Communication specifications deal with Agent Communication Language (ACL)", Specification SC00061, Foundation for Intelligent Physical Agents, 2002. Available: <http://www.fipa.org/repository/ac1specs.html>
- [24] Eclipse, "Eclipse Platform". [Online]. Accessed in 09/09/2012. Available: <http://www.eclipse.org/>
- [25] F. Jouault, F. Allilaire, J. Bézuvin, and I. Kurtev, ATL: A model transformation tool, *Science of Computer Programming*, 72(1-2), pp. 31-39, 2008.
- [26] ATL - ATLAS Transformation Language, "ATL - A Model Transformation Technology", Specification release 3.2.0, OBEO and AtlanMod, [Online]. Accessed in 08/24/2012. Available: <http://www.eclipse.org/at1/>
- [27] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks. *EMF Eclipse Modeling Framework*, 2nd ed., Boston, MA, USA: Addison Wesley/Pearson, 2011.
- [28] A. Pokahr, L. Braubach, and K. Jander, "Unifying Agent and Component Concepts", in J. Dix and C. Witteveen (Eds.), *Multiagent System Technologies*, Berlin, Germany: Springer Berlin Heidelberg, pp. 100-112, 2010.
- [29] A.S. Rao, and M.P. Georgeff, "BDI Agents: From Theory to Practice", in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 312-319, 1995.
- [30] A. Pokahr, L. Braubach, and W. Lamersdorf, "Jadex: A BDI Reasoning Engine", in R.H. Bordini, M. Dastani, J. Dix, and A.F. Seghrouchni (Eds.), *Multi-Agent Programming*, USA: Springer, pp. 149-174, 2005.
- [31] A. Pokahr, and L. Braubach, "The Notions of Application, Spaces and Agents - New Concepts for Constructing Agent Applications", in M. Schumann, L. Kolbe, M. Breitner and A. Frerichs (Eds.), *Multikonferenz Wirtschaftsinformatik (MKWI): Multi-agent Systems: Decentral approaches for designing, organizing, and operating information systems*, pp. 159-160, 2010.
- [32] T. Kahl, I. Zinnikus, S. Roser, C. Hahn, J. Ziemann, J.P. Müller, and F. Klaus. "Architecture for the Design and Agent-Based Implementation of Cross-Organizational Business Processes", in R.J. Gonçalves, J.P. Müller, K. Mertins, and M. Zelm (Eds.), *Enterprise Interoperability II*, London: Springer, pp. 207-218, 2007.
- [33] I. Zinnikus, C. Hahn, and F. Klaus, "A model-driven, agent-based approach for the integration of services into a collaborative business process", in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Volume 1 (AAMAS 2008), International Foundation for Autonomous Agents and Multiagent Systems, Estoril, Portugal, pp. 241-248, 2008.
- [34] C. Hahn, I. Zinnikus, S. Warwas, and F. Klaus, "Automatic Generation of Executable Behavior: A Protocol-Driven Approach", in M.-P. Gleizes, and J.J. Gomez-Sanz (Eds.), *Agent-Oriented Software Engineering X*, Lecture Notes in Computer Science, Berlin : Springer, pp. 110-124, 2011.
- [35] C. Hahn, "A domain specific modeling language for multiagent systems", in *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*, pp. 233-240, 2008.
- [36] N. Howden, R. Rönquist, A. Hodgson, and A. Lucas, "JACK Intelligent Agents - Summary of an Agent Infrastructure", in *Proceedings of the 5th ACM International conference on autonomous agents*, 2001.
- [37] P. Jiang, X. Shao, L. Gao, and Z. Yang, "A Multi-agent System for Cross-Organizational Workflows Management Based on Process-View", in Y. Shi, G.D. Albada, J. Dongarra, and P. M.A. Sloot (Eds.), *Computational Science*, Lecture Notes in Computer Science, Berlin: Springer, pp. 212-215, 2007.
- [38] C. Girault, and V. Rüdiger, *Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications*. Germany: Springer-Verlag, 2002.
- [39] T. Küster, M. Lützenberger, A. Hessler, and B. Hirsch, Integrating Process Modelling into Multi-Agent System Engineering, *Multiagent and Grid Systems*, 8(1), pp. 105-124, 2012.
- [40] B. Hirsch, T. Konnerth, and A. Hessler, "Merging Agents and Services - the JIAC Agent Platform", in A. El Fallah Seghrouchni, J. Dix, M. Dastani, and R.H. Bordini (Eds.), *Multi-Agent Programming*, USA: Springer, pp. 159-185, 2009.



**Edgar Tello-Leal.** Es Licenciado en Computación Administrativa (TI) por la Universidad Autónoma de Tamaulipas, México y recibió el grado de Doctor en Ingeniería en Sistemas de Información por la Universidad Tecnológica Nacional de la República Argentina en Noviembre de 2012. Actualmente trabaja como Profesor de Tiempo Completo e Investigador en la Universidad Autónoma de Tamaulipas, México. Las líneas de investigación y generación del conocimiento actuales son gestión de procesos de negocio, desarrollo dirigido por modelos, sistemas de información basados en agentes de software, y sistemas basados en conocimiento.



**Omar Chiotti.** Recibió el grado de Doctor por la Universidad Nacional del Litoral (UNL) de Argentina en 1989. Es profesor de Ingeniería de Sistemas de Información en la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional en Argentina desde 1986 y ha estado trabajando para el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) de la República Argentina como investigador desde 1991. Actualmente, es director del Centro de Investigación y Desarrollo de Ingeniería en Sistemas de Información (CIDISI). Sus líneas de investigación actuales incluyen e-colaboración, gestión del conocimiento y sistemas multi-agente.



**Pablo David Villarreal.** Es Doctor en Ingeniería de Sistemas de Información (2005) por la Universidad Tecnológica Nacional (UTN) de Argentina. Es profesor asistente de tiempo completo en la UTN e investigador asistente del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) de la República Argentina. Trabaja en el CIDISI desde el año 2000. En el año 2007 realizó una estancia posdoctoral en la Universidad Politécnica de Valencia, España. Su investigación actual incluye la gestión de procesos de negocio, gestión de workflows, arquitectura orientada a servicios, desarrollo dirigido por modelos, sistemas B2B, gestión de la cadena de suministro y sistemas multi-agente.