

# Modelado del Flujo de Control en Tareas Colaborativas utilizando OWL-DL

Manuel Noguera, María V. Hurtado,  
María Luisa Rodríguez, José Luis Garrido,  
José Ramón Polo  
Dept. de Lenguajes y Sistemas Informáticos  
ETS Ingeniería Informática y Telecomunicación  
Univ. de Granada  
18071 Granada  
{mnoquera, mhurtado, mlra, jgarrido, jrpl}@ugr.es

Lawrence Chung  
Dpt. of Computer Science  
The Erik Jonsson School of Engineering &  
Computer Science  
Univ. of Texas at Dallas  
Richardson, Texas 75083, USA  
chung@utdallas.edu

## Resumen

El uso de ontologías y tecnologías relacionadas se está mostrando como un medio adecuado para el modelado de sistemas. Los lenguajes de la Web Semántica para describir ontologías como el *Web Ontology Language* (OWL), están siendo ampliamente utilizados para describir las entidades de un sistema y sus relaciones. Los elementos clásicos de los diagramas de clases de UML, tales como clases y asociaciones, son fácilmente trasladables a clases y propiedades en OWL. Tal correspondencia entre clases resulta útil para tratar de aprovechar las ventajas de ambos lenguajes al modelar un sistema. Mientras los modelos descritos en UML son adecuados para discutir la especificación de un sistema con los *stakeholders*, los descritos en OWL, aunque más difíciles de interpretar por las personas, son más adecuados para su procesamiento por computadoras. Sin embargo, no ocurre la misma facilidad de traducción al tratar de especificar en OWL la información de los diagramas de comportamiento de UML. Este artículo propone un modo concreto de abordar este problema para especificar procesos de negocio colaborativos en ontologías de modelos de sistema.

## 1. Introducción

Con frecuencia, el modelado de organizaciones abarca procesos de negocio en los que participan equipos de personas de dentro y fuera de la misma organización [6].

El modelado de los procesos colaborativos entre ellas es similar independientemente de las actividades propias de cada organización. Un factor clave en este proceso es que, tanto los modelos como su semántica, puedan ser compartidos fácilmente [6].

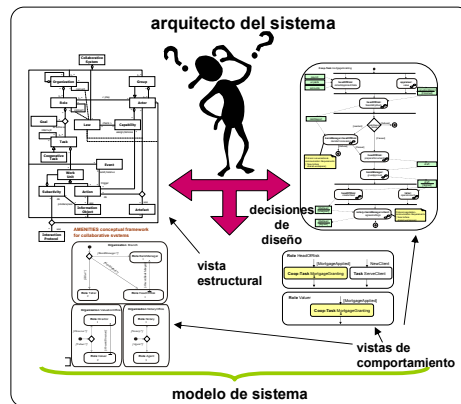


Figura 1. Vistas fraccionadas de una arquitectura de sistema

AMENITIES [3] es una metodología para diseñar sistemas colaborativos que extiende y hace uso de UML. Esta metodología propone el uso de diferentes modelos para comprender el funcionamiento de un sistema desde distintas perspectivas. La figura 1 ilustra esta situación.

Estos modelos son apropiados para discutir sobre ellos con los *stakeholders*, en tanto que son intuitivos y flexibles en cuanto a sintaxis, pero

tienen el inconveniente de presentar la información relativa a decisiones de diseño y activos de la empresa de forma dispersa [8], y también carecen de una semántica formal.

En este sentido, las ontologías se están convirtiendo en una tecnología apropiada para describir formalmente sistemas, de tal manera que se puedan llevar a cabo de forma automática procesos de razonamiento y de control de consistencia sobre los modelos del sistema. La publicación como *Recommendation* de lenguajes como *Resource Description Framework-Schema* (RDF-Schema) [14], y OWL [12], por parte del *World Wide Web Consortium* (W3C), también ha contribuido a fomentar su uso en otros ámbitos distintos del de la Web Semántica [9][10]. Por otro lado, las descripciones en este tipo de lenguajes son bastante prolijas e inapropiadas para comunicarse con los *stakeholders* [12].

Una correspondencia desde modelos semi-formales basados en UML hacia modelos formales basados en ontologías y viceversa, permitiría utilizar ambos lenguajes para el modelado y aprovechar las ventajas de ambas descripciones. Algunos modelos propuestos por AMENITIES basados en UML (por ejemplo, el marco de trabajo conceptual recogido en un diagrama de clases), son fácilmente traducibles a una ontología en OWL. Sin embargo, con otros modelos, como los diagramas de comportamiento, este paso no es trivial, ya que, OWL no proporciona constructores nativos para modelar secuencias de elementos ordenados [11]. En este artículo se presenta una forma concreta de realizar esta correspondencia. Como caso de estudio se ha utilizado un proceso de flujo de trabajo para la concesión de una hipoteca.

El resto de este artículo está organizado como sigue: la sección 2 propone un modo de traducir diagramas de actividades de UML a OWL; la sección 3 describe un caso de estudio en el que se ha puesto en práctica la propuesta; en la sección 4 concluimos y presentamos trabajos para el futuro.

## 2. Modelos basados en ontologías para sistemas colaborativos

La disciplina *Computer-Supported Cooperative Work* (CSCW) estudia y analiza mecanismos de coordinación, para que la comunicación y colaboración entre personas sea efectiva, así como

los sistemas utilizados para ello. La complejidad inherente de los sistemas CSCW requiere grandes esfuerzos en su especificación y desarrollo [2]. En este proceso es habitual que analistas y *stakeholders* hagan uso de diferentes modelos que recojan la lógica de negocio que rige las relaciones entre los distintos activos de una organización: un modelo del dominio para el entorno colaborativo y un modelo colaborativo que capture los procesos de negocio que impliquen una colaboración. Todo ello configura la arquitectura de sistema de la organización para el sistema colaborativo de que se trate. En este trabajo se ha optado por seguir el enfoque propuesto en AMENITIES [3], una metodología para el estudio y análisis de sistemas colaborativos. El punto de partida es el marco conceptual del dominio.

### 2.1. Marco conceptual de dominio para sistemas colaborativos

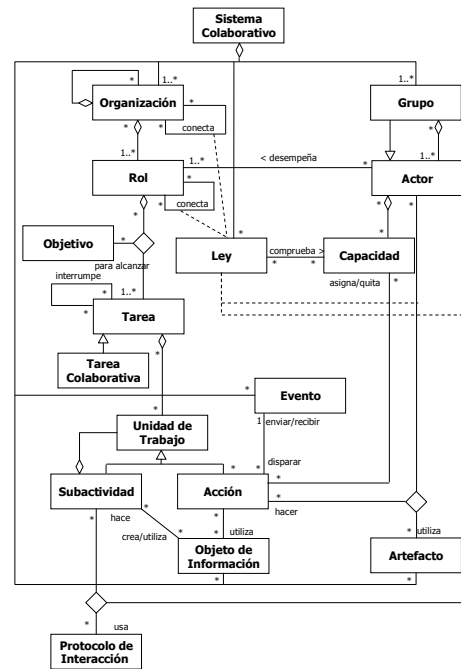


Figura 2. Marco conceptual propuesto en AMENITIES

La metodología AMENITIES propone un marco conceptual para describir las principales entidades (y sus relaciones), que habitualmente aparecen en un sistema colaborativo. Dicho marco conceptual se muestra en la figura 2 en forma de diagrama de clases UML.

Conforme a este marco conceptual, una *acción* es una unidad atómica de trabajo. Su ejecución dirigida por eventos puede requerir/modificar/generar información explícita. Una *subactividad* es un conjunto de otras subactividades y/o acciones. Una *tarea* es un conjunto de *subactividades* pensadas para alcanzar ciertos *objetivos*. Un *rol* es un identificador para un conjunto de tareas relacionadas a realizar. Un *actor* es un grupo, usuario, programa o entidad con ciertas capacidades adquiridas (destrezas, categoría, etc.), que puede desempeñar un rol al ejecutar (usando *artefactos*), o ser responsable de, acciones. Los *grupos* realizan ciertas subactividades en función de ciertos *protocolos de interacción*. Un grupo es un conjunto de actores organizados en torno a una tarea y puede estar compuesto de otros subgrupos. Una *tarea cooperativa* es aquella que debe realizarse por más de un actor, desempeñando el mismo o diferentes roles. Una *ley* es una limitación o restricción del sistema que permite ajustar dinámicamente el conjunto de comportamientos posibles. Una *organización* consiste en un conjunto de roles relacionados. Finalmente, un *sistema colaborativo* consta de grupos, organizaciones, leyes, eventos y artefactos.

## 2.2. Conceptos sobre ontologías y versiones de OWL

Básicamente, una ontología puede verse como un conjunto de clases o conceptos, relaciones entre ellos y restricciones que limitan su conjunto de posibles interpretaciones [5]. OWL es un lenguaje para describir ontologías pensado para la Web Semántica, pero también está siendo utilizado para describir ontologías en muchos otros ámbitos. Las traducciones de modelos en UML (como el mostrado en la figura 2), a modelos en OWL puede lograrse de forma sencilla simplemente mapeando *clases UML*, a *conceptos OWL*, y *asociaciones UML* a *propiedades* (relaciones) *OWL*. El resultado es una descripción del dominio del sistema procesable por computadoras que

posibilita llevar a cabo procesos de razonamiento y control de consistencia.

La especificación de OWL provee tres versiones del lenguaje –*Lite*, *DL* y *Full*–, que difieren en su poder expresivo. La versión *DL* proporciona el máximo poder expresivo conservando siempre la computabilidad del modelo para razonadores basados en Description Logics [1]. Esta propiedad es necesaria para que una ontología pueda ser procesable por razonadores automáticos basados en OWL-DL y realizar controles de consistencia. En efecto, cada versión de OWL se basa en RDF, que a su vez provee un vocabulario predefinido que puede hacer que una ontología *Lite* o *DL*, pase a ser OWL-Full [13]. De ahora en adelante, las apariciones en el texto al lenguaje OWL se referirán a su versión OWL-DL.

En [4] se describe una traducción del marco conceptual propuesto en AMENITIES (ver figura 2), a una ontología OWL-DL. En lo que sigue, se muestra la forma de complementar este marco conceptual para poder representar secuencias de procesos de negocio.

## 2.3. Marco conceptual para ontologías de procesos colaborativos

El marco conceptual de la figura 2 es útil como metamodelo contra el que validar modelos colaborativos construidos según AMENITIES. Dicho metamodelo define una *tarea* como un conjunto de *unidades de trabajo*. Esto bastaría para modelar la estructura de un sistema colaborativo, pero en lo que se refiere al modelado de procesos de negocio, las mencionadas *unidades de trabajo* deben disponerse ordenadas en una *secuencia*. Tal es el propósito de los diagramas de actividades de UML y con este fin son utilizados en AMENITIES.

En OWL, las entidades de un dominio se describen en términos de clases de recursos y propiedades. RDF provee el constructor *rdf:List* para implementar secuencias de elementos ordenados, pero su uso convierte la ontología en OWL Full, con la consiguiente pérdida de garantías computacionales para el razonamiento, y no ofrece otro soporte para la descripción de secuencias, que ha de realizarse por medio de soluciones alternativas [11].



## 2.4. Conexión entre conceptos

El conjunto de clases definido en la figura 3 es sólo una extensión a los conceptos propuestos en la sección 2.1. Estas nuevas clases están pensadas para que los procesos colaborativos puedan ser tratados convenientemente en una descripción ontológica “de fondo” para un sistema colaborativo.


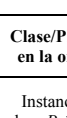


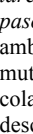
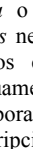
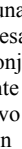

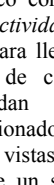

Elemento de diagrama de actividad UML	Símbolo	Clase/Propiedad en la ontología
Nodo inicial		Instancia de la clase <i>Primer_paso</i>
Actividad y roles-actores		Clase <i>Unidad de trabajo</i> (una <i>Subactividad</i> y o <i>Acción</i> ) Clase <i>Rol</i> y Clase <i>Actor</i>
Decisión		Clase <i>Decisión</i>
Nota		Clase <i>Protocolo de Interacción</i>
Señal		Clase <i>Evento</i>
Evento		Clase <i>Evento</i>
Fork/Join		Clase <i>Punto de Sincronización</i>
Flujo		Propiedad <i>seguido_por</i> de la Clase <i>Paso</i>
Objeto		Clase <i>Objeto de Información</i>
Final Nodo		Instancia de la clase <i>Paso_Final</i>

Tabla 1. Correspondencias entre elementos de los diagramas de actividades en UML y entidades en la ontología para sistemas colaborativos

Por esta razón, dichas clases únicamente definen un “esqueleto” con aquellos conceptos y relaciones necesarios para modelar secuencias de actividades en un proceso colaborativo (es decir,

*secuencia, primer paso, paso, paso final, decisión, punto de sincronización, descrita\_en, seguida\_por, ejecuta, es parte\_de y es\_un*), y hace uso de otras clases y propiedades ya definidas en el marco conceptual de la figura 2 (*tarea, unidad de trabajo, subactividad*, y otras relaciones *es\_un* y *es parte\_de*).

La relación *descrita\_en* enlaza las nuevas clases con el marco conceptual conectando una *tarea* o una *subactividad* con la secuencia de *pasos* necesarios para llevarla a cabo. De hecho, ambos conjuntos de clases se complementan mutuamente y dan lugar un metamodelo colaborativo cohesionado que ayuda a mejorar la descripción de vistas estructurales y de comportamiento de un sistema colaborativo. La figura 4 muestra una representación gráfica obtenida a partir de una descripción en OWL de las clases del dominio. La clase *Unidad de trabajo* aparece dos veces para no enmarañar el diagrama. Por la misma razón se han ocultado algunas relaciones, tales como las relaciones *es\_un*.

*Rol, Objeto de información, Protocolo de interacción* y *Evento* son otros tipos de elementos compartidos por ambos metamodelos. La tabla 1 recoge resumidamente los elementos de un diagrama de actividad y su correspondiente elemento en la ontología.

## 3. Caso de estudio: concesión de hipotecas

Se ha diseñado un ejemplo práctico para mostrar una aplicación de la propuesta. El diagrama de actividades de la figura 5 describe el proceso de negocio seguido para la concesión de una hipoteca que un cliente solicita en una sucursal bancaria. Este proceso lleva consigo la participación colaborativa de miembros de diferentes organizaciones: una *sucursal*, una *oficina de tasación* y una *notaría*. El proceso se inicia con dos subactividades: *recabarDatosDelCliente* y *tasar*, a llevar a cabo por el *responsableDeRiesgos* de la *sucursal* y un *tasador* de la *oficina de tasación*, respectivamente. Dichas subactividades se realizarán en paralelo. La tarea colaborativa *concesiónDeHipoteca* finaliza con la firma de la *escritura* por parte del *notario*, el *director* de la *sucursal* y el *cliente*.

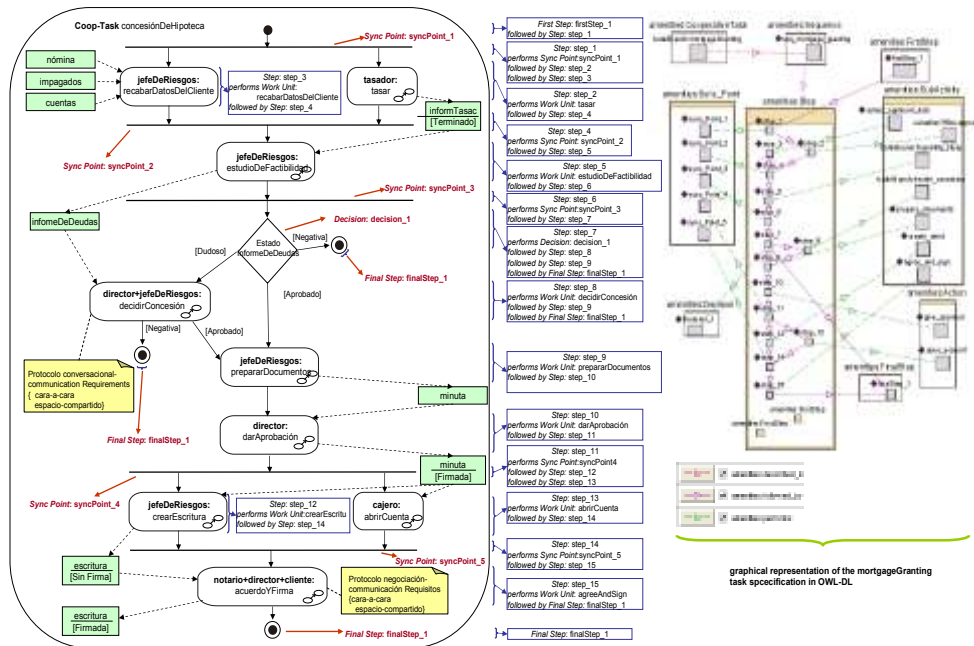


Figura 5. Proceso de negocio para *ConcesiónDeHipoteca* descrita en un diagrama de actividad de UML y utilizando ontologías

En la parte central de la figura 5, se ha tratado de representar la descripción ontológica para cada etapa por la que discurre la tarea *concesiónDeHipoteca*. Se ha utilizado un lenguaje *ad-hoc* menos prolijo, en lugar de OWL-RDF. Las cajas y paréntesis azules resaltan las instancias utilizadas. En la parte derecha de la figura 5 se muestra la representación gráfica de la tarea a partir de su especificación en OWL. Los nombres identificativos dados a cada paso de la secuencia (por ejemplo, *firstStep\_1*, *step\_1*, *step\_2*, etc.), son irrelevantes, aunque se ha tratado de asignarlos utilizando sufijos incrementales para facilitar su seguimiento por el diagrama. El orden potencial de cada paso en la secuencia viene determinado por los pasos a los que sucede y los pasos por los que puede estar seguido. El orden final en que se ejecuta cada paso en una secuencia depende de cómo vaya transcurriendo finalmente de la ejecución de la tarea a la que pertenece.

Lo mismo es aplicable a los nombres otorgados a los puntos de sincronización y decisiones. Hemos utilizado el color rojo para señalar algunos de estos elementos del marco conceptual para procesos colaborativos, pero que no forman parte del diagrama de actividad, como por ejemplo, *syncPoint\_6* y *decision\_1*.

Los nombres de las unidades de trabajo (*crearEscritura*, *abrirCuenta*, *acuerdoYFirma*, etc.), sí son relevantes y deben corresponderse con instancias de actividades y acciones del marco de trabajo (sección 2.1).

#### 4. Conclusiones y trabajos futuros

En el proceso de modelado de sistemas colaborativos, algunas decisiones tomadas en su diseño (por ejemplo, las relativas a qué grupos, actores, roles, artefactos, etc., hay en el sistema), pueden verse reflejadas en modelos estructurales del sistema tales como diagramas de clases,

mientras otras pueden recogerse en modelos de comportamiento como diagramas de actividades (por ejemplo, actividades, tareas, etc.). Es deseable que las mencionadas decisiones de diseño, aunque esparcidas entre los diferentes modelos del sistema, no lleven a descripciones del sistema inconsistentes entre sí. Igualmente, es deseable que la semántica de estos modelos pueda ser compartida por las organizaciones que trabajan colaborativamente. Las ontologías y las tecnologías relacionadas como el lenguaje OWL están demostrando satisfacer ambos requisitos. Sin embargo, mientras que la traducción de modelos estructurales (como los diagramas de clases), a ontologías es sencilla, la traducción de modelos de comportamiento no es tan simple. En estas páginas se ha mostrado un ejemplo de cómo abordar este problema.

Actualmente, se está investigando acerca de las posibilidades que ofrecen los distintos razonadores basados en Description Logics que existen para el lenguaje OWL. El objetivo es desarrollar una herramienta de soporte al diseño de sistemas colaborativos que provea al diseñador de cierta guía para obtener descripciones consistentes del sistema. Muchas limitaciones se derivan de las propiedades de computabilidad y decidibilidad que han de satisfacer las descripciones en este lenguaje. Asimismo, se está trabajando en la definición de relaciones que permitan especificar, obtener e inferir información más elaborada acerca del orden que ocupan o que pueden ocupar, las subactividades y acciones de las que consta una tarea en el marco conceptual propuesto.

Finalmente, señalar que actualmente el lenguaje OWL se encuentra inmerso en un amplio proceso de revisión, producto de los comentarios que han ido proporcionando tras usarlo las distintas comunidades de diseñadores y desarrolladores de sistemas. Se espera que este proceso de revisión resulte en una versión mejorada del lenguaje que facilite la posibilidad de especificar listas ordenadas de elementos en este lenguaje, así como constructores para la definición de relaciones que impliquen a más de dos clases, y de esta forma el mapeo entre los lenguajes UML y OWL sea más directo.

## Agradecimientos

Este trabajo ha sido financiado por el proyecto CICYT TIN2004-08000-C03-02 del Ministerio de Educación y Ciencia de España.

## Referencias

- [1] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., *The Description Logic Handbook*, Cambridge University Press, 2003
- [2] Carstensen, P.H., Schmidt, K., "Computer supported cooperative work: new challenges to systems design", *Handbook of Human Factors/Ergonomics*, Asakura Publishing, Tokyo, 2003, pp. 619-636
- [3] Garrido, J.L., Gea, M., Rodríguez, M.L., "Requirements Engineering in Cooperative Systems", *Requirements Engineering for Sociotechnical Systems*, Chapter XIV, IDEA GROUP, Inc.USA, 2005
- [4] Garrido, J.L., Noguera, M., González, M., Hurtado, M.V., Rodríguez, M.L., "Definition and use of Computation Independent Models in an MDA-based Groupware Development Process", *Science of Computer Programming* 66, 2007, pp. 25-43
- [5] Guarino, N., "Formal ontology and information systems", *Proceedings of FOIS '98*, IOS Press, Amsterdam, 1998, pp. 3-15
- [6] Jaekel, F.W., Perry, N., Campos, C., Mertins, K., Chalmers, R., "Interoperability Supported by Enterprise Modelling", *LNCS 3762*, Springer-Verlag, Berlin, 2005, pp. 552-561
- [7] Jonkers, H., Lankhorst, M.M., van Buuren, R., Hoppenbrouwers, S., Bonsangue, M.M., van der Torre, L.W.N., "Concepts For Modeling Enterprise Architectures", *Int. J. Cooperative Inf. Syst.* 13, 2004, pp. 257-287
- [8] Lange, C.F.J., Chaudron, M.R.V., Muskens, J., "In Practice: UML Software Architecture and Design Description", *IEEE Software* 23, IEEE, 2006, pp. 40-46
- [9] Lera I., Juiz, C., Puigjaner, R., "Performance-related Ontologies and Semantic Web Applications for On-line Performance assessment of Intelligent Systems", *Science of Computer Programming* 61, Elsevier, 2006, pp. 27-37

- [10] Pahl, C., "Layered Ontological Modelling for Web Service-Oriented Model-Driven Architecture", LNCS 3748, Springer-Verlag, Berlin, 2005, pp. 88-102
- [11] W3C, "Defining N-ary Relations on the Semantic Web", W3C Working Group Note, 12 April 2006, <http://www.w3.org/TR/swbp-n-aryRelations/>
- [12] W3C, "OWL Web Ontology Language Guide", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/owl-guide/>
- [13] W3C, "OWL Web Ontology Language Semantics and Abstract Syntax", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/owl-guide/>
- [14] W3C, "RDF Vocabulary Description Language 1.0: RDF Schema", W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/rdf-schema/>