

# Un modelo comprensivo para el diseño de procesos en una Aplicación E-Learning

**Alejandro R. Sartorio**

Departamento de Ciencias de La Computación, Universidad Nacional de Rosario (UNR)  
Rosario (2000), Argentina  
Centro Regional de Investigación y Desarrollo Rosario (CERIDER)  
Rosario (2000), Argentina  
sartorio@fceia.unr.edu.ar

## Abstract

This paper describes an approach to documenting the conceptual design of E-Learning Web Transactions using UWATc+. A E-Learning Web Transaction is a collection of activities, which are formed by contracts context-aware (as software component), are that contributes to achieving a user-oriented pedagogical objective using a E-learning-based application. UWATc+ is a meta-model for describing the various aspects of a Web Transaction formed by contracts. It is an extension of the Transaction Design Model that is part of the Ubiquitous Web Applications (UWA) framework, a comprehensive framework for designing ubiquitous Web applications. A series of (extended) UML diagrams are used to graphically document the UWATc+ meta-model, which greatly facilitates adoption of the approach by practicing software engineers, expert in education, and programmers. Use of the approach for documenting Web Transaction is described through a case of study belonging to Obra Abierta project (CONICET - UNR).

## Resumen

Este trabajo presenta UWATc+, una modelo conceptual y de diseño para la documentación de Transacciones Web E-Learning. Una Transacción Web E-Learning es una colección de actividades formadas por contratos con características context-aware (como componente de software), que permiten alcanzar un objetivo pedagógico orientado al usuario de Aplicaciones Web E-Learning. UWATc+ es un meta-modelo para la descripción de diferentes aspectos de una Transacción Web conformada con contratos. Esto es una extensión del Modelo de Diseño de Transacciones perteneciente al framework UWA (Ubiquitous Web Applications). Para documentar gráficamente el modelo UWATc+ fueron usados (extendido) diagramas UML, los cuales facilitaron su adopción por parte de los ingenieros en sistemas, expertos en educación y programadores. El uso de esta propuesta de documentación de Transacciones Web será descrito a través de un caso de uso perteneciente al proyecto Obra Abierta (CONICET-UNR)

## 1 INTRODUCCIÓN

A medida que el avance en la investigación y desarrollo de plataformas e-learning brinden mejoras e innovaciones de herramientas (videoconferencias, portfolios, wikis, workshops, etc.) y sus respectivos servicios, crece la cantidad de posibles configuraciones de los espacios e-learning. Abarcando diferentes tipos de requerimientos pertenecientes a las etapas de diseño, otros durante el desarrollo y otros en tiempo de ejecución. A partir de estos se definen los procesos e-learning (Pe-lrn). Al igual que los procesos de negocios en una Aplicación Web convencional (ej., [www.ebay.com](http://www.ebay.com)), los Pe-lrn están compuesto por transacciones Web [8]. En este contexto, una transacción (o transacción e-learning) es definida como una secuencia de actividades que el usuario ejecuta a través de una Aplicación E-learning con el propósito de efectuar una tarea o concretar un objetivo. El conjunto de actividades, sus propiedades y las reglas que gobiernan sus ejecuciones dependen del Pe-lrn que la Aplicación debe brindar.

Las características tecnológica de un Aplicación Web e-learning (AWe-lrn) son idénticas a las Aplicaciones E-learning convencionales (por ejemplo, las utilizadas en el proyecto de e-learning Sakai<sup>1</sup>) que proveen: navegación entre páginas a través de links, ejecución de Transacciones e-learning por medio de los servicios de las herramientas y las operaciones de un Pe-lrn. La principal diferencia tecnológica entre un AWe-lrn y una Aplicación e-learning convencional está en la incorporación de la teoría de coordinación de contratos [9, 1] en la implementación de algunos servicios que las herramientas brindan a los usuarios [2].

En base a la incorporación de los contratos en la implementación y diseño de Transacciones e-learning, aumenta las posibles configuraciones de los Pe-lrn. Comienzan a aparecer nuevas propiedades que tienen que ver con el campo de la Ingeniería de Software. Y, a su vez, fuertemente relacionados al trabajo multidisciplinario entre los actores de un proyecto de construcción de una AWe-lrn (ej. el proyecto *Obra Abierta*<sup>2</sup>). Para este fin, es imprescindible contar con un modelo de diseño que ayude en el ciclo de vida del desarrollo y configuración de una Aplicación e-learning. Integrando las tareas de los expertos en educación, diseñadores y desarrolladores. Desatender o resolver incorrectamente la documentación para el diseño de procesos tipo Pe-lrn pueden causar numerosos problemas reflejados en el proceso de configuración de un AWe-lrn. Estos problemas pueden ser: (1) Dificultades en la comunicación y entendimiento entre los clientes y diseñadores expertos en educación (primero), y entre los diseñadores y los desarrolladores (después), en el proceso de implementación de un AWe-lrn. (2) Determinación de las relaciones donde se justifique la inclusión de contratos. (3) Dificultad para visualizar la trazabilidad entre los procesos e-learning y las implementaciones de las Transacciones.

Este trabajo presenta un diseño comprensivo para el modelado de los procesos de educación e-learning (Pe-lrn) en un Aplicación Web E-learning con la inclusión de los contrato con propiedades context-aware [2]. El modelo está basado en UWAT+ (Distante, 2004), una versión extendida y adaptada de "UWA Transaction Design Model" para el diseño de transacciones en aplicaciones Web. La principales contribuciones de este trabajo son: - El acercamiento de un modelo útil para la representación de transacciones e-learning en una AWe-lrn; permitiendo una mejor distinción de la ubicación (entre servicio-usuario(s) y servicio-

<sup>1</sup> SaKai: Entorno colaborativo y de aprendizaje para enseñar. Es de código abierto y está resuelto con tecnología Java. Url: [www.sakaiprojet.org](http://www.sakaiprojet.org)

<sup>2</sup> *Obra Abierta*: Proyecto de ID (CONICET), que se centra en el desarrollo e implementación de dispositivos hipermediales context-aware dinámico para investigar y aprender en contextos físicos-virtuales de educación superior. Directora: Patricia San Martín

servicio(s)) de la componente contrato dentro del flujo de ejecución. - Ejemplificar, mediante un caso de uso concreto, el uso del modelo UWATc+.

El resto de este trabajo se encuentra organizado de la siguiente manera: Tras esta introducción se presenta una referencia conceptual de los contratos, utilizados en el proyecto Obra Abierta (Sección 2). Luego se continúa con definiciones, formas de documentación y representación de Transacciones e-learning y procesos. (sección 3). Siguiendo, se presenta un modelo conceptual y de diseño sobre la adaptación propuesta para el modelado de Pe-lrn (sección 4). Mediante un caso de uso concreto, se describe el uso de UWATc+ para cada una de sus etapas. Finalizando con una breve conclusión.

## 2 CONTRATOS CONTEXT-AWARE PARA TRANSACCIONES E-LEARNING

En términos generales, la coordinación de contratos es una conexión establecida entre un grupo de objetos (en nuestras consideraciones, un objeto cliente y un determinado servicio). Cuando un objeto cliente (ej. un usuario alumno) efectúa una llamada a un objeto suministro (ej. un servicio de la herramienta Foro), el contrato "intercepta" la llamada y establece una nueva relación teniendo en cuenta el contexto del objeto cliente, el del objeto servidor, e información relevante adquirida y representada como contexto del entorno. Como condición necesaria, la implementación de los contratos no debe alterar el diseño y funcionalidad en la implementación de los objetos.

### 2.1 Elementos de la componente contrato

El contrato puede ser configurado por medio de diferentes mecanismos, desde el lenguaje cotidiano hasta un lenguaje de especificación formal y un lenguaje basado en XML para los casos que sean necesarias especificaciones que puedan ser procesadas por máquinas.

El tipo de tecnología y forma de implementación de los contratos es transparente para los objetos que consumen los servicios en donde se encuentran involucrados. La configuración de un elemento contrato que forma parte de las componentes de un servicio, representa la información necesaria del mismo para ser utilizado por el invocador, sin necesidad que el dicho objeto tenga detalles de la ejecución. El contrato representa una enriquecida y efectiva interface de construcción que contiene toda la información sobre las componentes de los servicios, y deberá tener referencias sobre algún tipo de información de contexto para su utilización.

En la figura 1 se puede observar los elementos conceptuales básicos de esta componente a través de una serie de elementos que lo relacionan.

Para una mejor comprensión de las componentes del modelo se explica su caracterización y funciones particulares:

*Identificador* – Una componente servicio es identificada para un determinado contexto por un único nombre en el espacio de nombre.

*Comportamiento* - De acuerdo con los roles asignados en un determinado contexto, una componente servicio expone comportamientos correspondientes a la provisión y pedido de operaciones, y/o publicaciones y recepción desde/hacia cada contexto. Las operaciones pueden ser definidas en dos tipos – operaciones que ejecutan cálculos o transformaciones (tipo "update") y operaciones que proveen algún tipo de información sobre consultas (tipo "query").

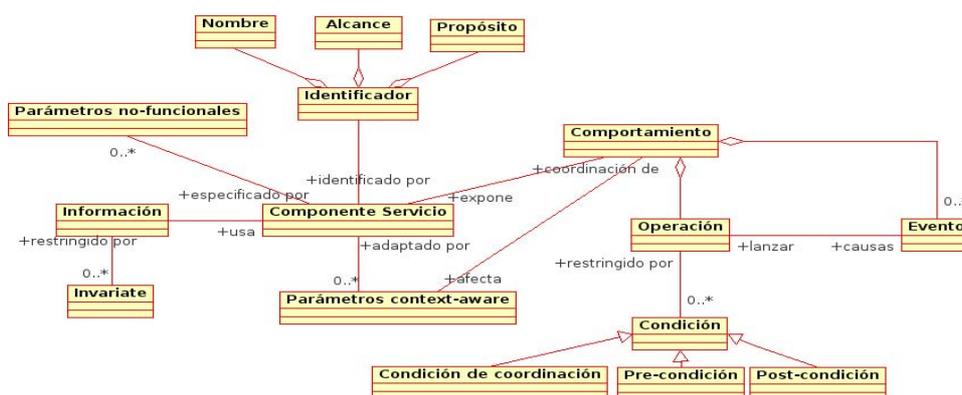


Fig. 1. Meta-modelo del contrato context-aware

Estas, se encuentran enteramente especificadas en base a un contrato, con el uso de pre-condición, post-condición y condicionales para lograr la coordinación entre contratos. En las condiciones de coordinación se especifican cómo requerir y proveer operaciones. Para lograr una comunicación precisa con una componente servicio, no sólo se tiene en cuenta qué operación fue provista o requerida y cómo el ejecutor ha lanzado el evento apropiado, sino también, cómo todas esas actividades están mutuamente relacionadas para ser aprovechadas por el objeto cliente. Un evento del contexto que lanza una operación dada, puede ser parte de un conjunto de pre-condición, mientras que un evento emitido a través de una exitosa operación puede ser parte de una pos-condición.

*Tipos de Información* – Una componente de servicio debe manejar, usar, crear o tener cierta información de recursos con el propósito de proveer servicios adecuadamente. Este elemento del contrato define el tipo de información relevante para las componentes asociadas al contrato, así como también restricciones y reglas sobre instancias de esos tipos. Esto representa un modelo de información lógica de una componente de servicio. Formalmente, esta información de tipos puede ser considerada como definiciones de tipos de los parámetros de las operaciones, o tipos relacionados a ellos.

*Configuración de Parámetros Context-Aware* – Una componente servicio depende del contexto de su actual entorno. Para utilizarse en diferentes contextos logrando la adaptación a eventuales cambios debe tener definido un conjunto de parámetros de configuración. Ejemplos de estos parámetros pueden ser: Contexto-del-Usuario (CU), locación en tiempo y espacio de los servicios consumidos y suministrados. Estos parámetros pueden ser enviados dentro de las invocaciones de las operaciones de los servicios o por medio de otros caminos, mediante componentes de servicios que pueden adaptar su comportamiento ante el cambio de contexto en una determinada situación.

*Parámetros no funcionales* – Una componente servicio puede definir un conjunto de los llamados parámetros no funcionales que caracterizan a la “calidad” de sus prestaciones dentro de un determinado contexto. Estos parámetros, son elementos para los consumidores de los servicios que permiten optar por el uso de un determinado servicio, o buscar otro con el mismo o similar contrato. Como ejemplo de parámetros no funcionales podemos mencionar: Performance, Fiabilidad, Tolerancia a Fallos, Costos, Prioridad y Seguridad.

### 3 DOCUMENTACIÓN DE LOS PROCESOS E-LEARNING

Las transacciones e-learning en un AWe-lrn están definidas como secuencias de actividades asociadas con un flujo de ejecución que permite al usuario desempeñar una determinada tarea y/o alcanzar una meta a través de la Aplicación. Entonces, un proceso e-learning (Pe-lrn) puede ser interpretada como una especificación del concepto de "workflow" en una Aplicación e-learning Web, con las condiciones (restricciones) que implica su concreción. En una Transacción e-learning Web, una *Actividad* está conformada por un conjunto de operaciones simples o complejas sobre datos y contenidos de la Aplicación. Como ejemplo de Transacciones e-learning se pueden mencionar un proceso en el cual un usuario (alumno) participa en un espacio de edición colaborativa (este caso será analizado posteriormente como caso de uso en las secciones posteriores)

Entonces, las transacciones en un AWe-lrn son el camino para la representación de los Pe-lrn, y proveer a los usuarios de servicios accederlos por medio de las herramientas que los contienen (wiki, foro, vídeo conferencia, taller, blog, etc.) La ejecución de Transacciones de un AWe-lrn supone tanto la navegación a través de las herramientas, por medios de los links de las componentes hipermediales, como el uso de sus servicios. Un ejemplo de servicio puede ser en una vídeo conferencia la posibilidad de que un docente edite en una pizarra compartida (con sus alumnos y colaboradores) una determinada ponencia.

El diseño de las transacciones abarca varios niveles de abstracción, distintos formalismos pueden ser usados para su representación y documentación. Al menos tres niveles de abstracción pueden ser representados: (1) nivel conceptual, (2) nivel lógico, y (3) nivel de implementación. El diseño conceptual permite una representación del sistema (las transacciones y su alcances) tal cual son percibidas" por el usuario y despejando las cuestiones de implementación. El diseño de la implementación se encuentra focalizado a proveer a los diseñadores de los AWe-lrn con todas las especificaciones necesarias para la configuración y realización de sus componentes El diseño lógico es un nivel intermedio del diseño de abstracción, utilizado para trasladar las especificaciones centrales del usuario desde el diseño conceptual hacia términos de especificaciones más cercana a las implementaciones.

Al igual que lo que ocurre con todos los artefactos de software, el diseño de una Transacción e-learning para un AWe-lrn se puede tornar muy complejo. Cuanto más complejo sea el diseño, las confusiones entre los diseñadores (expertos en educación y analistas informáticos) y los implementadores (programadores y encargados de la Aplicación) crecerán. Para comunicar efectivamente la idea del diseñador es necesario una apropiada documentación. Si bien la documentación textual ha sido muy utilizada para describir detalles de implementaciones de bajo nivel, teniendo en cuenta que el diseño de las transacciones e-learning se describen en un nivel conceptual, es más adecuado una representación gráfica.

Existen diferentes aportes directamente relacionado a modelos "visuales" en forma de documentación gráfica [3–5]. En este contexto los modelos visuales son representaciones de sistemas de software que soportan múltiples perspectivas. Para el caso del diseño de las transacciones e-learning, una vista puede ser representada por una serie de diagramas pertenecientes a UML (Unified Modeling Language) [12]

Los antecedentes relevantes que se relacionan con lo que entendemos por diseño de Pe-lrn, fueron estudiados de los aportes en el campo del métodos de diseños para Aplicaciones Web experimentados en los últimos años. Concretamente se pueden mencionar ADM (Atzeni y Parente, 2001), OO-H (Koch et al., 2003), OOHDM (Schmid and Rossi, 2004) y UWAT+ (Distante, 2005).

UWAT+ es un meta-modelo para la descripción de los distintos aspectos de Transacciones Web de manera holística. Es una extensión del Modelo de Diseño de Transacciones que forma parte del framework UWA (Ubiquitous Web Applications) [6]. Inspirado en este modelo y extendiéndolo para la contención de los contratos, se describe una adaptación para el diseño de Transacciones e-learning en un AWe-Irn.

#### 4 UWATc+: una adaptación de UWAT+ para el modelado de procesos e-learning

Si bien los métodos de diseño de Transacciones de UWAT+ pueden ser utilizados para la representación de Transacciones e-learning, es necesario efectuar adecuaciones que tengan en cuenta la inclusión a los contratos (según sección 2). Tal cual fue mencionado en la sección 1, desde la perspectiva de los diseñadores, los contratos deben ser visto como una pieza de software para la instrumentación de los servicios de las herramientas. En consecuencia, es necesario tener un modelo que permita una mejor representación de los contratos, la visualización de su inserción en los servicios y las relaciones que en ellos representan (relaciones entre objetos que implementan servicios, usuarios y herramientas en la Aplicación).

La figura 2 se muestra un diagrama de clase UML que representa a los conceptos, las relaciones entre conceptos y los modelos para la representación de Transacciones. Los esquemas en color blanco pertenecen al modelo original UWAT+. El rectángulo y los esquemas grises describen los objetos, modelos y relaciones que conforman el nuevo modelo denominado UWATc+.

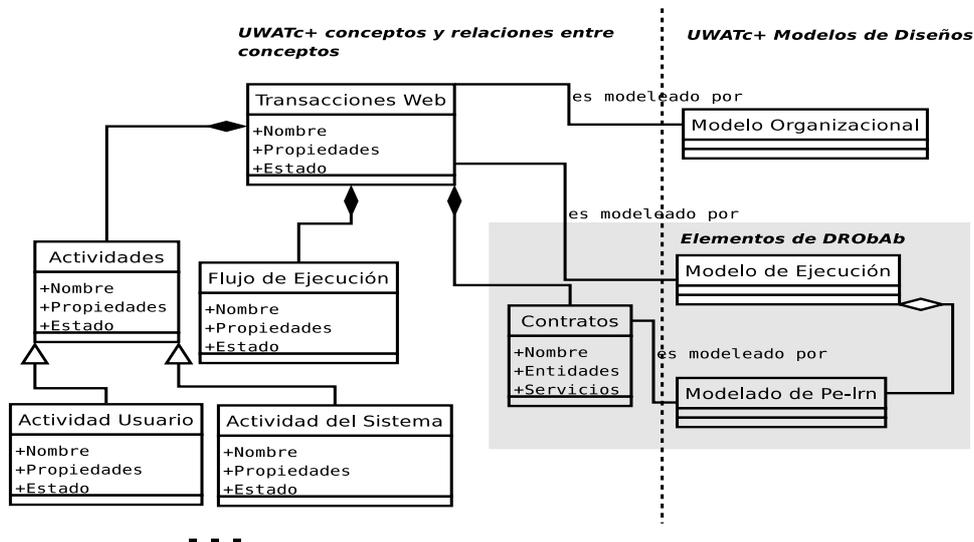


Fig. 2. UWATc+ modelo conceptual y de diseño

Como se describe en el diagrama, una Transacción Web es un objeto complejo (conceptual) compuesto por dos tipos de objetos principales pertenecientes al modelo original de UWAT+ y un tercer objeto agregado para la representación de los contratos pertenecientes a

las Transacciones e-learning. En el primer grupo se encuentra *Actividad* para la distinción de las actividades de los usuarios y del sistema. El objeto *FlujodeEjecución* representa el orden lógico y temporal para la ejecución de las actividades comprendidas en las Transacciones. A su vez, una Transacción Web puede ser descripta por el *ModeloOrganizacional* (desde el punto de vista estático) y el *ModelodeEjecución* para la definición de las reglas de ejecución de la componente actividad (desde el punto de vista dinámico).

Cuando una Transacción Web contiene un contrato (definida como transacción e-learning) debe ser incluida una nueva componente para el diseño (representada en la figura como una relación de agregación en el *ModelodeEjecución*), conjuntamente con un nuevo modelo de diseño, *ModelodePe – lrn*, que permitirá la representación del contrato (caracterizada como relación de asociación con el objeto *Contrato*).

De esta manera quedan conformados los elementos que componen el modelo UWATc+ (rectángulo gris) y sus relaciones con el modelo original UWAT+. A continuación se describe en detalle el modelo usado por UWATc+ para el diseño de las transacciones que utilizan contratos (transacciones e-learning).

#### 4.1 Requerimientos para el diseño de Transacciones e-learning

En esta sección se describen la caracterización de dos tipos representativo de requerimiento que motivaron la creación de este nuevo modelo de diseño (definidos en la sección 1). En base a experiencias recogidas por el grupo del proyecto Obra Abierta en el diseño y configuración de Aplicaciones e-learning, se presentarán dos tipos de requerimientos que deben ser cubiertos por el modelo de diseño. En primer lugar se enuncian cuestiones técnicas de diseño (desde el punto de vista de la Ingeniería de Software), seguido de un comentario sobre el tipo de Transacciones que el diseñador puede especificar a través del modelo.

- Especificar como son afectadas la ejecución de las actividades por los objetos contratos. La relación de una actividad con un contrato se produce cuando existen objetos interrelacionados por medio de un contrato [9]. De esta manera, el diseñador debe poder documentar las información de los objetos involucrados, sus métodos y parámetros. El contrato representa un tipo diferente de relación a la original entre los objetos, con la propiedad de reconfiguración en tiempo de ejecución. Característica que permiten una mejor adaptación a los requerimientos que resuelven las Transacciones e-learning.
- Definir cuál y cómo la información de los objetos contratos ("information object" [13]) es afectada por la ejecución de las Actividades. Una actividad funcional consiste en la ejecución de uno a más operaciones elementales (inserción, borrado, modificación, etc.) sobre los datos de la Aplicación y la información de los objetos contratos envuelta en la actividad. El modelo de diseño de Transacciones e-learning debe permitir al diseñador definir cuales operaciones del contrato son fundamentales para cada actividad, modelando el camino en que cada actividad elemental afecta la información de los objetos involucrados (modificando sus instancias por medios de sus ejecuciones).

## 5 DISEÑO DE PROCESOS E-LEARNING CON UWATc+

En esta sección se describen los resultados de la Aplicación de UWATc+ para el espacio dedicado al libro "Hacia un dispositivo hipermedial context-aware Dinámico. Educación e

investigación para el campo audiovisual interactivo” (San Martín P. 2007, et. al) [2]. Este modelo fue aplicado parcialmente para el diseño de los requerimientos fundamentales y el modelado del comportamiento funcional enmarcado bajo la perspectiva de proyecto Obra Abierta - mencionado anteriormente. Se presentará un caso de uso para ejemplificar el diseño de un Pe-Irn en la Aplicación e-learning para Obra Abierta. El diseño describe parte del proceso en que un usuario hace uso de los servicios de edición de la herramienta Foro a través de un contrato.

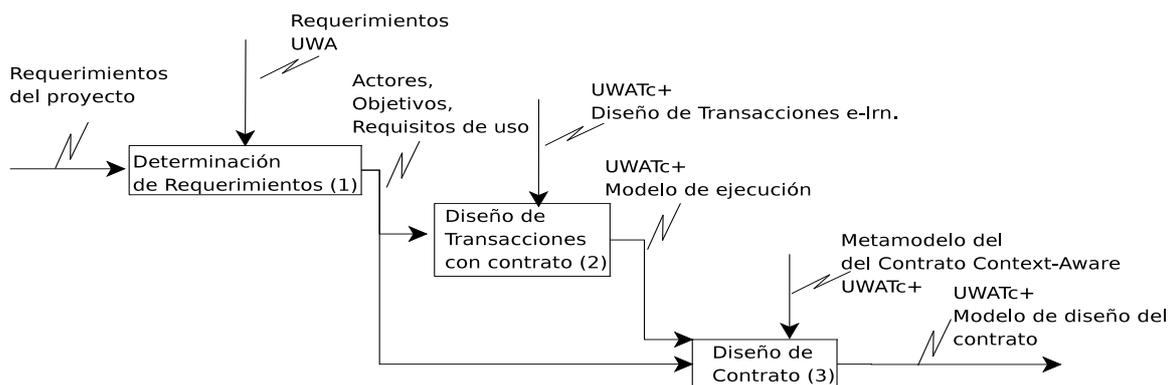


Fig. 3. El proceso de diseñar Pe-Irn en un AWe-Irn con UWATc+

El diagrama de la figura 3 muestra una adaptación del diseño de procesos de UWA [6] que se utiliza en UWATc+ para el diseño de transacciones con contratos. Para ilustrar el proceso de diseño se utiliza la notación IDEF0 (IDEF-0, 1993). En comparación con la metodología original de UWAT+, se agregaron las fases del diseño de contrato y fue modificado el modelo de diseño de la transacciones. Además, fueron excluidas las fases de "diseño de la información" y "diseño de publicación". Las fases del procesos de diseño son: (1) Determinación de Requerimientos; (2) Diseño de transacciones con contrato; (3) Diseño de contrato.

### 5.1 Determinación de Requerimientos

La fase de Determinación de Requerimientos toma como entrada las especificaciones del proyecto y produce, por medio de un mecanismo de refinamiento, las siguientes salidas: Cada actor con su objetivos relativo. Los requerimientos para la construcción y configuración de un AWe-Irn.

El modelo utilizado es orientado a objetivos: cada actor se identifica con al menos un objetivo, i.e, una abstracción de los objetivos que a través de la aplicación se debe alcanzar; cada objetivo es refinado en otros sub-objetivos, hasta poder definir el requerimiento en un bajo nivel suficiente para poder ser implementado. Este fase es similar a la descrita en (UWA Consortium, 2001) [6].

Para este caso de uso, atendiendo a los lineamientos en [2], se describe parte del modelo original donde se caracterizan los objetivos involucrado con un actor del sistema. A su vez, a medida que se van derivando los sub-objetivos comienzan a establecerse requerimientos concernientes a la teoría de coordinación de contratos context-aware [1, 9]. En la figura 4 dicha situación ocurre en la derivación de las tres ramas de objetivos y sub-objetivos, influyendo

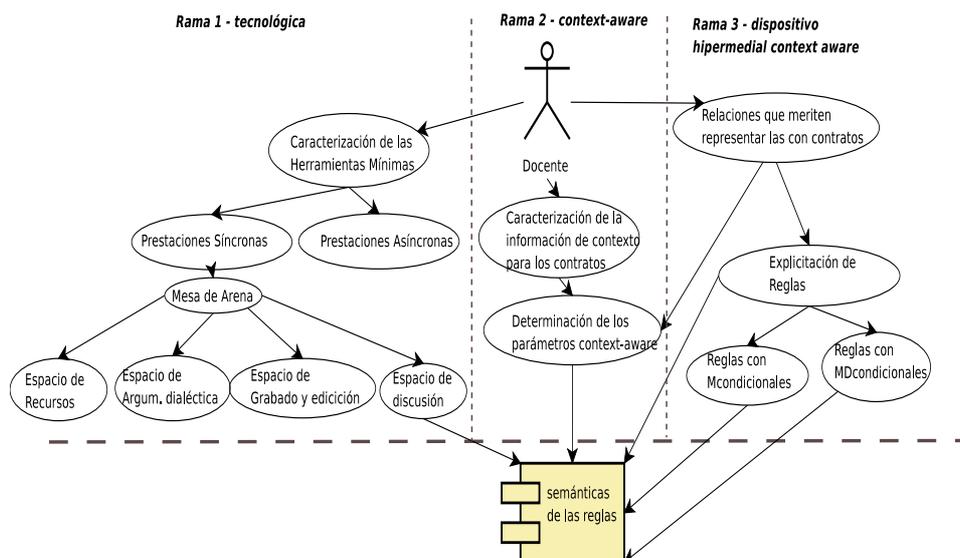


Fig. 4. Objetivos de alto nivel para un Pe-Irn

directamente en la composición de la componente contrato. En este caso, tomando desde la **rama 1** un servicio de una herramienta de un espacio de discusión (herramienta Foro); de la **rama 2** se desprenden las información necesaria para poder articular dicho servicio teniendo en cuenta la información de contexto; la **rama 3** aporta el consenso de los expertos (del dominio e-learning) para la inclusión de los contratos en aquellas relaciones que mantendrán las propiedades de la Aplicación e-learning [1]. A través de este modelo, se logra un primer acercamiento sobre como se relaciona un contrato con: los requerimientos, el tipo de objetivos para cada requerimiento y los actores del sistema.

## 5.2 Diseño de Transacciones e-learning

El diseño de transacciones e-learning retoma la misma idea y modelo propuesto por UWAT+ para el diseño de transacciones [7]. Partiendo de los resultados de la fase de *Determinación de Requerimientos*, fundamentalmente de la caracterización de los contratos, pueden ser seleccionados una serie de transacciones, i.e., objetivos que requieran la ejecución de una o más Actividades para su cumplimiento. Para cada uno de los objetivos que incluya contrato deben ser diseñadas Transacciones e-learning (de igual forma que con las transacciones en UWAT+), los cuales en principio deben ser establecidos desde un punto de vista estático (en este trabajo no abordaremos tal consideración) y luego, desde un punto de vista dinámico por medio de un modelo de ejecución. En la figura 5 se muestra una porción del modelo de ejecución de un transacción e-learning cuyo contrato asociado fue caracterizado en la fase de *Determinación de Requerimientos* (figura 4, sección 5.1). En el diseño se describe el flujo de ejecución entre las Actividades de la transacción. El modelo de ejecución es una adaptación del diagrama de actividades de UML [11] en el que las Actividades y sub-Actividades están representadas por estados (óvalos), y el flujo de ejecución entre ellos se representa por medio de transiciones (arcos). Los óvalos con el símbolo (\*) - un asterisco entre paréntesis - refiere a una Actividad que representa a un conjunto de Actividades compuestas, y cuyo modelo de ejecución debe

ser representado con otro diagrama. Un óvalo simple representa una Actividad Elemental. Un óvalo color gris indica una Actividad compuesta de las sub-actividad que se encuentra dentro. Una sub-Actividad representada con un óvalo color gris indica que es dependiente de la Actividad que la contiene, esto quiere decir que su ejecución estará acompañada por otra sub-Actividad y no puede ser incluida en otra composición. Los arcos de líneas continuas indican flujos de ejecución obligatoria (transacciones hacia Actividades requeridas), mientras que los actos con líneas de puntos representan flujo de ejecuciones opcionales (transacciones hacia Actividades opcionales).

Cada relación posible entre actividades es representada por medio de una arco entre ellas. A cada arco se le asocia un texto que indica bajo que condiciones se produce la transacción, o el resultado de la ejecución de la Actividad de origen. Para describir como colaboran los usuarios de la aplicación en la ejecución de la Actividades puede ser anexado un diagrama UML Swimlanes.

Cuando una Actividad ejecuta servicios implementados por contratos, entonces, se establece un arco saliente hacia un contrato. El contrato tiene un nombre, y entre paréntesis se indican cuales son los objetos participantes (en el caso de tener ese tipo de información). Para representarlo visualmente se utiliza el estereotipo del elemento componente de UML. Las Actividades que influyan en la modificación de los contratos en tiempo de ejecución se conectan a través de un arco de línea de puntos, igual a los utilizado en la representación de los flujos opcionales. Los detalles de implementación del contratos se detallan en un diagrama aparte, perteneciente a la fase descrita en la siguiente sección.

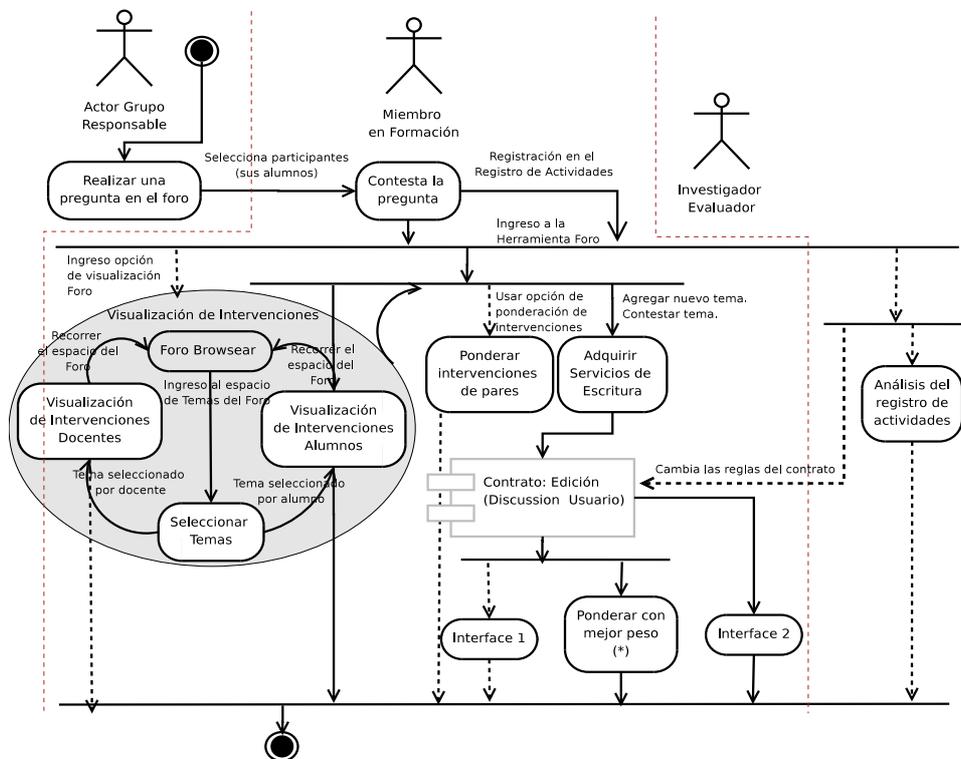


Fig. 5. Modelo de ejecución de Procesos con Transacciones e-learning

Por ejemplo, una de las opciones de la herramienta Foro de Obra Abierta (<http://200.80.157.171:8080/portal>) es la visualización de las intervenciones de los usuarios en el Foro. Un usuario docente puede seleccionar la opción "Foro" de la página principal de la Aplicación, luego seleccionar el tipo de "vista" (mediante un "comboBox") para ingresar en modo "browser" donde se muestran las intervenciones de los usuarios por temas. Una vez seleccionado el tema (por medio de la Actividad "Seleccionar Tema", figura 5), es posible ingresar al espacio de las intervenciones de los usuarios por medio de los roles de docente o alumno. Los docentes y alumnos tienen diferentes tipos de "vistas", permisos y servicios asociados (representadas en las actividades: "Visualización de Intervenciones Docentes" y "Visualización de Intervenciones Alumnos"). A través del Browser (Actividad "Foro Browser") se recorre todo el contenido del espacio y al mismo tiempo puede ser seleccionado otro tema para visualizar.

En cambio, si la opción seleccionado es añadir o responder temas, se ingresa a una página Web configurada para editar texto (por medio de la Actividad "Adquirir servicios de escritura"). Algunos de los servicios de edición y configuración de opciones son implementadas a través del contrato "Edición" (representado por la figura de la componente UML, con el contorno color gris). El flujo de ejecución, luego de la intervención de los contrato, dependerán de las reglas de coordinación y se representan con los arcos salientes similares a los usados para representar las relaciones entre estados.

### 5.3 Diagrama de un Contrato

Existen diferentes formas de representación de los contratos definidos en la sección 2, la herramienta CED (Coordination Development Environment) <sup>3</sup> los implementa a través de un lenguaje llamado Oblog [10]. En [14] se muestra como a través de CommUnity se definen primitivas de modelado y técnicas de diseño basadas en la separación de la "coordinación" del "cómputo".

En UWATc+ se brinda un diagrama de representación de contrato, donde se describen todos los datos que lo instancian. Cada tipo de dato y valor, pertenece a un elemento del meta-modelo de la figura 1. Teniendo en cuenta la figura 6, en primer lugar (item 1) se identifican los objetos participantes en el contrato; en el ejemplo de la figura 6 *DiscussionAction* y *UserAction* hacen referencias a dos clases reales perteneciente a la implementación de la herramienta Foro y Usuarios de la Aplicación Obra Abierta, respectivamente. Luego, se identifican los nombres de los parámetros context-aware significativos para el contrato, alineados en la misma columna del objeto que lo comparte (item 2). En Servicios (item 3) deben ser representados los métodos del objeto, que al ser ejecutados, provocan la intervención del contrato. Para este ejemplo *initState* y *getIdentifier* son ejecutados cuando un usuario ingresa a la herramienta Foro y las posteriores funcionalidades (servicios) disponibles dependen de la ejecución del contrato *Edición* (la figura 5 muestra la superposición del contrato entre los servicios de edición y las nuevas interfaces o funcionalidades). Las siguientes filas (items 4 y 5) se refieren a las pre y post-condiciones que se deben cumplir en la ejecución del contrato. Por último se explicitan las reglas de coordinación. Siguiendo con el ejemplo, en la parte del condicional  $u.contexto = 'l1;p1;docente;r1;c1;'$  verifica si el contexto del usuario  $u$  está compuesto por la locación  $l1$ , tienen el perfil  $p1$ , es un *docente*, cumple el rol  $r1$  y pertenece a la categoría  $c1$  (este tipo de representación de contexto se encuentra desarrollado en [2]). En

<sup>3</sup> CED: es el primer prototipo de una herramienta que implementa el uso de la coordinación de contrato en aplicaciones Java. La herramienta pertenece a ATX Software ([www.atxsoftware.com.ar](http://www.atxsoftware.com.ar)); fue desarrollada en Java y es de código abierta.

cuanto a la acción de la regla de coordinación, continuando con el mismo ejemplo, se induce la ejecución del método *showMessage* del objeto *d* (DiscussionAction). El final del diagrama está dedicado a comentarios generales; cada comentario debe ir acompañado con el número de ítem (1,2,3,4,5 o 6) al que hace referencia.

<b>Contrato: Edición</b>		
1. <i>Participantes:</i>	<b>d:</b> DiscussionAction	<b>u:</b> UserAction
2. <i>Param. c-a:</i>	state, portlet, rundata, context	contextidentifier, identifier
3. <i>Servicios:</i>	initState()	getIdentifier()
4. <i>Pre-Cond:</i>	existe < <i>contexto</i> >	existe < <i>contexto</i> >
5. <i>Pos-Cond:</i>	modifica < <i>contexto</i> >	
6. <i>Reglas de Coordinación:</i>	<b>Si</b> u.contexto='p1;d;r1;c1;' <b>entonces</b> d.showMessage(data,string)	
<b>Comentario</b>		
1. DiscussionAction y UserAction pertenecen a clases implementadas en JAVA del proyecto Sakai. 4 y 5. < <i>contexto</i> > refiere a un objeto donde se oculta toda la información de contexto que caracteriza a los usuarios de la plataforma		

**Fig. 6.** Diagrama del contrato: Edición

## 6 CONCLUSIÓN

En base a la experiencia recogida en el proyecto Obra Abierta, es posible asegurar que la implementación del modelo de diseño UWATc+ en el ciclo de vida del desarrollo Aplicaciones E-Learning permitió un mejor entendimientos entre los expertos en educación, diseñadores y programadores. El modelo también ayudó en la comprensión de la teoría de coordinación de contrato aplicada a transacciones para Aplicaciones e-learning.

## References

1. Sartorio A., San Martín P. Sistemas Context-Aware en dispositivos hipermediales dinámicos para educación e investigación. *Universidad Nacional de Quilmes (UNQ). Capítulo. En impresión.*
2. San Martín P., Sartorio A., Guarnieri G., De la Riestra M. Hacia un dispositivo hipermedial context aware dinámico. Educación e Investigación para el campo audiovisual interactivo. *Universidad Nacional de Quilmes (UNQ). Libro. En impresión.*
3. Hartmann J., Huang S., and Tilley S. Documenting Software Systems with Views II: An Integrated Approach Based on XML. Proceedings of the 19th Annual International Conference on Systems Documentation (SIGDOC 2001: October 21-24, 2001; Santa Fe, NM), pp. 237-246. ACM Press: New York, NY, 2001.
4. Tilley S. and Huang S. Documenting Software Systems with Views III: Towards a Task-Oriented Classification of Program Visualization Techniques. Proceedings of the 20th Annual International Conference on Systems Documentation (SIGDOC 2002: October 20-23, 2002; Toronto, Canada), pp. 226-233. ACM Press: New York, NY, 2002.
5. Tilley S., Müller H. and Orgun M. Documenting Software Systems with Views. *Proceedings of the 10th Annual International Conference on Systems Documentation (SIGDOC '92: October 13-16, 1992; Ottawa, Canada), pp. 211-219. ACM Press: New York, NY, 1992.*
6. UWA Consortium. Ubiquitous web applications. *Proceedings of The eBusiness and eWork Conference (e2002), 16-18 October, Prague, Czech Republic.*
7. Distante D., Tilley S. and Huang S. (2004b). Documenting software systems with views IV: documenting web transaction design with UWAT+. *Proceedings of the 22nd International Conference on Design of Communication (SIGDOC 2004), Memphis, TN, New York, NY: ACM Press, 10-13 October.*
8. Brambilla M., Ceri S., Fraternali P. and Manolescu I. Process modeling in web applications. *ACM Transactions on Software Engineering and Methodology (TOSEM), in print.*
9. Andrade L. and Fiadeiro J.L. Interconnecting Objects via Contracts. In *UML'99 - Beyond the Standard, R. France and B. Rumpe (eds), LNCS 1723, Springer Verlag 1999, 566-583.*
10. The Oblog Corporation. The Oblog Specification Language. <http://www.oblog.com/tech/spec.html>
11. Object Management Group (OMG). Unified Language Modeling Specification (Version 2.0). *Online at www.omg.org, 2004.*
12. Murphy S., Tilley S. and Huang S. The 4th Workshop on Graphical Documentation: UML Style Guidelines. *To be held as part of The 22nd Annual International Conference on Design of Communication (SIGDOC 2004: October 10-13, 2004; Memphis, TN)*
13. Distante D., Tilley S. and Huang S. Documenting software systems with views IV: documenting web transaction design with UWAT+. *Proceedings of the 22nd International Conference on Design of Communication (SIGDOC 2004), Memphis, TN, New York, NY: ACM, 10-13 October.*
14. Gelernter D. and Carriero N. Coordination Languages and their Significance. *Communications ACM 35, 2, pp. 97- 107, 1992.*