

SEPI: una herramienta para el Seguimiento y Evaluación de Procesos Interactivos del DHD

Guillermo Rodríguez ¹
Alejandro Sartorio ¹
Patricia San Martín ¹

¹ Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas, CIFASIS (CONICET-UNR-UPCAM), Bv. 27 de febrero 210 bis, 2000 Rosario, Argentina.

guille@fceia.unr.edu.ar; sartorio@cifasis-conicet.gov.ar; sanmartin@cifasis-conicet.gov.ar

Resumen. El trabajo describe el primer prototipo de una herramienta integrada que colabora en el seguimiento y evaluación analítica de procesos de interacción mediatizados por un Dispositivo Hipermedial Dinámico (DHD). A su vez, dichos resultados retroalimentados como información de contexto consumido por una pieza de software denominada “contrato”. El desarrollo presentado ha sido adaptado al entorno colaborativo SAKAI. Para esto se describen los elementos tecnológicos necesarios para la integración, como así también un modelo que tiene en cuenta experiencias vinculadas al agregado de nuevas componentes en implementaciones resueltas para esta clase de framework y que articula los resultados de la simulación obtenida de un módulo de ejecución DEVS (Discrete Event System) que integra las métricas necesarias. Por último, se presenta un caso de uso brindando características y detalles funcionales de la herramienta que señala los seis pasos metodológicos del modelo desde la perspectiva de un usuario final. De esta manera, se ha posibilitado un análisis cualicuantitativo de las participaciones y nuevas propiedades de adaptación al DHD.

Palabras Clave: Interactividad – Seguimiento cualitativo – Dispositivo Hipermedial Dinámico – Coordinación de Contratos – Eventos Discretos.

1 Introducción

Las actuales Tecnologías de la Información y Comunicación (TIC) han posibilitado la construcción de una nueva realidad físico-virtual donde los sujetos pueden ser partícipes de diversas redes sociotécnicas. Estas redes están conformadas por una multiplicidad de componentes y relaciones, que se configuran y reconfiguran por las

diversas interacciones mediatizadas, en función de una gran diversidad de requerimientos.

La necesidad de evaluar y propiciar una mejor calidad de los procesos interactivos, cuyos propósitos se centren en investigar, educar y/o producir a partir de la participación responsable en contextos físicos-virtuales, adquiere especial relevancia para la construcción conjunta e inclusiva de la denominada “Sociedad de la Información y del Conocimiento”. En atención a esta necesidad insoslayable, el presente trabajo describe tecnológicamente el primer prototipo de una herramienta integrada denominada “SEPI-DHD” que colabora en el Seguimiento y Evaluación analítica de los mencionados procesos de interacción mediatizados por un Dispositivo Hipermedial Dinámico (DHD) retroalimentando información de contexto de los participantes.

Se conceptualiza como DHD a una red sociotécnica [1] conformada por la conjunción de tecnologías y aspectos sociales que posibilita a los sujetos realizar con el otro, acciones en interacción responsable para investigar, aprender, dialogar, confrontar, componer, evaluar, diseminar bajo la modalidad de taller físico-virtual, utilizando la potencialidad comunicacional, transformadora y abierta de lo hipermedial [2], reguladas según el caso por una “coordinación de contratos” [3].

De esta manera, el DHD se constituye como una entidad compleja [4] compuesta por la integración de dos dimensiones indisociables: una técnica (o conjunto de técnicas constructivas que comportan una materialidad y una configuración particular) y una social dada por las relaciones intersubjetivas y la situación en la que se inscriben.

La necesidad de SEPI-DHD, se fundamenta en el marco de dos requerimientos de I+D relacionados: el primero está referido a la importancia de efectuar análisis evaluativos de los desarrollos e implementaciones llevados adelante en el entorno colaborativo de I+D+T del Programa “Dispositivos Hipermediales Dinámicos” [5]. El segundo requerimiento, se encuentra en una fase experimental e investigativa y está relacionado con la implementación de un original desarrollo de pieza de software denominada “contrato” que debe ser ubicada reflexivamente en el sistema por los usuarios calificados para diseñar e implementar el espacio de formación, investigación y/o transferencia con la finalidad de potenciar el aspecto dinámico de los procesos de interacción responsable del DHD [6].

En referencia a dichos requerimientos, adoptó la utilización del formalismo DEVS (Discrete EVents dynamic Systems) [7] que propone una teoría de modelado de eventos discretos en sistemas a tiempo continuo, permitiendo a su vez, una descripción modular de los fenómenos y el abordaje de la complejidad usando una aproximación jerárquica. En la implementación de dicho formalismo se integran las métricas de ponderación siguiendo las recomendaciones del framework INCAMI [8].

A su vez, tecnológicamente el entorno colaborativo de prueba está provisto por un agregado de una pieza de software para la inyección de propiedades de coordinación de contratos sensibles al contexto [9]. Esta propiedad se logra a través de la implementación de contratos [10] con mecanismos de coordinación y componentes de sistemas Context-Aware. La utilización de reglas es esencial para la implementación de las acciones de los contratos estando compuestas por los condicionales, en donde se centra parte de la lógica de adaptación que requiere el DHD. Algunos condicionales implementados requieren de mecanismos externos que

colaboren en la composición de sus valores de verdad que son ponderados a partir de simulaciones por medio del modelo DEVS [11]. La herramienta SEPI-DHD de código abierto funcionalmente logra la integración tecnológica de los Condicionales DEVS, pertenecientes a los contratos con el módulo de ejecución del modelo DEVS [12].

En la siguiente sección describiremos los elementos tecnológicos del DHD en relación directa con los Condicionales DEVS. Luego, en la sección 3 presentaremos el modelo de integración adoptado para la implementación de la herramienta. En la sección 4 abordaremos algunas de las características, funcionalidades y un caso de uso concreto para arribar finalmente a breves conclusiones y prospectiva del desarrollo.

2. Elementos para la integración

En esta sección describiremos aspectos tecnológicos y componentes del DHD que la herramienta SEPI toma en cuenta para la conexión entre los Condicionales DEVS y las métricas de interacción DEVS.

Sintéticamente, los *Condicionales DEVS* son condicionales simples que pertenecen a un conjunto de reglas que definen acciones dentro de componentes computacionales contratos. Los contratos, definidos por Meyer [10], se basan en la metáfora de que un elemento de un sistema de software colabora con otro, manteniendo obligaciones y beneficios mutuos. En este caso, los contratos son objetos de primera clase dentro de la original propuesta tecnológica del DHD [6] donde se le agregan propiedades de sensibilidad al contexto y dinamismo.

Dado que la evaluación sobre los valores de verdad de los *Condicionales DEVS* se referencian en resultados obtenidos al ejecutarse un simulador de un modelo de eventos discretos que representa una métrica cuali-cuantitativa [12], podemos afirmar que estos condicionales promueven nuevas propiedades de adaptación del DHD posibilitando que las reglas de los contratos tengan mayor grado de expresión.

En la siguiente Figura 1, se muestra cuales son los elementos que se deben tener en cuenta para implementar los valores calculados que deben contrastarse con valores testigos para formar los condicionales de las proposiciones lógicas de las reglas. Las relaciones entre las componentes influyen en elementos conceptuales que determinarán algunas de las características funcionales de los requerimientos del DHD [9] relacionados a los requisitos de comportamiento y configuración.

La herramienta cuenta con tres tipos de servicios que derivan del núcleo base del framework del DHD, uno de ellos (*S.Conexión*) encargado de establecer la conexión para incorporar valores calculados dentro de los Condicionales DEVS. Otro tipo de servicio (*S.Ejecución*) implementa las secuencias de ejecuciones del modelo DEVS que obtiene los resultados de las interacciones. Algunos de estos pasos serán descritos en la sección 4 y forman parte de los requisitos que se deben cumplir para ejecutar la simulación implementada en el módulo DEVS (*ModDEVS*), que en este caso se ajusta a las interfaces que provee PowerDEVS [13].

El último servicio que extiende los servicios base (*S.Configuración*) se encarga de implementar todas las cuestiones de configuración de la métrica que provienen de la interfaz de usuario. En este caso se provee al usuario de un prototipo de interfaz

que permite establecer las ponderaciones de los coeficientes de la métrica de interacción representada en DEVS, para que luego se puedan transformar en los parámetros necesarios al invocar el método de configuración de *S.Configuración*. El componente *Configuración* refiere a la relación entre el servicio de interfaz (*S.Conf*) y los elementos utilizados para implementar la interfaz del usuario de la herramienta, en este caso representada con la clase *UI_Usuario*.

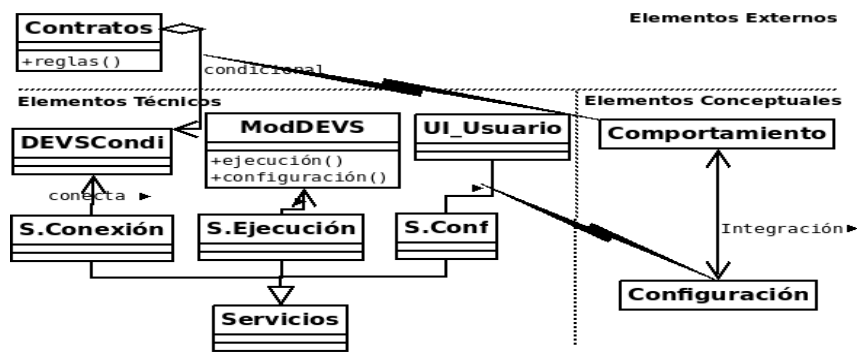


Fig. 1. Modelo de elementos y relaciones de los DEVS condicionales.

De esta manera queda establecida una relación conceptual entre los servicios (*Servicios*) y los contratos que definen el comportamiento adaptativo de algunas de las acciones del DHD. Esta última relación fue representada en la Figura 1 mediante la relación de agregación entre el contrato (*Contrato*) y el Condicional DEVS (*DEVSCondi*), determinada por la relación conceptual *Comportamiento*.

Las interfaces de los métodos las abordaremos seguidamente como parte de la herramienta que implementa el modelo de integración. De esta manera queda establecida una nueva relación conceptual, llamada *relación de Integración*, a partir de las componentes conceptuales *Comportamiento* y *Configuración*.

Cabe destacar que los elementos conceptuales representados en la figura forman parte de los requerimientos de adaptabilidad que debe cumplir el DHD [6]. A su vez, los elementos técnicos pertenecen al diseño propuesto para el modelo de integración que se implementa a través de la herramienta. Las funcionalidades que interpretan el servicio de ejecución las describiremos en la sección 4.

3. Modelo conceptual de integración

En esta sección representaremos un modelo general de integración teniendo en cuenta experiencias vinculadas al agregado de nuevas componentes en determinadas implementaciones resueltas para entornos colaborativos similares al diseño del framework [6]. En este caso, el modelo determina los componentes, sub-sistemas y relaciones sobre la herramienta de integración encargada de articular los resultados de la simulación del modelo DEVS para insertarlo en los contratos.

En la siguiente Figura 2 observamos una primera conexión entre los servicios de la herramienta, que fueron introducidos en la sección anterior, con el modelo de simulación DEVS. Para este caso contamos con los métodos del módulo *Integrador*, representado por una clase UML [14] que implementan las secuencias de ejecuciones que se deben respetar para que el intérprete DEVS (en este caso PoweDEVS [13]) tome los valores de entrada adecuados (método *get_inputDEVS*) que son procesados por medio de una función transferencia parametrizada (método *set_Transferencia*) a partir de un archivo de configuración XML. Luego se toman los valores de salida (método *get_outputDEVS*) para su posterior procesamiento. Todos los datos de entradas son extraídos de la base de datos perteneciente al entorno colaborativo utilizado, a través de métodos privado *get_DB*. Los parámetros que implican la ejecución del simulador DEVS deben ser alcanzados por las interfaces representadas en la clase *Integrador*.

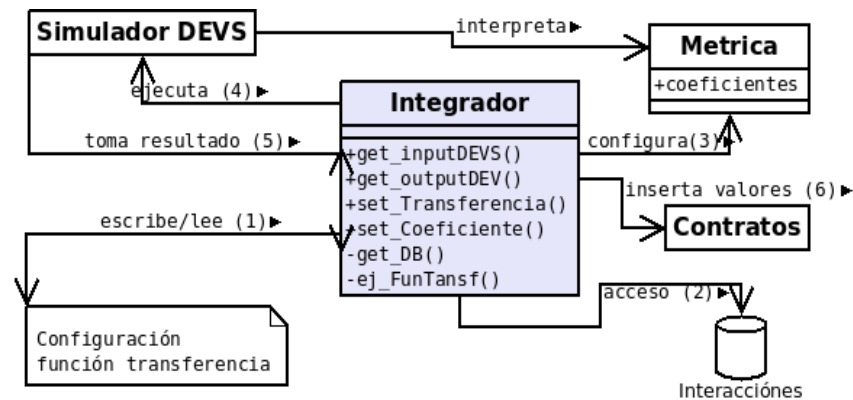


Fig. 2. Modelo de integración para contratos, métricas y modelo DEVS

Las cuestiones de parametrización y representación de los valores, pre y pos-condiciones que se encuadren en un nivel superior para lograr una mejor interpretación y manejo por parte de los usuarios se guardan en el archivo de configuración e interviene el servicio *S.Configuración* mencionado en la sección 2.

Si bien, en este caso, el diseño de integración propuesto se basa en la descripción de componentes de software de la herramienta de integración aquí propuesta, es necesario tener en cuenta otros elementos que en este diseño no se muestran. En otro trabajo [15] hemos desarrollado lo correspondiente a cada una de las áreas mencionadas con un nivel más profundo de detalle. Además, se muestra que la principal componente para lograr la integración está representada por la incorporación de una relación de agregación entre una componente Contrato y una entidad Método que se corresponde con la arquitectura que propone INCAMI [8] para la construcción de métricas.

En términos generales, la SEPI-DHD es una aplicación que respeta la arquitectura del framework, utilizando los servicios base para el acceso a la base de datos. Por otro lado, permite la aplicación de una función transferencia que transforma dichos datos teniendo en cuenta un archivo de parametrización. Los diversos componentes tecnológicos que complementan el desarrollo cumplen los

estándares del framework, en este caso utilizamos Servlets y Beans teniendo en cuenta el acceso a los servicios base del framework que permitan el registro de la aplicación como herramienta.

4 Implementación en un caso de uso

A continuación desarrollaremos un caso de uso brindando detalles funcionales sobre la herramienta desde la perspectiva de un usuario final. Dicho camino se divide en seis pasos metodológicos coincidentes con las relaciones descriptas en la Figura 2.

Paso 1: Escritura y lectura del XML

La herramienta contiene una interfaz apropiada para la representación de los coeficientes que componen la métrica, permitiendo de esta manera que el usuario controle las ponderaciones que reflejan propiedades cuali-cuantitativas sobre las interacciones de los participantes. Se presentan los pesos necesarios (Rol, Tipo de Paquete Hipermedial, Tipo de Herramienta, Tipo de Servicio) con un valor por defecto igual a 1 (para los valores nulos el valor por defecto es cero). A su vez se da la posibilidad de modificar esos valores a través de la opción *Modificar Ponderación*, seleccionando el coeficiente y el subtipo de coeficiente.

Luego se crea automáticamente un archivo XML donde se detallan todos los valores necesarios para que la función transferencia procese las tablas y valores de sus campos que sirvan como entradas al simulador DEVS. La siguiente tabla describe el significado de las marcas (“tags”) donde se brindan los detalles concretos para los parámetros de las interfases de cada subsistema de la sección 3.

<code><dhdDB> Sakai </dhdDB></code>	Nombre de la base de datos del sistema colaborativo
<code><dhdTable> sakai_event </dhdTable></code>	Tabla base de datos Sakai
<code><dhdField> EVENT </dhdField></code>	Campo de la base de datos
<code><dhdConditionType> igual </dhdConditionType></code>	Tipo de comparación
<code><dhdValue> content.new </dhdValue></code>	Valor del campo para comparar
<code><dhdValueTrue> 1 </dhdValueTrue></code>	Valor que se devuelve la función transferencia si la condición es verdadera
<code><dhdValueFalse> nothing </dhdValueFalse></code>	Valor que se devuelve la función transferencia si la condición es falsa
<code><dhdValueFalse> DEVS.c1.DEVS.c2 </dhdValueFalse></code>	Referencia el coeficiente de ponderación de la métrica representada en DEVS

Paso 2: Acceso a la base de datos

Si bien este paso es transparente para el usuario final los diseñadores e implementadores deben cumplir con ciertos requisitos que se ajustan a las necesidades del modelo de integración aquí propuesto. En este sentido se establecen diferentes tipos de conexiones entre subsistemas a través de mensajes y secuenciación

de ejecución de tareas (ej, relación toma resultado (5), sección 3). Por otro lado, es necesario implementar acciones de penetración entre sistemas que deben respetar una infraestructura tecnológica y de diseño, este es el caso de la vinculación que se propone en el modelo de integración entre el módulo Integrador y la base de datos que contiene las interacciones (relación acceso (2), sección 3).

En este caso, proponemos un modelo de acceso a base de datos a través de iBATIS teniendo en cuenta que la capa de abstracción será la interfaz con la capa de la lógica de la función transferencia, haciendo las veces de “facade” entre la aplicación y la persistencia. Teniendo en cuenta el patrón Data Access Object (DAO), nuestra implementación utilizó del framework DAO (ibatis-dao.jar). La capa de Framework de Persistencia será la interfaz con el gestor de Base de Datos ocupándose de la gestión de los datos mediante el framework SQL-MAP (ibatis-sqlmap.jar).

En la siguiente Figura 3, mostraremos un ejemplo de las clases y relaciones que implementa el acceso a la tabla sakai_event de la base de datos Sakai.

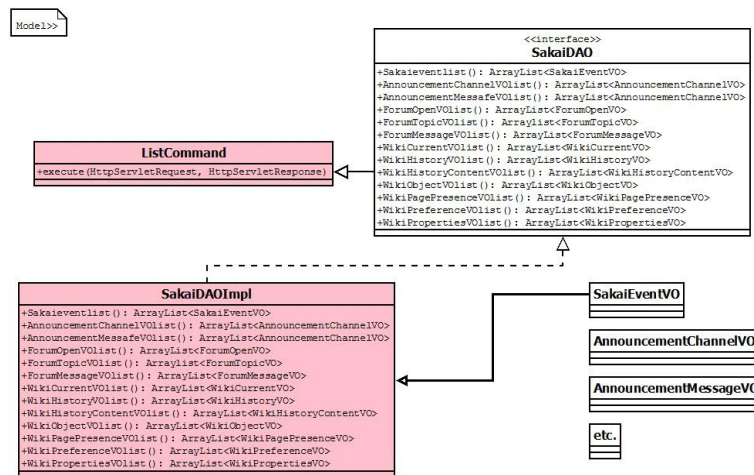


Fig. 3. Estructura para el acceso a la Base de Datos.

Paso 3: Configuración de coeficientes de la métrica

A partir del acceso a la base de datos, la aplicación genera un archivo *input.csv* en el cual se guardan los vectores (de ocho componentes) [12] que serán los datos de entrada del entorno utilizado para correr el modelo DEVS que integra las métricas (relación acceso (3), sección 3). Cada métrica directa tiene asociado un método de medición claramente especificado.

Paso 4: Ejecución de la simulación

Una vez generados los valores de entrada correctos, estos serán utilizados en la simulación. Se ejecutará el modelo DEVS general como un módulo *model.exe* que posee dos parámetros, el archivo de entrada antes generado: *input.csv*, y el número de eventos totales (relación acceso (4), sección 3). Luego de la ejecución se genera el archivo *output.csv* el cual contiene los valores de niveles de interactividad para cada

participación.

Paso 5: Lectura de resultados

A continuación, la Figura 4 muestra gráficamente a modo de ejemplo, los resultados obtenidos de Nivel de Interactividad para cada participación a través del tiempo en el entorno colaborativo de investigación del Programa DHD, antes mencionado. La interfase posibilita diversos filtrados y colorea los niveles de interactividad según los requerimientos del usuario (relación acceso (5), sección 3).

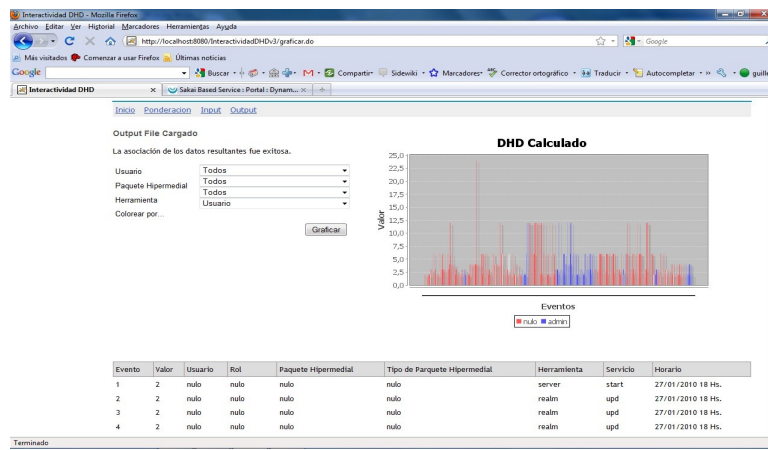


Fig. 4. Resultados obtenidos.

Paso 6: Insertar valores en los contratos

El resultado de este análisis es considerado una información de contexto, resignificando una característica del comportamiento de los participantes y atendiendo a la posibilidad de usar la información de los niveles de interactividad como parámetro context-aware de los contratos (relación acceso (6), sección 3).

5. Conclusiones

En este trabajo hemos presentado la herramienta integrada SEPI para el seguimiento y evaluación de procesos de interacción del DHD adaptado al entorno colaborativo SAKAI. A su vez, se extendieron las propiedades expresivas de las reglas de coordinación de contratos del DHD a partir de los resultados de un mecanismo externo. De esta manera, es posible establecer un lazo de retroalimentación entre las acciones efectuadas en los entornos colaborativos y las que devengan de los contratos.

Además de las cualidades funcionales logradas, la herramienta SEPI concreta la integración de los subsistemas por medio de mensajes, ejecución de tareas y penetración, atendiendo a las líneas de diseño e implementación establecidas dentro del marco teórico-metodológico del DHD.

La perspectiva actual se centra en la optimización de la herramienta, su prueba

en diversos escenarios de uso y la puesta a disposición de la misma en los foros de discusión con la documentación correspondiente.

Referencias

1. Foucault, M.: Saber y verdad. La Piqueta, Madrid (1991).
2. San Martín, P.; Guarnieri, G.; Rodríguez G.; Bongiovani, P.; Sartorio A. El dispositivo Hipermedial Dinámico Campus Virtual UNR. Secretaría de Tecnologías Educativas y Gestión. UNR, Rosario (2010) disponible en: <http://rephip.unr.edu.ar/handle/2133/1390>.
3. Dey, A.K.; Salber, D.; Abowd, G.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, anchor article of a special issue on Context-Aware Computing. Human-Computer Interaction (HCI) Journal, Vol. 16 (2-4), pp. 97-166. (2001).
4. - Gell-Mann, M.: El quark y el jaguar. Aventuras en lo simple y lo complejo. Tusquets, Barcelona (1995).
- García, R.: Sistemas Complejos. Conceptos, métodos y fundamentación epistemológica de la investigación interdisciplinaria, Gedisa, Buenos Aires (2007).
5. Programa I+D+T “Dispositivos Hipermediales Dinámicos”, <http://www.mesadearena.edu.ar>. Proyecto PIP N° 718 (CONICET) “Obra Abierta: DHD para educar e investigar.” Dir. Dra. Patricia San Martín.
6. San Martín, P.; Sartorio, A.; Guarnieri, G.; Rodríguez, G.: Hacia la construcción de un dispositivo hipermedial dinámico. Educación e investigación para el campo audiovisual interactivo. Universidad Nacional de Quilmes Editorial, Buenos Aires (2008).
7. - Zeigler, B.; King, Tan Gon; Praehofer, H.: Theory of modeling and Simulation. Second edition, Academic Press, New York. (2000).
- Zeigler, B.: Theory of modeling and Simulation. John Wiley & Sons, New York. (1976).
8. Rivera, M.B.; Molina, H.; Olsina, L.: Sistema Colaborativo de Revisión para el soporte de información de contexto en el marco C-INCAMI, XIII Congreso Argentino de Ciencias de la Computación, CACIC. (2007).
9. Sartorio, A.; Cristiá, M.: First Approximation to DHD Design and Implementation. Clei electronic journal, Vol.12 N. 1. (2009).
10. Meyer, B.: Applying Design by Contract, IEEE Computer, 40-51. (1992).
11. Rodríguez, G.; San Martín, P.; Sartorio, A.: Aproximación al modelado del componente conceptual básico del Dispositivo Hipermedial Dinámico. XV Congreso Argentino de Ciencias de la Computación. CACIC 2009. San Salvador de Jujuy. (2009).
12. Rodríguez, G.: Desarrollo e implementación de métricas para el análisis de las interacciones del Dispositivo Hipermedial Dinámico. Jornadas Argentinas de Informática. JAIIO 2010, Caba. (2010).
13. PowerDEVs 2.0 Integrated Tool for Edition and Simulation of Discrete Event Systems. Desarrollado por: Esteban Pagliero, Marcelo Lapadula, Federico Bergero. Dirigido por Ernesto Kofman. (<http://www.fceia.unr.edu.ar/lsd/powerdevs/index.html>).
14. Rumbaugh, J.; Jacobson, I.; Booch, G.: The Unified Modeling Language Reference Manual. Addison Wesley Logman, Inc.; Massachusetts (1999).
15. Sartorio, A., Guarnieri, G., San Martín, P.: Students’ interaction in an e-learning contract contextaware application with associated metric”, Actas del INTED2007, International Technology, Education and Development Conference, IATED, Valencia, España. (2007).