

Condicionales DEVS en la coordinación de contratos sensibles al contexto para los DHD

Alejandro R. Sartorio^{1,2}
Guillermo L. Rodríguez¹

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

² Centro de Altos Estudios en Tecnología Informática, Sede Rosario, Universidad Abierta Interamericana, Ov. Lagos 944, 2000 Rosario, Argentina
sartorio@cifasis-conicet.gov.ar, guille@fceia.unr.edu.ar

Resumen. Teniendo en cuenta los aspectos tecnológicos y de diseño de los mecanismos para la inyección de las propiedades de coordinación de contratos sensibles al contextos en los Dispositivos Hipermediales Dinámicos (DHD), se propuso la creación de un nuevo condicional (Condicional DEVS) que permita establecer valores de verdad a partir de la inferencia de un modelo de eventos discretos (DEVS) que representa una métrica para determinar el grado de interactividad que se produce para cada participación en los DHD. A través de un caso de uso se muestra la implementación del modelo de integración propuesto que permite incluir nueva información de contexto a las reglas teniendo en cuenta solamente los coeficientes de las métricas intervinientes.

Palabras Clave: Dispositivo Hipermedial Dinámico – Coordinación de Contratos – Métricas – Eventos Discretos – TIC.

1 Introducción

El actual contexto físico-virtual que se construye a partir de la utilización de las Tecnologías de la Información y Comunicación (TIC) posibilita a los sujetos ser partícipes de redes sociotécnicas conformadas por una multiplicidad de componentes y relaciones, que se configuran y reconfiguran por las diversas interacciones en función de una gran diversidad de requerimientos. En este sentido, el Programa interdisciplinario de I+D+T “Dispositivos Hipermediales Dinámicos” (DHD) [1], radicado en CIFASIS (CONICET-UNR-UPCAM), estudia la complejidad evidente de las mencionadas redes, integrando aportes de diversas disciplinas como informática, educación, ingeniería, psicología y antropología, entre otras.

Se conceptualiza como Dispositivo Hipermedial Dinámico -DHD- a la red heterogénea [2] conformada por la conjunción de tecnologías y aspectos sociales que posibilitan a los sujetos realizar acciones en interacción responsable con el otro para

investigar, aprender, dialogar, confrontar, componer, evaluar, bajo la modalidad de taller físico-virtual, utilizando la potencialidad comunicacional, transformadora y abierta de lo hipermedial, regulados según el caso, por una “coordinación de contratos” [3].

En la implementación y optimización de un DHD para la producción y disseminación de conocimiento, los mecanismos de medición y evaluación suponen una de las actividades principales para el análisis y el aseguramiento de la calidad de lo que se desarrolla a través de los mismos.

Los procesos de medición son fundamentales dado que permiten cuantificar un conjunto de características deseadas acerca de un aspecto específico de algún ente en particular, proveyendo una visión más o menos detallada de su estado o condición. Por su parte, la evaluación interpreta los valores obtenidos en la medición. Para dichos procesos de medición y evaluación es necesario obtener datos cuantitativos, a partir de métricas de atributos de entes y la posterior interpretación de la medida a partir de indicadores [4].

Funcionalmente el DHD es conceptualizado como sistema complejo [5], en el cual los Participantes (P) a través del intercambio analítico y de producción de textos mediatizados en diversos tipos de formatos digitales, construyen las posibilidades y limitaciones de la mediación interdisciplinaria responsable en su área de incumbencia, siendo deseable que se pueda observar un paulatino cambio en su situación contextual. Al constatar que la característica primordial del DHD, es que las interacciones se deben a la ocurrencia asincrónica de eventos, hemos optado por el modelado con DEVS, Discrete Event System specification [6], considerándose además la gran adaptación del formalismo para modelizar sistemas complejos, y su simplicidad y eficiencia en la implementación de simulaciones.

Tecnológicamente el DHD está provista por un agregado de una pieza de software para la inyección de propiedades de coordinación de contratos sensibles al contexto [7]. Esta propiedad se logra a través de la implementación de contratos [8] con mecanismos de coordinación y componentes de sistemas context aware [9].

La utilización de reglas es parte esencial en la implementación de las acciones de los contratos y las tareas de coordinación. A su vez, las reglas están compuestas por los condicionales donde se centra parte de la lógica de adaptación que se requiere en los DHD. Algunos condicionales implementados requieren de mecanismos externos que colaboren en la composición de sus valores de verdad [10].

En este trabajo se define un nuevo tipo de condicional, denominado **Condicional DEVS**, para la inclusión de valores de verdad que puedan ser inferidos por medio de la implementación de un conjunto de métricas flexibles que contemplan las principales características de las interacciones de los participantes de los DHD. Tras esta introducción, en la sección 2 se identifican los elementos de los DHD en relación directa con los Condicionales DEVS. Luego, en la sección 3 se presenta un modelo de integración para el funcionamiento en una herramienta del framework SAKAI (para la justificación de dicha elección ver [7]). En la sección 4 se describen algunas de las características principales de las métricas y un caso de uso concreto. Para finalizar, se presentan las conclusiones y consideraciones generales.

2. Aspecto tecnológico de los DHD

En esta sección se describen aspectos tecnológicos y componentes de los DHD que intervienen en el modelo de integración entre las métricas DEVS y el framework de nuestra propuesta. De esta manera, se solucionan los requerimientos sobre adaptación dinámica mediante la construcción de un modelo de contrato orientado a la implementación de servicios sensibles al contexto.

El uso de contratos parte de la noción de Programación por Contrato ("Programming by Contract") de Meyer [8] basada en la metáfora de que un elemento de un sistema de software colabora con otro, manteniendo obligaciones y beneficios mutuos. En nuestro dominio de aplicación consideraremos que un objeto cliente y un objeto servidor "acuerdan" a través de un contrato, -representado con un nuevo objeto-, que el objeto servidor satisfaga el pedido del cliente, y al mismo tiempo el cliente cumpla con las condiciones impuestas por el proveedor. A su vez las decisiones de comportamiento partirán de los condicionales de las acciones de los contratos.

Como ejemplo de la aplicación de la idea de Meyer en nuestro dominio de sistemas e-learning planteamos la situación en que un usuario (cliente) utiliza un servicio de edición de mensajes (servidor) a través de un contrato que garantizará las siguientes condiciones: el usuario debe poder editar aquellos mensajes que tiene autorización según su perfil (obligación del proveedor y beneficio del cliente); el proveedor debe tener acceso a la información del perfil del usuario (obligación del cliente y beneficio del proveedor).

A partir de la conceptualización de contratos según Meyer se propone una extensión por medio del agregado de nuevas componentes para instrumentar mecanismos que permitan ejecutar acciones dependiendo del contexto. En aplicaciones sensibles al contexto [9], el contexto (o información de contexto) es definido como la información que puede ser usada para caracterizar la situación de una entidad más allá de los atributos que la definen. En nuestro caso, una entidad es un usuario (alumno, docente, etc.), lugar (aula, biblioteca, sala de consulta, etc.), recurso (impresora, fax, etc.), u objeto (examen, trabajo práctico, etc.) que se comunica con otra entidad a través del contrato.

En [2] se propone una especificación del concepto de contexto partiendo de las consideraciones de Dourish [11] y adaptadas al dominio e-learning, que será la que consideraremos en este trabajo. Contexto es todo tipo de información que pueda ser censada y procesada, a través de la aplicación e-learning, que caracterizan a un usuario o entorno, por ejemplo: intervenciones en los foros, promedios de notas, habilidades, niveles de conocimientos, máquinas (direcciones ip) conectadas, nivel de intervención en los foros, cantidad de usuarios conectados, fechas y horarios, estadísticas sobre cursos, etc.

En términos generales, la coordinación de contratos es una conexión establecida entre un grupo de objetos influidas por condicionales que representan parte de la lógica de adaptación, aunque en este trabajo se consideran sólo dos objetos: un cliente y un servidor. Cuando un objeto cliente efectúa una llamada a un objeto servidor (ej., el servicio de edición 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 [1]. En este trabajo en los condicionales de las reglas se representarán diferente tipo de información de contexto con distinto grado de representación y abstracción, donde se requieren mecanismos de inferencias basados en la recolección, representación y simulación.

A continuación se brindarán detalles sobre algunas de los componentes y relaciones esenciales para la integración de este modelo con el framework utilizado y con los módulos que instrumentan la coordinación de contratos.

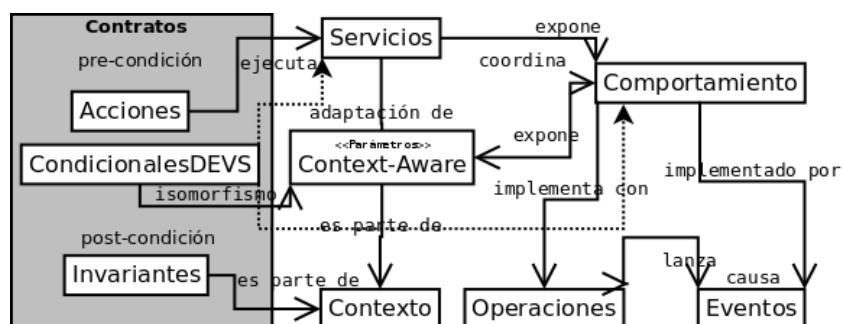


Fig. 1. Modelo de elementos y relaciones de los DEVS condicionales.

Un contrato que siga las ideas de Meyer contiene toda la información sobre los servicios que utilizarán los clientes. Para incorporar sensibilidad al contexto nuestros contratos deberán tener referencias sobre algún tipo de información de contexto para su utilización. En el diagrama de relaciones entre entidades mostrado en la Figura 1 se describen los elementos que componen el concepto de contrato sensible al contexto donde se tiene participación de los condicionales DEVS. La figura comienza con la representación de un contrato según Meyer donde se caracterizan los principales elementos que lo componen (pre-condiciones, acciones, pos-condiciones). La flechas salientes de la zona gris indican los dos tipos de relaciones (acción-servicio e invariante-contexto) que se debe instrumentar para incorporar un mecanismo que provea a los contratos de la característica de sensibilidad al contexto. En la porción derecha de la Figura 1 aparecen las entidades necesarias para obtener contratos sensibles al contexto. A continuación se explica cada uno de los elementos y su relación con los condicionales de las acciones.

- **Servicios:** En esta componente se representan los elementos necesarios para la identificación y clasificación de los servicios que pueden formar parte de las acciones de los contratos. Por ejemplo, nombre del servicio, identificadores, alcance, propósito, etc. En este caso existe una relación indirecta con el condicional DEVS establecida por la relación *ejecutar* entre la acción del contrato y el servicio.

- **Comportamiento:** El comportamiento de un servicio se logra a partir de combinar operaciones y eventos que son representadas con las componentes *Operaciones* y *Eventos*. De la misma manera el servicio puede ser implementado a través del uso de eventos, representados con el componente *Eventos*, que puede lanzar operaciones del componente *Operaciones*. Por ejemplo, de acuerdo con los roles (ej., alumno, instructor, docente, etc.) asignados a un usuario de una herramienta

involucrado en un determinado contexto del entorno (ej., si está en un espacio Foro) y del usuario (ej., si tiene permiso de moderador), la componente *Servicios* brinda distintas funcionalidades (ej., editar un mensaje), que son instrumentadas por medio de operaciones concretas (ej., guardar un mensaje en una tabla) y/o a través de la publicación o suscripción de eventos. Aquí se establece una relación directa con el Condicional DEVS teniendo en cuenta todas las acciones que dependan de valoraciones influidas por el contexto, representadas por la simulación a través de un modelos DEVS.

- **Parámetros Context-Aware:** Se denomina *Parámetros Context-Aware* a la representación de la información de contexto que forma parte de los parámetros de entrada de las funciones y métodos exportados por los servicios, estableciendo de esta manera una relación entre el componente *Servicios* y el componente *Parámetros Context-Aware*. Existe una relación isomórfica entre los valores usados en los Condicionales DEVS y los elementos del conjunto de *Parámetros Context-Aware*.

- **Contexto:** Para nuestro modelo este tipo de información es utilizada de dos maneras diferentes: en primer lugar para la asignación de los valores que toman los *Parámetros Context-Aware*; en segundo lugar esta información puede ser utilizada para definir los invariantes que se representan en los contratos. Nuevamente se establece una relación indirecta entre los Condicionales DEVS y el contexto mediada por su representación como elemento de los *Parámetros Context-Aware*.

2.1 Condicionales para la coordinación de contratos

Ahora a través de un diagrama UML se definen las clases utilizadas en la implementación de los Condicionales DEVS dentro de las reglas de los contratos, donde se mantienen las propiedades e influencias (relaciones entre elementos conceptuales) descriptas en la figura 1.

La figura 2 describe los elementos y relaciones relevantes en la creación de condicionales inferidos por métricas de interacción (sección 4) implementadas en un modelo de simulación DEVS integrado (sección 3).

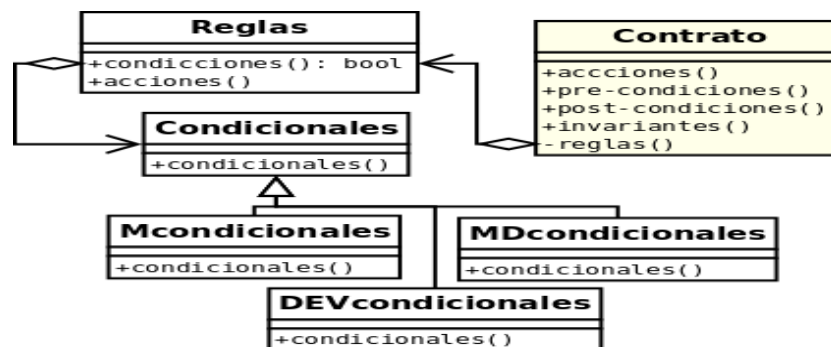


Fig. 2. Elementos y relaciones relevantes en la creación de los condicionales.

Partiendo de una de las propiedades de las reglas de los contratos sobre la posibilidad de definir comportamientos a través de parámetros context-aware e

inducidos por reglas donde se designan partes de las acciones del contrato. De esta manera, las reglas forman parte de un mecanismo de agregación encargado de la composición de diferentes tipos de condicionales, en los que se encuentran una familia de condicionales (**Condicionales DEVS**) conectados a los métodos que implementan las métricas definidas particularmente para la simulación de interacciones (sección 4). Las otras dos familias de condicionales representadas por las clases **Mcondicionales** y **MDcondicionales** se comportan de manera similares teniendo en cuenta el mismo modelo de integración propuesto [10].

A continuación se describen los aspectos principales que se tuvieron en cuenta en la integración de los anteriores sistemas de coordinación de contratos sensibles al contexto y extensiones de condicionales para la aplicación de un nuevo sistema de métricas de interacciones mediante un modelo de simulación DEVS.

3. Modelo conceptual de integración

Para implementar la invocación de métricas mediante métodos correctos, propusimos desde la perspectiva del rediseño e implementación computacional, un modelo de integración de muy bajo costo, sin cambios sustanciales ni en la arquitectura original ni en el código de la implementación dentro de la aplicación modificada en el proceso de inyección de las propiedades de coordinación de contratos sensibles al contexto [3].

El modelo conceptual de métrica pertenece al Modelo INCAMI (Information Need, Concept model, Attribute, Metric and Indicator: Información relevante, Modelo Conceptual, Atributos, Métricas e Indicadores) [12]. INCAMI es un framework organizacional, orientado a la medición y evaluación que permite economizar consistentemente, no sólo metadata de métricas e indicadores, sino también valores mensurables en contextos físicos.

Por medio de un diagrama UML, se representa un modelo general de integración, teniendo en cuenta experiencias vinculadas al agregado de nuevas componentes en determinadas implementaciones resueltas para sistemas e-learning similares al diseño del framework [2]. La integración se produce mediante la conexión de las reglas, a través de sus condicionales, con una métrica representada con un método. A su vez, la métrica es interpretada por un modelo DEVS diseñado para devolver valores de simulación [13].

En la figura 3 se puede observar lo correspondiente a cada una de las áreas mencionadas, representadas con colores diferentes. Además, se muestra que la principal componente para lograr la integración está representada por la incorporación de una relación de agregación entre la componente Contrato y la entidad Método. Los condicionales de las reglas de los contratos son invocados (mediante un método explícito relacionado con la noción de los Condicionales DEVS, por ejemplo, `getForum_theme`) por medio de un mecanismo de callback que permite la correcta invocación de la métrica.

La primera fase de la misma corresponde a la definición y especificación de requerimientos. Este módulo trata con la definición de la necesidad de información (es decir, el foco de la evaluación) y el diseño de los requerimientos no funcionales,

que servirán como guías para las actividades posteriores de medición y evaluación.

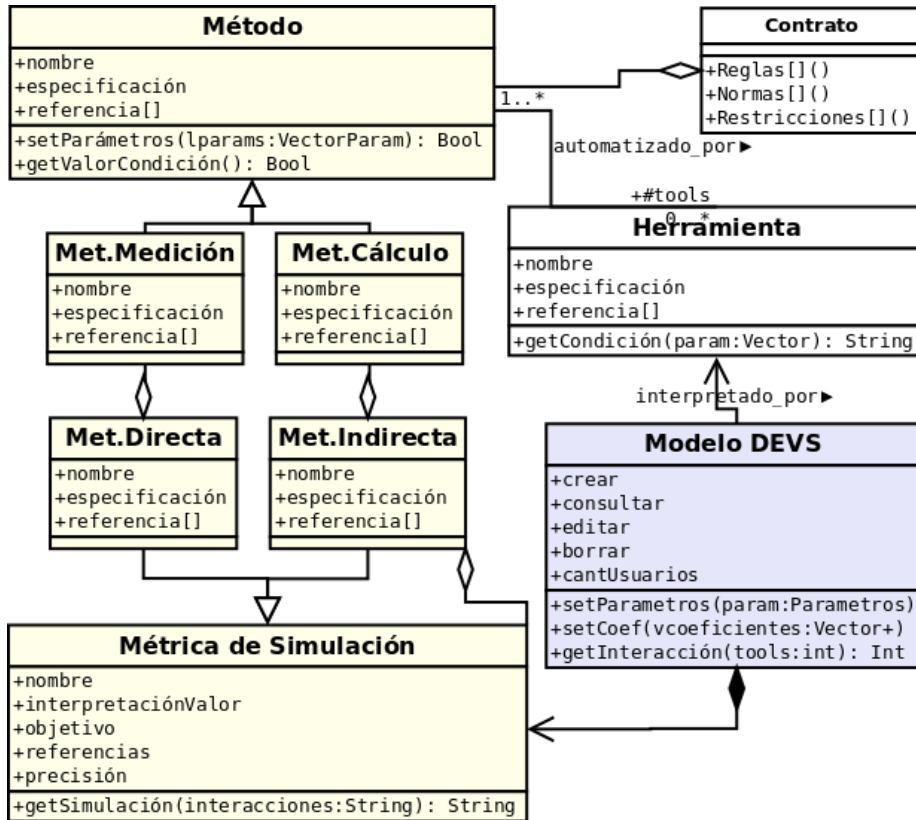


Fig. 3. Modelo de integración para contratos, métricas y modelo DEVS

Tomamos como punto de partida la descripción del Dispositivo Hipermedial Dinámico [14]. De esta manera se desprende que la información necesaria en nuestro caso es función de las interacciones de los participantes, las cuales estarán definidas por: id del participante, rol del mismo, Paquete Hipermedial (PH) sobre el cual participa, tipo de PH, herramienta sobre la cual realiza la interacción, tipo de herramienta, servicio con el cual interactúa y el tiempo, día y hora de la interacción [13].

El principal objetivo de la implementación de la evaluación global permite mayores niveles de flexibilización para los valores de los indicadores globales y parciales, a partir de los valores de indicadores elementales utilizando el modelo de agrupamiento obtenido para efectuar el cálculo. En este proceso, dichos valores deben ser acordados y consensuados por expertos con experiencia en el uso de este tipo de sistemas. El seteo de los coeficientes serán establecido a través de la interface *setcoeficient*, de la clase *Modelo DEVS* (figura 3). En cada caso, el valor resultado brinda una medida sobre el grado de interactividad de la participación. El responsable

de la evaluación pueda dar valor a los diversos coeficientes subrayando aquel atributo que considere más importante en el proceso. En cada caso, el valor resultado brinda una medida sobre el grado de interactividad de la participación. Este valor se obtiene a través de la interfase *getInteracción*, que toma como argumento en este caso un número entero que identifica la herramienta.

Por último mencionamos que la interfase *setParametros* queda reservada para posibilitar diferentes relaciones algebraicas dentro de la métrica potenciando el nivel de expresión de la misma.

La interpretación y manipulación de los resultados de interacciones resueltos en el *Modelo DEVS* es manipulado por una herramienta representada por la clase *Herramienta*. A su vez la herramienta es la encargada de brindar la información necesaria sobre los parámetros que necesita la clase *Método* que es utilizada como argumento de la función *setParámetro*. El método *getValorCondición* representa los valores de verdad del condicional que formará parte de la regla explícita representada por el método *reglas* de la clase *Contrato*.

Técnicamente la *Herramienta* es una aplicación que respeta la arquitectura del framework colaborativo SAKAI [7], 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. En nuestros desarrollos se utilizó iBatis (<http://ibatis.apache.org/>) para el acceso a datos y XML DOM [<http://msdn.microsoft.com/en-us/library/ms764730%28VS.85%29.aspx>] Parser para la parametrización de la función transferencia. Los demás componentes tecnológicos que complementan el desarrollo cumplen los estándares del framework, en este caso se utilizaron 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 se describe un caso de uso para ejemplificar las distintas etapas que se deben cumplimentar como usuario para la activación de las propiedades que brindan los **Condicionales DEVS** dentro de los contratos sensibles al contexto (sección 2). Además, se brindarán detalles funcionales sobre el uso de la herramienta Sakai que implementa la conexión entre los métodos de la métrica y el modelo de simulación DEVS, manteniendo la perspectiva de un usuario final.

Se comienza con el diseño de las reglas de los contratos y sus correspondientes condicionales DEVS. Para esta etapa tomaremos como referencia el esquema para el diseño de contrato propuesto en UWATc en la última etapa del diseño de procesos e-learning Web [10].

4.1 Representación de los Condicionales DEVS a través de UWATc

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 metamodelo de la figura 1.

Teniendo en cuenta el diagrama siguiente, en primer lugar (item 1) se identifican los objetos participantes en el contrato; en dicho ejemplo *DiscussionAction* y *UserAction* hacen referencia a dos clases reales perteneciente a la implementación de la herramienta Foro y Usuarios de una aplicación, 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 2) 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. Las siguientes filas (item 2) 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 (item 3). Siguiendo con el ejemplo, en la parte del condicional *u.contexto = 'l1; p1; docente; r1; c1; IT1; IPH1'* 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]). Además, los últimos dos valores *IT1* y *IPH1*, representan el nivel de interacciones que un determinado usuario y de una herramienta particular, respectivamente. A diferencia de los otros valores, *IT1* y *IPH1* serán comparados con la resultante de la aplicación de una métrica de interacción a través de un elemento externo con interfaz para comunicarse a través de los Condicionales DEVS.

Contrato: Edición			
1	Participantes:	d: DiscussionAction u: UserAction	
2	Param. c-a: Servicios: Pre-Cond: Pos-Cond:	state, portlet, rundata, context initState() existe < contexto > modifica < contexto > contextidentifier, identifier getIdentifier() existe < contexto >	
3	Reglas:	Si u.contexto='p1;d;r1;c1;IT1,IPH1' entonces d.showMessage(data,string)	
Condicionales			
4	DEVS	Valores	
		IT1	C1..4: valores de servicios B1..4: valores de las herramientas , donde $B_i = C_1 * C_2 * C_3 * C_4$
		IPH1	PH: valor para el PH, donde $PH = B_1 * B_2 * B_3 * B_4$
5	MD	-	
6	M	-	
7	Comentario	Del Item 4: Especificación de las Métricas: Herramienta - PH - DHD [13].	

En el ítem 4 se describe la forma de representar en el diagrama que dichos valores pertenecen al tipo de Condicional DEVS y la definición de los coeficientes que formarán parte de la métrica y serán alcanzado por los valores de los parámetros de las interfaces de los métodos que la representan.

En cuanto a la acción de la regla de coordinación, continuando con el mismo ejemplo, en el ítem 3 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 al que hace referencia.

De esta manera, a través de la extensión del diagrama de contrato de la etapa 4 en el proceso de diseño de los `Pe-Irn` [10] se logra describir los principales componentes que se tienen en cuenta en el diseño e implementación de los Condicionales DEVS para los casos de uso similares al presentado.

4.2 Ejemplo de ejecución de la métrica en PowerDEVS

Atendiendo a lo expuesto, seguidamente se implementan lo explicado en el entorno `PowerDEVS` [15] teniendo en cuenta el mismo caso de uso introducido en esta sección. En la figura 3, la clase *Modelo DEVS* contiene las interfaces que son implementadas a través de la herramienta `PowerDEVS` para la interpretación de las métricas presentes en el comentario del diagrama de contrato.

Cada métrica directa tiene asociado un método de medición claramente especificado. Las colecciones de datos son capturados desde la bases de datos (en este caso `MySQL`). Luego los datos son formateados para posibilitar su lectura desde el entorno.

En la figura 4 se muestran, a manera de ejemplo, los resultados obtenidos de Nivel de Interactividad para cada participación a través del tiempo, en los meses de Noviembre-Diciembre para un curso seleccionado en el 2009, del Campus Virtual de la Universidad Nacional de Rosario (<http://www.campusvirtualunr.edu.ar>).

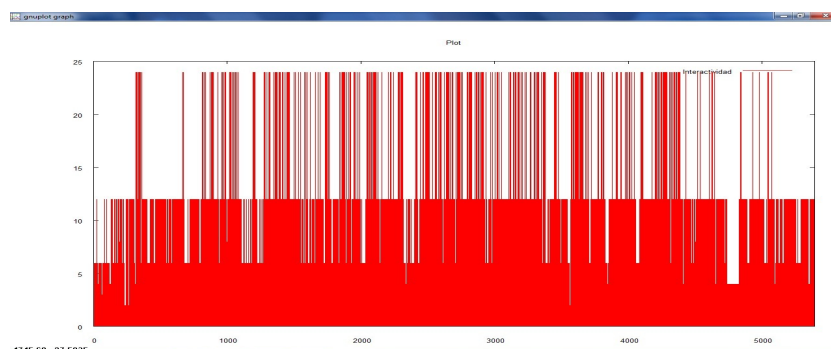


Fig. 4. Resultados obtenidos en el entorno `PowerDEVS`.

Cabe mencionar que los resultados obtenidos son los globales del DHD, pero que sin embargo se disponen a su vez, los valores parciales tanto a nivel Paquete Hipermédial, como a nivel Herramienta individual, (por cuestiones de espacio no

mostramos aquí dichos gráficos). Los mismos se exportan a un archivo que relaciona el número de participación, con su nivel de interactividad.

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 interactividad como parámetro context-aware de los contratos [3]. Podremos entonces, establecer un lazo de retroalimentación entre las prácticas efectuadas en los entornos colaborativos, informadas en el Registro de Actividad y las acciones que devengan de los contratos.

5. Conclusiones

En este trabajo se fundamentó la posibilidad de extender las propiedades expresivas de las reglas de coordinación de contratos de los DHD a partir de los resultados de un mecanismo externo. La propuesta de integración expuesta sigue respetando las líneas de diseño e implementación establecidas por el modelo de los DHD.

Ahora se tiene un nuevo mecanismo para la escritura de las reglas de los contratos que permitirá ahorrar esfuerzo en el diseño de los condicionales que verifiquen información de contexto parametrizado. Además, se evita la necesidad de tener conocimiento sobre las leyes (diseño) de las métricas y su implementación, con sólo tener información sobre la definición de los coeficientes que participan en la métrica.

Referencias

1. Programa I+D+T “Dispositivos Hipermediales Dinámicos”, <http://www.mesadearena.edu.ar>
2. San Martin, 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).
3. Sartorio, A.; Cristiá, M.: Primera aproximación al diseño e implementación de los DHD. XXXIV Congreso Latinoamericano de Informática, CLEI. (2008).
4. 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).
5. Gell-Mann, M.: El quark y el jaguar. Aventuras en lo simple y lo complejo. Tusquets, Barcelona. (1995).
6. - 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).
7. Sartorio, A.; Cristiá, M.: First Approximation to DHD Design and Implementation. Clei electronic journal, Vol.12 N. 1. (2009).
8. Meyer, B.: Applying Design by Contract, IEEE Computer, 40-51. (1992).
9. 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).
10. Sartorio, A.: Un modelo comprensivo para el diseño de procesos en una Aplicación E-

Learning. XIII Congreso Argentino de Ciencias de la Computación. CACIC 2007. ISBN 978-950-656-109-3. (2007).

11. Dourish, P.: What we talk about when we talk about context. *Personal and Ubiquitous Computing*, vol. 8, No 1, Roma, pp. 19-30, disponible en: <http://www.springerlink.com/content/y8h819me8yabycl3/>. (2004).
12. Olsina, L., Rossi, G. Measuring Web Application Quality with WebQEM, *IEEE Multimedia*, 9(4), pp. 20-29. (2002).
13. 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).
14. 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).
15. 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/lcd/powerdevs/index.html>).