

Aplicación de los condicionales DHD en la herramienta del foro de Sakai

Alejandro R. Sartorio ^{1,2}

Marcelo A. Vaquero ²

Guillermo Rodríguez ^{1,2}

Daniel Tedini ²

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, CAETI, Universidad Abierta Interamericana, Sede Rosario, Ov. Lagos 944, 2000 Rosario, Argentina

{Sartorio, Guille}@fceia.unr.edu.ar, {Marcelo.Vaquero, Daniel.Tedini}@uai.edu.ar

Resumen. En este trabajo se brinda la información necesaria de la aplicación de un modelo de condicionales de la infraestructura de los contratos sensibles al contexto para el Dispositivo Hipermedial Dinámico (DHD). Partiendo de la implementación tecnológica en el entorno colaborativo SAKAI se muestra una forma de adaptar los servicios de la herramienta foro a la información de contexto de los usuarios. De esta manera, se describen un diseño general de integración para ser utilizado en implementaciones de cualquier tipo de condicionales para los DHD.

Palabras Clave: Coordinación de Contratos – Sistemas sensibles al contexto – Dispositivo Hipermedial Dinámico – 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 “Dispositivos Hipermediales Dinámicos” [1], radicado en CIFASIS (CONICET-UNR-UPCAM), estudia la complejidad evidente de las mencionadas redes, integrando aportes de diversas disciplinas como informática, educación, ingeniería y psicología, entre otras.

Se conceptualiza como Dispositivo Hipermedial Dinámico (DHD) [2] a la red heterogénea 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].

Funcionalmente, el DHD es conceptualizado como sistema complejo [4], en el cual los participantes realizan acciones de participación mediadas por diversas tecnologías. Estas interacciones se efectúan por medio de servicios provistos por herramientas específicas agrupadas según el espacio colaborativo utilizado. Además, se busca que los efectos de dichas acciones estén condicionados por la información de contexto de los participantes que la involucran.

Para esto, tecnológicamente el DHD está provisto por el agregado de una pieza de software diseñada para la inyección de propiedades de coordinación de contratos sensibles al contexto [5]. Esta propiedad se logra a través de la implementación de contratos [6] con mecanismos de coordinación y componentes de sistemas sensibles al contexto [7].

La utilización de reglas es parte esencial en la implementación de las acciones que contienen los contratos y las tareas de coordinación. A su vez, las estructuras de las reglas contienen condicionales donde se establece parte de la lógica de adaptación a los requerimientos funcionales de los DHD. En este trabajo se retoman los tres tipos de condicionales lógicos diseñados con el propósito de representar valores de verdad que dependan de la información de contexto de usuarios [19]. De esta manera se pretende diseñar una estructura conceptual que permita implementar estos condicionales en el marco tecnológico del DHD.

Tras esta introducción, en la sección 2 se identifican los elementos tecnológicos del DHD teniendo en cuenta su relación con los condicionales. Luego, en la sección 3, se presentan los tres tipos de condicionales con sus principales características y modelos de integración dentro del *framework* SAKAI (<http://www.sakaiproject.org>) [8]. En la sección 4 se describe un modelo conceptual genérico de condicionales junto a un ejemplo de integración. Por último, en la sección 5 se presentan las principales conclusiones generales.

2. El uso de contratos en el DHD

El uso de contratos en el DHD parte de la noción de Programación por Contrato (*Programming by Contract*) de Meyer [6] 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. De esta manera, las decisiones de comportamiento de los servicios se verán influenciadas por el valor de verdad de las instancias de los condicionales que integren al contrato.

Como ejemplo de la aplicación de la idea de Meyer [9] en nuestro dominio de tecnologías de la información y comunicación planteamos el escenario 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 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, 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 (participantes, coordinadores, etc.), lugar (casa, universidad, parque, etc.), recurso (impresora, fax, etc.), u objeto (archivos de texto, fotos, videos digitales, etc.) que se comunica con otra entidad a través del contrato.

En [9] se propone una especificación del concepto de contexto partiendo de las consideraciones de Dourish [10] y adaptadas al dominio de las tecnologías de la información y comunicación, que será el punto de partida del actual trabajo. Contexto es todo tipo de información que pueda ser censada y procesada, a través de una aplicación, que caracteriza a un usuario o entorno, por ejemplo: intervenciones en foros, participaciones en wikis, habilidades, niveles de conocimientos, direcciones *ip* conectadas, cantidad de usuarios conectados, fechas y horarios, 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. Cuando un objeto cliente efectúa una llamada a un objeto servidor, 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. Los condicionales de las reglas 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 través de un diagrama UML se definen las clases utilizadas en la implementación de los condicionales dentro de las reglas de los contratos. La Figura 1 describe los elementos y relaciones relevantes en la composición de condicionales.

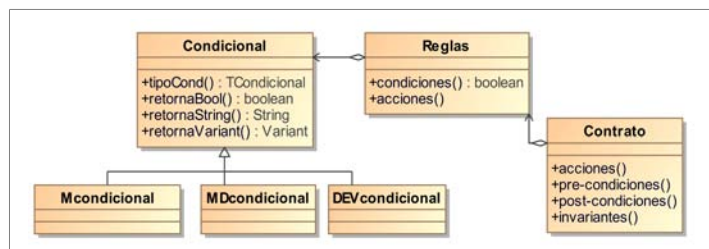


Fig.1. Elementos y relaciones relevantes en la creación de condicionales.

Dentro del mecanismo de configuración de condicionales, el contrato presenta en su interfaz métodos para la ordenación de acciones y reglas; además de las que cubren las propiedades inherentes a la definición original de contrato sobre la configuración de *pre-condiciones*, *post-condiciones* e *invariantes*. De esta manera se representa a las reglas anteriormente mencionadas como una clase de agregación del contrato. Así se determina una nueva clase para la representación de los condicionales. Las reglas contienen referencias a las acciones de los contratos por medio de la interfaz *acciones*.

Se decide representar a los condicionales como objetos de primera clase con el propósito de establecer un nuevo grado de abstracción que permitirá conectar a los contratos a subsistemas externos que le proporcionen nuevos mecanismos de adaptabilidad, dinamismo e interpretación.

De esta manera, teniendo en cuenta las experiencias de diseño e implementación del uso de condicionales [11] [12] [13] se extienden al objeto condicional en tres tipos diferentes. Cada uno de ellos hereda la interfaz *Condicional*, encargada de establecer las presentaciones para el tipo de dato necesario en las reglas.

El primer paso, es lograr la construcción de las reglas del contrato y que los condicionales representen criterios de decisiones sobre aspectos relevantes de los procesos didácticos, investigativos, de producción y/o de gestión mediatizados por un DHD; por ejemplo: un participante puede adquirir un servicio determinado de una herramienta a partir de la evaluación de una condición representada como condicional de una regla.

En general, a estas reglas debemos diseñarlas con el cuidado de no incorporar redundancias, ambigüedades o incoherencias; tanto entre las propias reglas de un contrato como con otras reglas implícitas que se desprenden de los servicios. Entonces, definimos a las *Reglas* del contrato como un conjunto de condiciones, acciones y prioridades. La condición es un expresión booleana sobre relaciones (mayor, menor, igual, distinto, etc.) entre parámetros y valores concretos. Las acciones conforman un conjunto de asignaciones de valor a otros parámetros también definidos por el tipo de regla. Algunos de los parámetros de las acciones deben ser “métodos de cálculo” que permiten cambios en el comportamiento de los servicios en los cuales estas reglas son aplicadas. La prioridad permite simplificar la cantidad de reglas que se deben escribir: en lugar de la escritura de una regla para cada

combinación de posibilidades de los valores de los parámetros, se asegura que dos reglas no puedan ser ejecutadas simultáneamente. Por ejemplo: el usuario podría escribir una prioridad baja para todas las reglas y luego con prioridades altas ir identificando las excepciones para el caso configurado inicialmente. En síntesis, las reglas son ejecutadas mediante un orden de prioridades.

Entonces, las reglas forman parte de un mecanismo de agregación encargado de la composición de diferentes tipos de condicionales: *Mcondicional*, *MDcondicional* y *DEVScndicional*, que se comportan de manera similar teniendo en cuenta diversos modelos de integración que explicitaremos en las secciones siguientes.

A continuación, se analizarán las estructuras de tres tipos de condicionales con el propósito de abstraer características de su conformación. En particular se identificarán tipos de elementos, patrones, sub-estructuras y relaciones que sean necesarias para la composición con los contratos.

3. Los tipos de condicionales

Retomando la sección anterior y de manera general a través de los elementos utilizados en el diseño de la Figura 1 para la creación de condicionales, se identifica como la primer característica de diseño, a la estructura que define los diferentes tipos de condicionales para el DHD [16,17,18.19].

Luego, partiendo de las necesidades adaptativas de los contratos y ante determinados casos de uso, se diseñaron diferentes implementaciones de condicionales. Estas diferencias tienen que ver con cuestiones de diseño en los que algunos aspectos pueden ser generalizados, posibilitando una representación más genérica de los condicionales.

Partiendo de los avances para la implementación del modelo conceptual de [19] se propone una método de inyección del modelo en la herramienta foro del *framework* Sakai. De esta manera, en la Figura 2 se representa un diagrama de las principales clases que integran la herramienta original del foro. En este caso, se toma la clase *BaseDiscussionMessage*, que extiende las operaciones de *BdDiscussionService*, para reimplementar el método *addDiscussionMessage()* encargado de ingresar el texto de una entrada de usuario al foro. A su vez, *addDiscussionMessage()* usa dos operaciones de *DiscussionMessageEdit* que utilizan servicios bases colaborativos extendiendo la infraestructura provista por las clases *Message* y *Edit*.

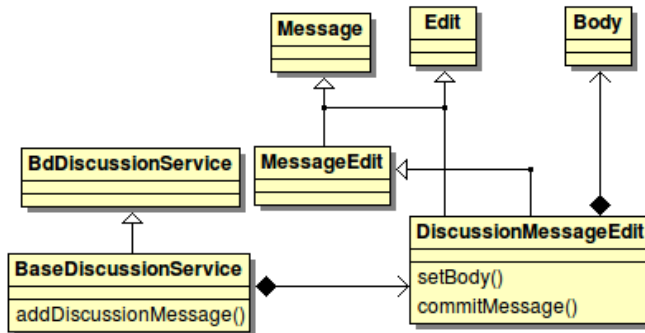


Fig. 2. Representación de los elementos de diseño de condicionales.

4. Inyección de los condicionalesDHD

En esta sección se presenta un diseño conceptual e información necesaria para la inyección de condicionales en una herramienta del DHD.

En este caso, se propone un diseño de integración para conectar un subsistema de configuración (*Calculo*) para la instanciación de los condicionales de las reglas de contratos.

La Figura 3 establece el diseño propuesto para la implementación de condicionales. Se define un módulo para efectuar los cálculos finales que determinan el valor de verdad del condicional (*Calculo*). Otro módulo es el encargado de la recolección y toma de datos (*TomarDatos*), extendiéndose para los casos particulares donde es necesario contar con estructuras de datos (*Estructuras*) conteniendo métodos que implementan cada una de ellas.

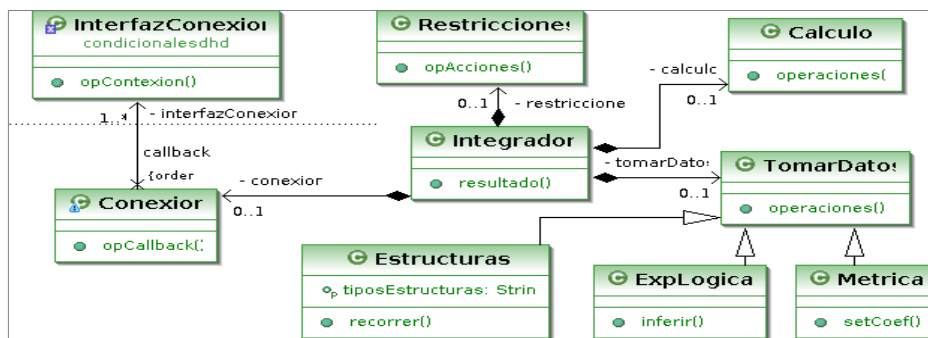


Fig. 3. Modelo de diseño conceptual de condicionales para contratos sensibles al contexto.

Además, un módulo aparte se configura para describir todas las restricciones que

debe cumplir el condicional (*Restricciones*), teniendo en cuenta su utilización dentro de las reglas de los contratos, con el propósito de no incurrir en contradicciones o inconsistencias en relación a las precondiciones, poscondiciones e invariantes.

Las conexiones con otros subsistemas, por ejemplo, el subsistema sensible al contexto representado, se encuentran encapsuladas en otro módulo de conexión (*Conexion*). De esta manera, se implementa un “callback” del método perteneciente a la interfaz de un subsistema externo.

4.1 Ejemplo

En la Figura 4 se muestra un ejemplo reducido de laboratorio para sistemas web colaborativo sensibles al contexto. Para esto exponemos un diagrama de secuencia que interpreta la ejecución de una regla del contrato que utiliza uno de los tipos de condicionales respetando el diseño original a través del objeto *BdDiscussionService*, y el modelo conceptual de integración a través de los objetos *Condicional*, *Metodo* y *TipoMetrica* [19]. De esta manera queda evidenciada la ventaja de contar con un modelo de diseño que facilite la manipulación de las reglas de los contratos sensibles al contexto.

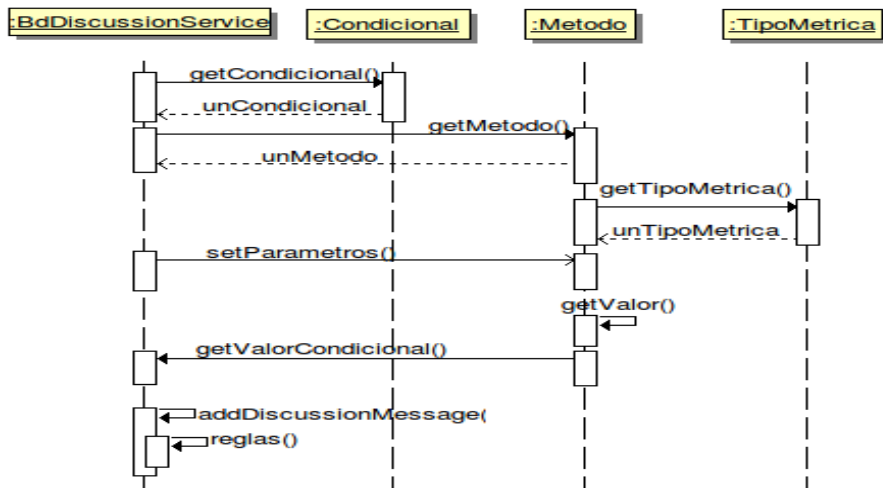


Fig. 4. Secuencia de invocación de los valores de verdad.

Luego, usaremos como referencias un fragmento de código descrito en el caso de estudio de la sección 5 de [5] para ejemplificar el impacto de esta propuesta. En este caso, se muestra que es necesario influir solamente sobre los conectores de la infraestructura de coordinación propuesta. La forma correcta de hacerlo es caracterizar la siguiente porción de código como una interfaz de conexión:

```
public CrdPartnerRules messageEdit_rules(string texto,Student c) throws
DiscussionException, CrdExFailure {return new CrdPartnerRules (this);}
```

Entonces, se debe agregar a esta función las siguientes líneas de códigos que posibilitarán cambiar las referencias de los condicionales de las reglas de los contratos:

```
Integrador unIntegrador = (Integrador) nuevoIntegrador(); ...
unValorCondional = unIntegrador.resultado(unDato, unCalculo);
Conexion unConexion=(Conexion) generarConexion (unValorCondional);...
InterfazConexion unInterfazConexion = (InterfazConexion) nueva;...
unConexion.setConexion(unaInterfazConexion.metodoConexion());..
```

De esta manera se debe agregar estas líneas dentro del código fuente de la operación *addDiscussionMessage()* del la clase *DiscussionMessage*. Para este ejemplo particular, se reemplaza la línea 5. Este procedimiento se denomina inyección de condicionalesDHD y se puede aplicar en reemplazo de cualquier porción de código que representa la invocación de un servicio base [13,20] de las herramientas del *framework* Sakai.

```
public DiscussionMessage addDiscussionMessage(String category, String
subject, boolean draft, String replyTo,List attachments, String body)
throws PermissionException
{
1. DiscussionMessageEdit edit=(DiscussionMessageEdit) addMessage();
   ...
2. edit.setBody(body);
3. header.replaceAttachments(attachments);
   ...
4. commitMessage(edit);
5. return edit;
6. } // addDiscussionMessage
```

5. Conclusiones

Partiendo de los avances de diseño e implementación de una infraestructura para la integración de condicionales para los contratos sensibles al contexto del DHD, se brindaron información necesaria para la inyección en el servicio base de edición que implementa la herramienta foro del *framework* Sakai.

Como principal aporte de este trabajo se incorpora una extensión de las propiedades de los contratos sensibles al contexto para su adaptación en la capa de servicio de un *framework* web colaborativo particular.

Referencias

1. Programa I+D “Dispositivos Hipermediales Dinámicos”, radicado en el Centro Internacional Franco Argentino de Ciencias de la Información y Sistemas (CIFASIS: CONICET-UNR-UPCAM). <http://www.mesadearena.edu.ar>. Directora: Dra. Patricia San Martín.
2. San Martín, P.: *Hacia un dispositivo hipermedial dinámico. Educación e investigación para el campo audiovisual interactivo*. Bernal: Universidad Nacional de Quilmes Editorial. 2008.
3. Sartorio, A. y Cristiá, M.: *Primera aproximación al diseño e implementación de los DHD*. XXXIV Congreso Latinoamericano de Informática, CLEI 2008. 2008.
4. Gell-Mann, M.: *El quark y el jaguar. Aventuras en lo simple y lo complejo*. Barcelona: Tusquets. 1995.
5. Sartorio, A. and Cristiá, M.: *First Approximation to DHD Design and Implementation*. Clei electronic journal, Vol.12, N° 1. 2009.
6. Meyer, B.: *Applying Design by Contract*. IEEE Computer Society Press, Volume 25 Issue 10. pp. 40-51. 1992.
7. Dey, A.K., Salber, D. and 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.
8. 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. 2007.
9. Sartorio, A. y San Martín, P.: *Sistemas Context-Aware en dispositivos hipermediales dinámicos para educación e investigación, en San Martín, P.: Hacia un dispositivo hipermedial dinámico. Educación e investigación para el campo audiovisual interactivo*. Bernal: Universidad Nacional de Quilmes Editorial. 2008.
10. Dourish, P.: *What we talk about when we talk about context*. Personal and Ubiquitous Computing, Vol. 8, N° 1. pp. 19-30. 2004.
11. Sartorio, A., Rodríguez, G. y Vaquero, M.: *Condicionales DEVS en la coordinación de contratos sensibles al contexto para los DHD. XVI Congreso Argentino de Ciencias de la Computación, CACIC 2010*. 2010.
12. Sartorio, A.: *Los contratos context-aware en aplicaciones para educación e investigación, en San Martín, P.: Hacia un dispositivo hipermedial dinámico. Educación e investigación para el campo audiovisual interactivo*. Bernal: Universidad Nacional de Quilmes Editorial. 2008.
13. Sartorio, A., Rodríguez, G. y Vaquero, M.: *Investigación en el diseño y desarrollo para el enriquecimiento de un framework colaborativo web sensible al contexto*. XIII Workshop de Investigadores en Ciencias de la Computación, WICC 2011. 2011.
14. Rivera, M.B., Molina, H. y 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. 2007.
15. Olsina, L. and Rossi, G.: *Measuring Web Application Quality with WebQEM*. IEEE Multimedia, 9(4). pp. 20-29. 2002.
16. Rodríguez, G.: *Desarrollo e implementación de métricas para el análisis de las interacciones del Dispositivo Hipermedial Dinámico*. Jornadas Argentinas de Informática, JAIIO 2010. 2010.
17. 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. Disponible en: <http://www.fceia.unr.edu.ar/lzd/powerdevs/index.html>
18. Rodríguez, G.: *SEPI-DHD: Herramienta integrada para el Seguimiento y Evaluación de los Procesos de Interactividad del DHD*, en: San Martín, P. y Traversa, O.: *El Dispositivo Hipermedial Dinámico Pantallas críticas: I+D+I para la Formación Superior en Crítica y Difusión de las Artes*. Ciudad Autónoma de Buenos Aires: Santiago Arcos. 2011.
 19. Sartorio, A., Rodríguez, G., Vaquero, M.: *Hacia un diseño general de integración de condicionales para los contratos sensible al contexto del DHD*. CACIC 2012. 2012
 20. Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V.P.: *Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services*. Internet Computing, IEEE (Volume:11 , Issue: 3). Page(s): 28 – 35. 2007