

Diseño de un Profile para el Modelado de Aplicaciones Paralelas y Concurrentes

Universidad Nacional de La Matanza
Departamento de Ingeniería e Investigaciones Tecnológicas
Instituto de Investigación y Desarrollo
Florencio Varela 1704. San Justo, Buenos Aires, Argentina

Daniel A. Giulianelli
dgiulian@unlam.edu.ar

Rocío A. Rodríguez
rodri@unlam.edu.ar

Pablo M. Vera
pablovera@unlam.edu.ar

Victor M. Fernandez
vfernandez@unlam.com.ar

Contexto

Este artículo se basa en el trabajo realizado por el equipo de investigación de MOA (modelado de aplicaciones) de La Universidad Nacional de La Matanza. Este equipo trabaja en el desarrollo de extensiones, es decir nuevos artefactos UML (Lenguaje Unificado de Modelado) para la mejor representación de problemáticas de aplicaciones paralelas y concurrentes.

Resumen

UML es un lenguaje de representación de amplio propósito general, el cual para tal fin cuenta con vocabulario gráfico. En algunos casos cuando se quiere modelar un tipo de aplicación particular el vocabulario gráfico de UML resulta ser muy reducido. Por esta razón es necesario extender el lenguaje con nuevos artefactos que permitan modelar las características particulares del dominio en cuestión.

El equipo de investigación trabaja con aplicaciones paralelas las cuales poseen procesos concurrentes y distribuidos. Al intentar modelar este tipo de aplicaciones con UML surge la necesidad de crear nuevos artefactos, los cuales deben ser especificados formalmente para lo cual se genera un Profile.

Palabras clave: UML, Dominio, Artefactos, Extensiones, Aplicaciones, Procesos, Paralelos, Concurrentes, Distribuidos, Recursos, Profile, Meta-Modelado.

1. Introducción

OMG (Object Management Group) ha creado profiles como solución para la falta de

elementos de representación de dominios específicos. Un profile es un mecanismo para extender un lenguaje a fin de expresar conceptos más específicos de ciertos dominios de aplicaciones. Según OMG “un profile es un subconjunto del meta-modelado de UML, este subconjunto del meta-modelado determina las reglas para representar a este subconjunto del UML” [3]. Para aclarar la definición es posible considerar el siguiente ejemplo: Si a través de UML se modela el diagrama de actividades con un diagrama de clases, este diagrama de clases que representa características del diagrama de actividades es un meta-modelado de UML, un profile es un subconjunto de este meta-modelado.

Al momento de modelar para dominios específicos surgen problemas en la expresividad de UML, por lo que es necesaria la extensión del lenguaje. Esta extensión del lenguaje permitirá crear nuevos artefactos dedicados a una tarea o con un significado determinado en el dominio de la aplicación y por consiguiente permitirá modelar aquellas características que no eran contempladas por la concepción original de UML. Por esta razón UML nos provee un mecanismo de extensibilidad para poder ampliar el vocabulario, estos mecanismos se encuadran dentro de la definición de los profiles:

- Estereotipos: Permiten la creación de nuevos tipos de bloques de construcción que derivan de otros existentes pero no son específicos de un problema particular. Estos son definidos por un nombre y un grupo de elementos del meta-modelado. Los estereotipos representan una nueva característica agregada al UML para extender el lenguaje.

- **Valores Etiquetados:** Los valores etiquetados son propiedades nuevas para elementos existentes, estos son meta-atributos que son asociados a una meta-clase de un meta-modelado extendido del profile. Cada valor etiquetado tiene un tipo y es asociado a un estereotipo.
- **Limitaciones o Restricciones:** Forman reglas (de consistencia o de negocios) sobre los elementos y sus propiedades. Las limitaciones son asociadas a los estereotipos, imponen condiciones a los elementos del meta-modelado que fueron estereotipados. Las limitaciones son escritas en un lenguaje natural denominado OCL (Object Constraint Language).

En el sitio de OMG es posible encontrar los siguientes ejemplos de Profiles extendidos para propósitos particulares [2]:

- **UML Profile for CORBA ®:** Permite expresar la semántica de CORBA a través de herramientas UML.
- **UML Profile for Data Distribution:** El propósito de este Profile UML es brindar soporte al análisis y diseño de sistemas orientados a objetos usando servicios de datos distribuidos.
- **UML Testing Profile:** Define un lenguaje para diseño, visualización, especificación, análisis, construcción y documentación de artefactos de testing de sistemas. Este profile es un lenguaje de modelado de testing que puede ser usado con la mayoría de los objetos y componentes y a la vez ser aplicado al testing de sistemas de varios dominios de aplicación. El UML Testing profile puede ser usado de manera autónoma para el manejo de artefactos de test o ser integrado a UML para el manejo y artefactos de test juntos.
- **UML Profile for Enterprise Application Integration (EAI):** El objetivo de este profile es simplificar la integración de aplicaciones estandarizando los metadatos de invocación y translación de información entre aplicaciones.
- **UML Profile for System on a Chip (SoC):** Esta especificación define un profile para sistemas sobre chip, permite la

representación jerárquica de módulos y canales, los cuales son elementos fundamentales de los sistemas sobre chips.

2. Línea de Investigación y Desarrollo

Bajo la línea inicial de investigación que consistió en analizar las extensiones que resultan necesarias para modelar aplicaciones con PRocesos COncurrentes y DIstribuidos se realiza un vocabulario gráfico que conforma un lenguaje al que se denomina PROCODI. Luego se desarrolla un profile que permita definir formalmente al nuevo lenguaje. PROCODI se basa en UML e incorpora nomenclatura para:

- Acceso a los recursos compartidos (Semáforos y Monitores).
- Diferencia entre mensajes y RPC (Remote procedure Call – Llamada a procedimientos Remotos).
- Identificación de área de memoria compartida.
- Cardinalidad para tareas que se harán en forma idéntica en distintos threads.
- Timer para aquellos casos en que las acciones repetitivas tienen un tiempo establecido de repetición.
- Diferenciación entre RPC Sincrónicos y Asincrónicos.
- Indicar el uso de cola de mensajes
- Identificar procesos que requieran independencia en la ejecución paralela de manera de resaltar claramente aquellos procesos cuya ejecución no puede ser segmentada en distintos procesadores.
- Identificar web services.

Tomando los elementos propios del diagrama de actividades es posible hacer una adaptación la cual permitirá:

- Diferenciar nodos mediante las calles del Diagrama de Actividades.
- Indicar subcalles para distinguir hilos.
- Establecer tipo de conexión entre nodos.

Para la definición de este nuevo lenguaje (sub-lenguaje basado en UML el cual extiende al mismo añadiéndole expresividad para este dominio en particular) es necesario

generar un profile. Los pasos para la construcción del profile consisten en definir:

1) Un meta modelado del dominio de la aplicación: Si no existe, entonces es necesario definirlo utilizando los mecanismos del propio UML [1]. Para lo cual habrá que incluir la definición de las entidades propias del dominio, las relaciones entre ellas, así como las restricciones que limitan el uso de estas entidades y de sus relaciones.

2) El profile: Para lo cual un estereotipo debe ser creado por cada elemento del meta modelado. Es conveniente que los estereotipos tengan el mismo nombre que los elementos del meta-modelado.

Estableciéndose de esa forma una relación entre el meta modelado y el profile. En principio cualquier elemento que se necesite para definir el meta modelado puede ser etiquetado posteriormente con un estereotipo.

3) Los elementos UML que van a ser extendidos para cada estereotipo: Es importante tener en claro cuales son los elementos del meta modelado de UML que se están extendiendo sobre los que es posible aplicar un estereotipo. Ejemplos de estos son: las clases, sus asociaciones, sus atributos, las operaciones, las transiciones etc. De esta forma cada estereotipo se aplicará a la meta-clase de UML que se utilizó en el meta-modelo del dominio para definir un concepto o una relación.

4) Los valores etiquetados. En este paso los valores etiquetados deben ser agregados para cada atributo del meta modelado que es asociado al profile.

5) Las limitaciones del profile, a partir de las restricciones del dominio: Esto se realiza mediante OCL (Lenguaje de restricción de objetos) [6].

3. Resultados Obtenidos/Esperados

En este apartado se muestra el resultado del seguimiento de los pasos para la construcción del profile de PROCODI:

1) Se debe identificar los elementos que forman la extensión del lenguaje UML para poder determinar el meta-modelado del mismo. Los componentes introducidos por PROCODI son los siguientes:

- Semáforos y Monitores: Cuando se utilizan hilos es muy posible que existan recursos que se deben compartir y por lo tanto es necesario administrar su acceso, ya que solamente un hilo puede utilizarlo en un momento dado.
- Cardinalidad: Esta construcción se propone para el caso de que varios hilos realicen una operación idéntica.
- Timer: Extendiendo la cardinalidad se puede realizar una construcción que represente actividades que se realicen cada determinado tiempo.
- Mensajes y RPC: El procesamiento distribuido requiere que los procesos alojados en distintos host, se comuniquen de alguna manera para poder intercambiar información. Existen dos formas distintas de realizar dicha comunicación: mediante el envío de mensajes y mediante la utilización de RPC.
- Colas de Mensajes: Para aplicaciones con comunicación asíncrona muchas veces los mensajes recibidos desde distintos nodos y aplicaciones se van encolando para luego ser procesados cuando el recurso esté disponible. Ejemplos de implementaciones de esta tecnología son MQSeries de IBM [4] y Microsoft Message Queuing (MSMQ) [5].
- Web Services: Actualmente tanto en entornos GRID como en entornos empresariales para lograr la reutilización de funcionalidades entre aplicaciones es posible exponer una funcionalidad determinada por medio de web services, lo que permite a otras aplicaciones consumirlos mediante una simple llamada http.

Los siguientes elementos surgen de hacer adaptaciones a los elementos preexistentes del diagrama de Actividades.

- Nodos: Cada calle del diagrama de actividades será un nodo de aplicación.
- Hilos: Se propone una notación mas clara, cuando se mencionan estados dispares de cada hilo a través de líneas punteadas.

Por otra parte es muy común en las aplicaciones paralelas y procesamiento concurrente la utilización de memoria

compartida. Seguidamente se detallan dos tipos específicos de las mismas.

- Memoria compartida: una memoria única y global accesible desde todos los procesadores.
- Memoria compartida – distribuida: la memoria está físicamente distribuida pero lógicamente compartida.

2) Una vez identificados los elementos para el metamodelado se realiza la definición de los estereotipos para cada elemento que se está extendiendo.

Es importante tener en cuenta que elementos del metamodelado de UML se están extendiendo y sobre los que es posible aplicar un estereotipo. Ejemplos de tales son clases, asociaciones, relaciones, operaciones, atributos, etc. De esta forma el estereotipo se aplicara a una metaclassa de UML. En la tabla 1 se puede observar la asociación entre los elementos de la extensión y las metaclassas en la cual se definen los estereotipos para los

mismos. A su vez los estereotipos definidos generan nuevas metaclassas que también pueden ser extendidas, por ejemplo en la tabla 1 se muestra una Cola de Mensajes utilizando como metaclassa ProMemoria que es un estereotipo propio del profile.

Tabla 1. Asociación entre los elementos de extensión y las metaclassas

Elemento de ProCodi	Estereotipo	UML Metaclass
Swimlanes	ProSwimlanes	Colaboración
Memoria	ProMemoria	Clase
Actividades	ProActividad	Operación
Semáforos	ProSemaforo	Clase
Monitor	ProMonitor	Clase
Timer	ProTimer	Parámetro de Actividad
Cardinalidad	ProCardinalidad	Operación
Cola de Mensajes	ProColaMsg	ProMemoria
Web Services	ProWebServices	Operación
SubCalles	ProSubCalles	Colaboración

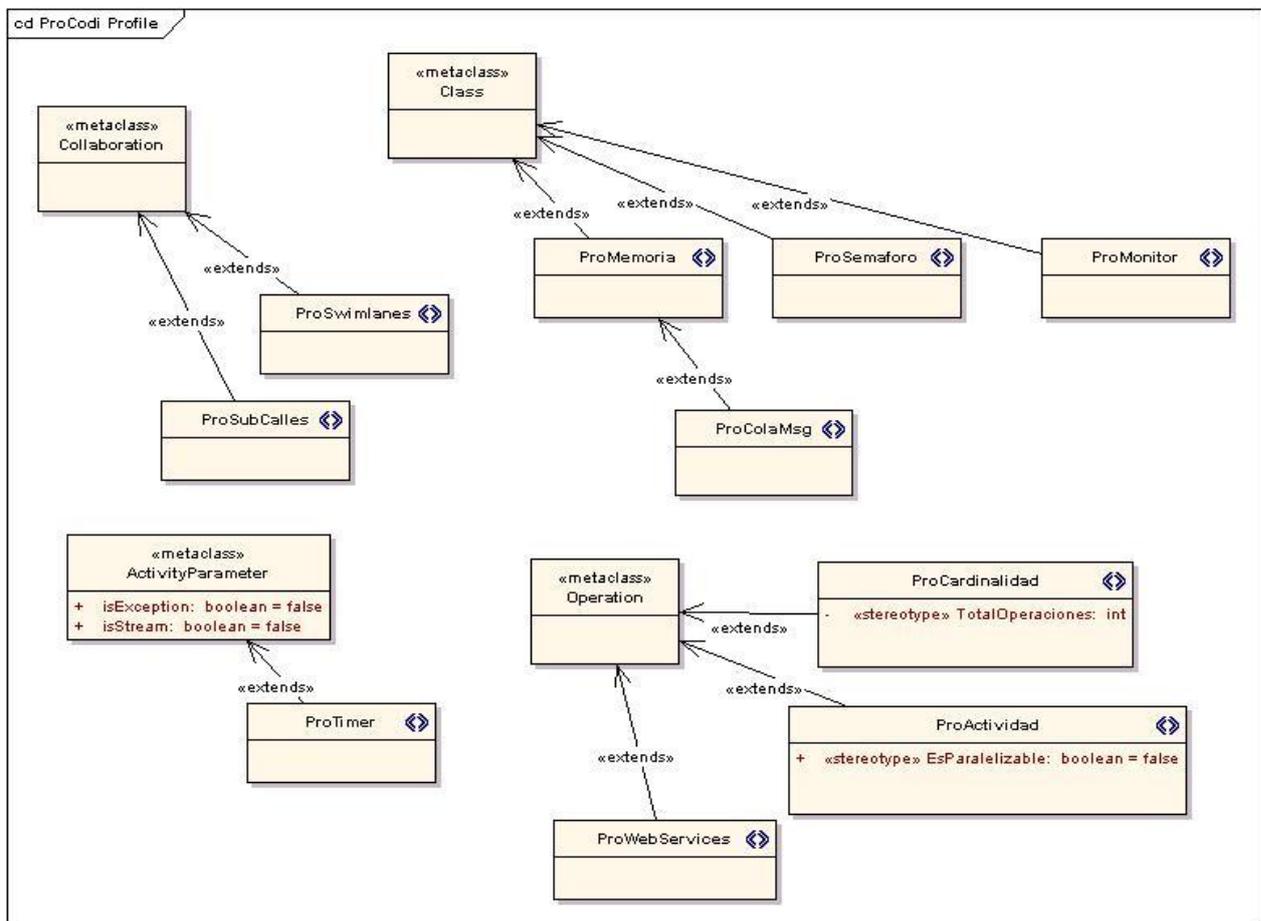


Figura 1. Determinación de los valores etiquetados para cada elemento del Profile

- 3) Se procede a la determinación de los valores etiquetados de los elementos del profile, estos valores etiquetados son los atributos de los elementos que estamos extendiendo. Deben incluir la definición de sus tipos, y sus posibles valores iniciales.

En la figura 1 se pueden observar los estereotipos derivados de las metaclasses: Class, Collaboration, Operation, ActivityParameter, así como algunos valores etiquetados en Cardinalidad para representar la cantidad total de Operaciones.

Como resultado final se obtiene el paquete del Profile mostrado en la figura 2. Dentro del paquete se incluyen todos los estereotipos usados y los creados por el profile. Este paquete es creado automáticamente por la herramienta utilizada.

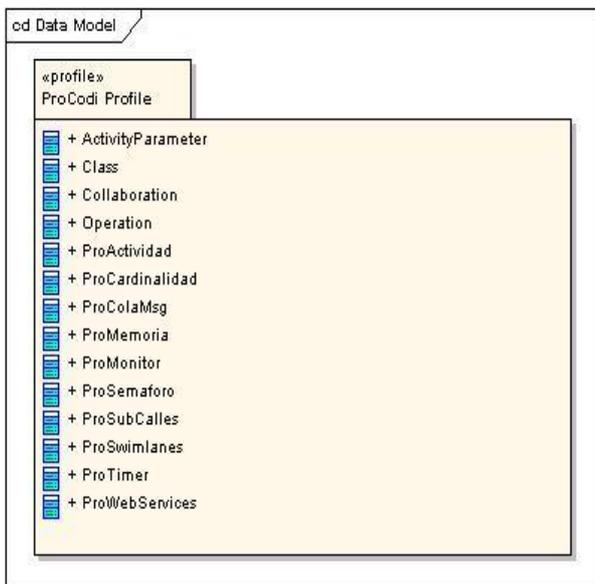


Figura 2. Paquete del profile

Actualmente el grupo de investigación se encuentra en desarrollo de las restricciones del profile, estas restricciones son escritas en OCL. Mediante OCL se podrán escribir las restricciones de las extensiones a través de un lenguaje formal que facilita la eliminación de ambigüedades.

4. Formación de recursos humanos

El equipo de investigación está formado actualmente por 6 personas, entre los cuales se encuentran docentes, egresados y colaboradores externos vinculados con

empresas privadas. Tres de los miembros del equipo se encuentran haciendo un doctorado en ciencias informáticas, el tema de tesis de uno de ellos está asociado con la presente investigación. El equipo trabaja en la definición de restricciones mediante OCL y prevé para lograr finalizar el profile incluir 3 personas más al equipo de trabajo (alumnos avanzados o becarios).

Esta línea de investigación comienza en el 2005 y ha ido evolucionando permitiendo obtener varios resultados intermedios los cuales han sido presentados en diversos congresos nacionales e internacionales. A fines del 2008 un paper presentado por el equipo de trabajo ha sido elegido en la Jornada Chilena de Computación entre los 4 mejores papers publicados en el “Workshop de Sistemas Distribuidos y Paralelismo”, obteniendo la invitación de escribir un artículo extendido para su publicación en una revista digital.

5. Bibliografía

[1] Booch G, Rumbaugh J y Jacobson I. El proceso unificado de desarrollo de software. Addison Wesley, 2001.

[2] Catalog of UML Profile of Specification http://www.omg.org/technology/documents/profile_catalog.htm

[3] Fuentes L. y Vallecillo A. Una Introducción a los Perfiles UML. Depto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga Campus de Teatinos. España

[4] IBM, MQSeries. Disponible en: <http://www-306.ibm.com/software/integration/wmq/>

[5] Microsoft, Message Queuing. Disponible en: <http://www.microsoft.com/windowsserver2003/technologies/msmq/default.mspx>

[6] Warmer J., Kleppe A. The Object Constraint Language: Precise Modeling With UML.