

# SODM+T: Inferencia de restricciones de rendimiento

A. García Domínguez      I. Medina Bulo      M. Marcos Bárcena  
Escuela Superior de Ingeniería    Escuela Superior de Ingeniería    Escuela Superior de Ingeniería  
Universidad de Cádiz            Universidad de Cádiz            Universidad de Cádiz  
antonio.garciadominguez@uca.es    inmaculada.medina@uca.es      mariano.marcos@uca.es

## Resumen

Las metodologías dirigidas por modelos (MDM) han experimentado un gran impulso en los últimos años, así como su aplicación a las Arquitecturas Orientadas a Servicios (AOS). Sin embargo, su avance no ha ido paralelo a la integración en ellas de técnicas de verificación y validación. En este trabajo se presenta SODM+T, el conjunto de herramientas que apoya a la metodología SODM+Testing que extiende la MDM de desarrollo de AOS SODM, integrando técnicas y modelos para la realización de pruebas. Se muestra uno de sus componentes, dedicado a asistir al usuario a diseñar pruebas de rendimiento sobre composiciones de servicios en una fase temprana de desarrollo, infringiendo las restricciones a partir de la información parcial disponible.

## 1. Introducción

Las Arquitecturas Orientadas a Servicios (AOS) prometen aumentar la flexibilidad de los sistemas de información de las organizaciones y reducir los costes de desarrollo, aprovechando en distintas partes de la organización un catálogo bien organizado de servicios. Cuando sea necesaria una nueva funcionalidad, se puede reunir una selección de estos servicios en una *composición* de manera flexible. Sin embargo, las AOS no están exentas de complicaciones: son complejas de desarrollar, al afectar potencialmente a toda la organización, y la reutilización de servicios crea mayor dependencia en su buen funcionamiento.

Ante el primer problema, han surgido diver-

sas metodologías para simplificar el desarrollo de las AOS con distintos alcances y enfoques, desde las pautas generales de Erl [1] hasta la extensiva SOMA de IBM [2]. De entre ellas, la metodología SODM alcanza un equilibrio entre alcance y coste de aplicación [3]. SODM es una metodología dirigida por modelos según el enfoque MDA [4] del OMG, en la que el sistema se va desarrollando a través del refinamiento sucesivo de una serie de modelos.

Sin embargo, estas metodologías no resuelven el segundo problema, al no integrar explícitamente técnicas y modelos para pruebas que ayuden a obtener servicios fiables. Por este motivo, los autores de este trabajo propusieron en [5] la integración de pruebas en SODM. A raíz de este trabajo, se comenzó a desarrollar la metodología resultante SODM+Testing, y su herramienta de apoyo SODM+T, ofreciéndose los primeros resultados en [6].

En este trabajo se presenta el componente de SODM+T dedicado a asistir a los usuarios a diseñar pruebas de rendimiento de los servicios de sus composiciones. Permite inferir de manera automática las restricciones de rendimiento que faltan de un modelo de composición de servicio y comprobar la consistencia de todas las restricciones. De esta forma, ayuda a paliar la falta de información que se da en una fase temprana del desarrollo de la AOS, y asiste al usuario en diversos escenarios usuales al desarrollar composiciones de servicios.

## 2. Algoritmos de inferencia

Los algoritmos de inferencia implementados en SODM+T permiten completar las anota-

ciones que faltan en los modelos de proceso de servicio y composición de servicio de SODM+Testing. Estos modelos están basados en los de SODM, pero han sido replanteados para añadir anotaciones de rendimiento y usar un núcleo común basado en una simplificación de los diagramas de actividad UML.

En la gráfica 1 se muestra un ejemplo de un modelo de proceso de servicio de SODM+Testing. El modelo incluye:

1. Una anotación manual de rendimiento para todo el proceso de servicio, con el estereotipo `<<sppc>>`. En el ejemplo se indica que todo el proceso de servicio ha de poder atender 5 peticiones concurrentes (atributo *concurrentRequests*) en un tiempo máximo de 1 segundo (atributo *timeLimit*).
2. Un grafo inspirado en los diagramas de actividad UML, donde cada arco saliente de un nodo de decisión incluye la probabilidad de que se cumpla su condición.
3. Cero o más anotaciones manuales (con el estereotipo `<<apc>>`) sobre las actividades del modelo de proceso de servicio o las acciones del modelo de composición de servicio. «Evaluar Pedido» debe atender 5 peticiones concurrentes en 0,4 segundos como mucho.

Los modelos de composición de servicio de SODM+Testing subdividen cada actividad del modelo de proceso de servicio en un subgrafo de acciones, permitiendo que el usuario infiera restricciones de rendimiento en su interior.

Las anotaciones locales con el atributo *manual* igual a `false` de la figura 1 han sido inferidas. Si el usuario cambia la estructura del grafo o las restricciones manuales, sólo ha de ejecutar los algoritmos de nuevo para actualizarlas.

Dado que los algoritmos no sólo inferen, sino que además comprueban la consistencia de tanto las restricciones manuales como las automáticas, se pueden utilizar para:

1. Diseñar los casos de prueba de rendimiento de las distintas partes del sistema.

2. Establecer valores iniciales para los acuerdos de nivel de servicio, en una fase temprana del desarrollo del sistema en la que no se dispone de mediciones exactas.

3. Comprobar si una determinada selección de servicios externos, con acuerdos de nivel de servicio predefinidos, consigue que se cumpla la restricción de rendimiento de todo el modelo de proceso de servicio o una actividad del modelo de composición de servicio.

### 3. SODM+T

SODM+T se halla implementado como un conjunto de extensiones para el entorno de desarrollo integrado Eclipse. El código se halla disponible como software libre bajo la licencia Eclipse Public License 1.0 a través de su web [7]. Puede verse una toma de pantalla en la figura 2.

Los editores gráficos de SODM+T están implementados en Java. El código es generado y personalizado automáticamente siguiendo una metodología dirigida por modelos y utilizando los marcos de desarrollo Eclipse Modeling Framework y Graphical Modeling Framework. El proceso se dirige desde un guión Apache Ant, sin requerir intervención manual del usuario. Las personalizaciones de los modelos intermedios y del código se definen explícitamente por separado, asegurando que el código pueda regenerarse desde cero en cualquier momento.

SODM+T valida automáticamente los modelos al guardar y asiste en la corrección de los problemas detectados usando un guión escrito en el Epsilon Validation Language (EVL). Al mismo tiempo, varias restricciones OCL impiden que el usuario realice ciertas acciones que producirían modelos inconsistentes.

El usuario dispone de accesos directos en los menús de Eclipse para transformar automáticamente los modelos de proceso de servicio a modelos de composición de servicio, conservando las restricciones de rendimiento. La transformación se implementa mediante el Epsilon Transformation Language (ETL).

Finalmente, el usuario puede lanzar desde el editor gráfico los algoritmos de inferencia, que

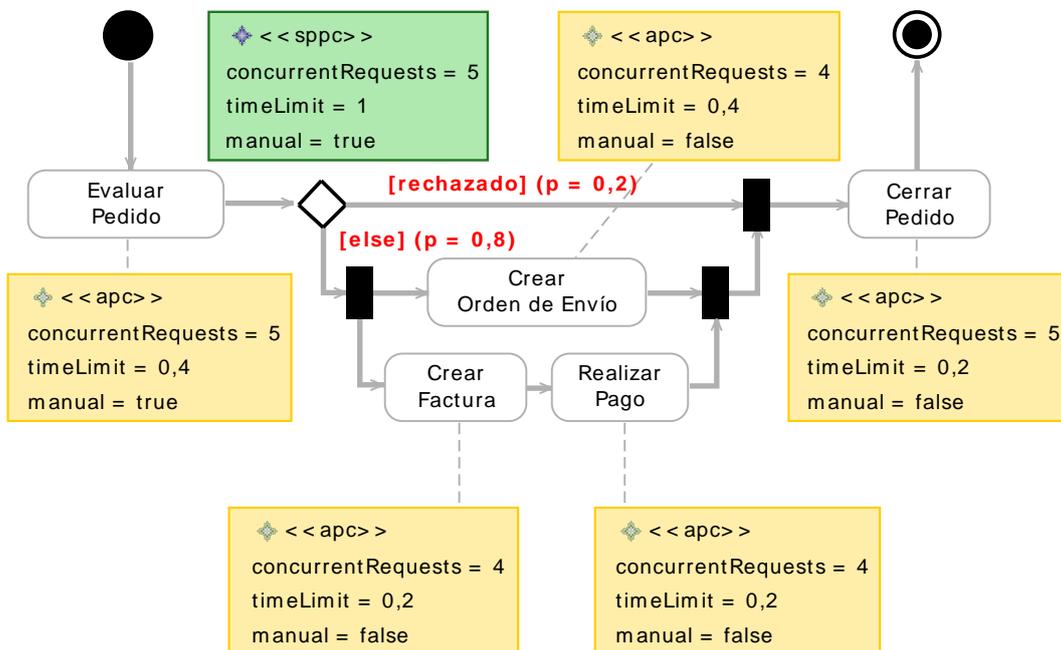


Figura 1: Ejemplo de modelo de proceso de servicio de SODM+Testing

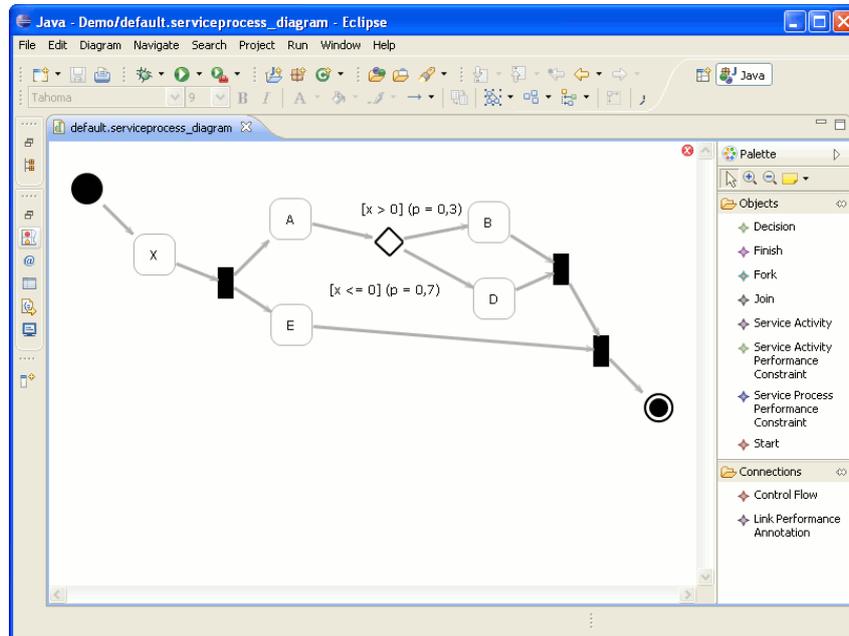


Figura 2: Toma de pantalla de SODM+T

actualizarán automáticamente el modelo con los resultados. Los algoritmos se encuentran implementados en el Epsilon Object Language (EOL), y se integran en la interfaz gráfica de Eclipse mediante un guión en el Epsilon Wizard Language (EWL).

Como puede verse, las distintas tareas de manejo de modelos en SODM+T se encuentran implementadas en lenguajes de la familia Epsilon [8]. Esta familia parte del lenguaje EOL (una versión imperativa de OCL) y lo extiende en una serie de lenguajes específicos a cada tarea, como EVL, ETL y EWL. Esto permite reutilizar el mismo código EOL entre validaciones, transformaciones y asistentes.

#### 4. Conclusiones y Trabajo Futuro

Las AOS prometen mejorar la flexibilidad y reducir los costes en las organizaciones. Sin embargo, es difícil desarrollarlas de forma que se obtengan todos estos beneficios, por lo que han surgido una serie de metodologías para dirigir su creación. SODM+Testing es una de ellas: extiende a la metodología SODM con modelos, técnicas y tareas de pruebas.

En este trabajo se ha presentado SODM+T, el conjunto de herramientas que apoya a la metodología SODM+Testing. SODM+T permite inferir restricciones de rendimiento a partir de la información parcial disponible en una fase temprana del desarrollo del sistema. SODM+T puede usarse para establecer valores iniciales para los niveles de servicio exigidos a los servicios externos, diseñar casos de prueba de rendimiento para los servicios internos y validar la selección de una combinación determinada de servicios.

En un futuro, se planea pasar las anotaciones de rendimiento a un perfil estándar, como el OMG SPTP [9], y adaptar los algoritmos a un metamodelo que permita generar código de una forma más directa, como BPMN [10].

#### Referencias

[1] T. Erl. *SOA: Principles of Service Design*. Prentice Hall, Indiana, EEUU, 2008.

- ISBN 0132344823.
- [2] S. Ghosh, A. Arsanjani y A. Allam. SOMA: a method for developing service-oriented solutions. *IBM Systems Journal*, **47**(3):377–396, 2008.
- [3] M. de Castro. *Aproximación MDA para el desarrollo orientado a servicios de sistemas de información web: del modelo de negocio al modelo de composición de servicios web*. Tesis Doctoral, Universidad Rey Juan Carlos, marzo 2007.
- [4] Object Management Group. MDA Guide version 1.0.1. <http://www.omg.org/mda/>, junio 2003.
- [5] A. García Domínguez, I. Medina Bulo y M. Marcos Bárcena. Hacia la integración de técnicas de pruebas en metodologías dirigidas por modelos para SOA. En *Actas de las V Jornadas Científico-Técnicas en Servicios Web y SOA*. Madrid, España, octubre 2009.
- [6] A. García Domínguez, I. Medina Bulo y M. Marcos Bárcena. Primeros avances en una metodología orientada a modelos para los sistemas de información de empresas de fabricación distribuida. En *Actas de las I Jornadas Predoctorales de la ESI*, páginas 109–112. Cádiz, España, diciembre 2009. ISBN 978-84-608-0981-4.
- [7] A. García Domínguez. Página web de SODM+T. <https://neptuno.uca.es/redmine/projects/sodmt>, marzo 2010.
- [8] D. Kolovos, R. Paige, L. Rose y F. Polack. *The Epsilon Book*. <http://www.eclipse.org/gmt/epsilon>, 2010.
- [9] Object Management Group. UML Profile for Schedulability, Performance, and Time (SPTP) 1.1. <http://www.omg.org/spec/SPTP/1.1/>, enero 2002.
- [10] Object Management Group. Business Process Modeling Notation (BPMN) 1.2. <http://www.omg.org/spec/BPMN/1.2/>, enero 2009.