
Primeros Avances en una Metodología Orientada a Modelos para los Sistemas de Información de Empresas de Fabricación Distribuida

A. García Domínguez ⁽¹⁾, I. Medina Buló ⁽¹⁾ y M. Marcos Bárcena ⁽²⁾

⁽¹⁾Departamento de Lenguajes y Sistemas Informáticos, Universidad de Cádiz, Escuela Superior de Ingeniería. C/Chile 1, 11002, Cádiz.

⁽²⁾Departamento de Ingeniería Mecánica y Diseño Industrial, Universidad de Cádiz, Escuela Superior de Ingeniería. C/Chile 1, 11002, Cádiz.

Resumen

Para mantenerse competitivas, las empresas de fabricación necesitan aumentar su flexibilidad y reducir costes, para lo que han de integrar a entes externos dentro de su tejido. El uso de una arquitectura orientada a servicios para estructurar sus sistemas de información puede facilitar esta tarea. Han surgido diversas metodologías dedicadas a estas arquitecturas, pero ninguna cubre sus pruebas. Este artículo presenta los primeros avances en la definición de una metodología que cubra estas carencias, y se ofrecen algunos resultados preliminares. La metodología refina una serie de modelos, cada vez más cercanos al código que implementará el sistema.

1. Introducción

Hoy en día, las empresas se hallan bajo la necesidad de competir en un mercado en el que los ciclos de vida de los productos son cada vez más cortos y se exigen mayores niveles de flexibilidad y calidad a menor coste. Para ello, necesitan ser capaces de reorientar y mejorar continuamente sus procesos de negocio en función de la situación y de forma rentable. Sin embargo, las plataformas centralizadas normalmente usadas en las empresas de hoy en día no pueden cambiarse tan rápidamente como las situaciones lo requerirían, y finalmente son éstas las que definen las prácticas a seguir, más que la propia situación del mercado. Se necesita, por lo tanto, un enfoque distinto para estructurar los sistemas de información en la así llamada Siguiete Generación de Sistemas de Fabricación (SGSF).

Actualmente, se admite a nivel conceptual la necesidad de distribuir las actividades a lo largo de varios sistemas de información especializados y de posteriormente integrarlas en lo que se conoce como una empresa extendida [1]. Entre los protomodelos de empresa distribuida más conocidos, se encuentran las organizaciones holónicas, llamadas así por constituirse de una serie de actores semiautónomos que se interrelacionan entre sí a varios niveles, conocidos como holones [2].

Para plasmar este marco conceptual, afortunadamente, se ha popularizado en los últimos años una forma de estructurar los sistemas de información que se ajusta bien al

enfoque holónico. Se trata de las Arquitecturas Orientadas a Servicios o “*Service-Oriented Architectures*” (SOA). La idea central detrás de ellas es organizar los sistemas de información no como sistemas integrados unidad a unidad de negocio o proyecto a proyecto, como se ha hecho comúnmente, sino como servicios individuales que pueden ser reutilizados a lo largo de la organización, o incluso por otras organizaciones con las que se tiene relación.

Posteriormente, estos servicios individuales pueden ser integrados en servicios de nivel superior con su propio valor añadido, modelando procesos de negocio completos en vez de operaciones individuales. La información deja de concentrarse en “silos” de departamentos concretos y pasa a estar disponible a toda la organización.

Este enfoque ofrece grandes ventajas, pero no deja de tener sus propias dificultades. Implantar una SOA es complejo, ya que afecta a toda la organización, y necesita tanto una buena visión global como una correcta implementación.

Otro problema es el hecho de que el mayor grado de integración y la mayor visibilidad de los servicios de la organización implican también una mayor dependencia en su correcto funcionamiento, y un mayor grado de impacto en caso de fallo. Esto se acentúa en el caso de SOA, ya que integrar servicios externos a la organización es central a dicho enfoque.

2. Propuesta

Bajo esta situación, en este trabajo se propone la definición de una metodología para la

implantación de arquitecturas orientadas a servicios para los sistemas de información de empresas de fabricación distribuida. Esta metodología debería tener las siguientes características:

- Poder responder a las verdaderas necesidades de la organización, por lo que tendrá que apoyarse en una descripción de la organización y su entorno actual de negocio.
- Ser lo suficientemente flexible como para ser utilizada en organizaciones de menor tamaño, con menos recursos para realizar un modelado extensivo.
- Poder modelar las pruebas sobre la funcionalidad y otras propiedades deseadas del sistema, más que únicamente su funcionalidad.
- Ser automatizable, incluyendo herramientas que asistan al usuario a comprobar la consistencia de sus descripciones, conseguir el código del sistema y realizar pruebas sobre él.
- Apoyarse sobre tecnologías fácilmente disponibles y extensibles.

Esta metodología se validará sobre casos prácticos basados en empresas de diversos sectores, como el alimentario o el aeronáutico.

3. Estado Actual de la Metodología

Durante el Trabajo Fin del Máster en Ingeniería de Fabricación se ha realizado un estudio de viabilidad de los objetivos anteriores, apoyándose en metodologías existentes y en un caso práctico inspirado en una importante empresa del sector tabaquero. En esta sección se

describen los progresos realizados en la definición de dicha metodología, dejándose los aspectos técnicos para una posterior sección.

Desde el comienzo, se decidió basarse en una metodología existente y extenderla, en vez de partir de cero. En particular, se compararon tres metodologías:

- Stojanović [3], que extiende su metodología de desarrollo basado en componentes para tratar aspectos de SOA.
- La metodología SOMA [4] de IBM, que añade los aspectos necesarios al *Rational Unified Process* (RUP) para tratar aspectos de SOA.
- La metodología SOD-M [5] del grupo Kybele de la Universidad Rey Juan Carlos, creada desde cero alrededor del concepto de servicio, y que integra el enfoque de arquitectura dirigida por modelos propuesto por el Object Management Group [6].

De entre las tres metodologías se ha escogido SOD-M, por su potencial de automatización frente a la de Stojanović y por su menor coste de aplicación frente a SOMA. Puede verse un esquema del proceso en la figura 1. Se trata de una metodología dirigida por modelos, que consiste en la definición de una serie de modelos cada vez más detallados, primero del negocio (nivel de los modelos independientes de computación o CIM), luego del sistema a nivel abstracto (modelos independientes de plataforma o PIM) y finalmente del sistema concreto, considerando las tecnologías usadas (modelos específicos de plataforma o PSM).

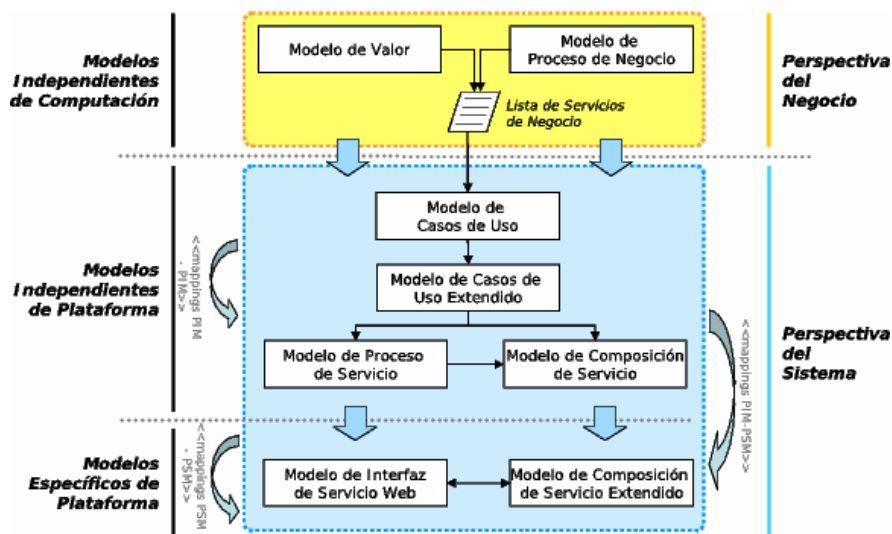


Figura 1. Esquema de la metodología SOD-M

Durante el análisis de SOD-M se identificó una importante carencia en dicha metodología: no asistía en las pruebas sobre la SOA finalmente implementada. Sin embargo, para que una SOA sea efectiva, debe minimizarse el riesgo que puede suponer la integración de un servicio de otra parte de la organización o incluso de una organización externa, y esto incluye la definición y ejecución metódica de un plan de pruebas sobre el sistema. Por esta razón, se han propuesto [7,8] varias extensiones sobre SOD-M para cubrir dicha carencia:

- Modelar cada una de las actividades del sistema con el tiempo límite de respuesta ante una determinada carga.
- Definir la funcionalidad esperada de cada parte del sistema de forma abstracta, usando condiciones a cumplir antes y después de su ejecución, como JML [9] hace con Java o WS-CoL [10] con WS-BPEL.

Estas descripciones pueden integrarse con una serie de técnicas de prueba existentes, como los grafos causa-efecto de Paradkar [11] o la partición del espacio de entrada sugerida por Lohmann [12].

4. Estado Actual de las Herramientas

Las herramientas de la metodología SOD-M, tal y como se recibieron en marzo del 2009 del grupo Kybele, se hallaban desarrolladas como extensiones de la plataforma de desarrollo Eclipse, impulsada por IBM. Su flexibilidad se debe a que todo lo hace con componentes externos integrados por un núcleo mínimo.

En particular, empleaban los populares marcos de trabajo Eclipse Modeling Framework (EMF) y Graphical Modeling Framework (GMF). EMF permite implementar los modelos dando una especificación abstracta de su estructura. GMF permite conseguir editores gráficos más cómodos para el usuario a partir de una serie de especificaciones más detalladas y de retoques manuales sobre el código generado.

Sin embargo, EMF y GMF por sí solos imponen un flujo de trabajo que carece de la robustez y agilidad necesaria para permitir la experimentación sobre distintos enfoques que requiere la implementación de las propuestas realizadas en el anterior apartado. Se han integrado una serie de técnicas dirigidas a resolver estos problemas:

- Las especificaciones de los modelos para EMF ahora se hacen en formato texto, mediante el editor Emfatic, potenciando su manipulación automática.

- Se utilizan los lenguajes definidos por el proyecto Epsilon para manipular automáticamente los modelos, validándose con EVL y transformándose con ETL y EWL.
- Se emplea EuGENia para producir las especificaciones para GMF en combinación con programas escritos en el lenguaje EOL de Epsilon, evitando depender de una versión concreta y automatizando los retoques manuales.
- El proceso de obtención de los editores gráficos a partir de los modelos iniciales se ha automatizado con Apache Ant, tras añadir los aspectos que aún no eran automatizables.

Utilizando este flujo de trabajo mejorado, se reemplazaron dos de las herramientas recibidas de Kybele con nuevas versiones desarrolladas desde cero que incorporaban las extensiones propuestas para modelar el rendimiento del sistema, junto con:

- Validación automática de los modelos, ofreciendo la posibilidad al usuario de corregir automáticamente sus errores en la mayoría de los casos.
- Completado de las restricciones de rendimiento no especificadas manualmente por el usuario, evitando que las restricciones manuales se contradijeran entre sí.
- Transformación automática entre los modelos extendidos, para que el usuario pueda rápidamente empezar a definir los modelos más detallados.

5. Validación

Para validar la metodología y la implementación de las herramientas, se aplicaron sobre un caso práctico inspirado en una parcela del negocio de una importante empresa del sector tabaquero.

Tras recopilar información sobre la organización y sus prácticas, se elaboró un modelo de los intercambios de valor producidos en la organización (propuestos por Gordijn [13]) y se definió un diagrama de actividad UML (parecido a un diagrama de flujo) con el proceso de negocio de la organización.

A partir de estos dos modelos, se estableció una lista de los servicios a ofrecer por el sistema, y tras varios pasos se produjo un modelo que descomponía las 46 tareas concretas a implementar en el sistema, y seleccionaba las que se ofrecerían como servicios reutilizables en el resto de la organización.

Cada una de estas 46 tareas incluía sus propias anotaciones sobre el rendimiento esperado a una

determinada carga. Sólo hubo que especificar el rendimiento esperado para los servicios: la estructura de los diagramas de actividad y las estimaciones de probabilidades de sus arcos proporcionaron la información necesaria a los algoritmos de completado de estimaciones.

La aplicación de esta primera versión parcial de la metodología fue positiva, permitiendo especificar de forma detallada un sistema partiendo únicamente de una descripción inicial de sus prácticas actuales de negocio y los participantes involucrados. Sin embargo, la metodología aún no se halla completa: no se genera código, y no se llegan a ejecutar las pruebas definidas. Estas características se irán añadiendo gradualmente en el futuro, y se describen en más detalle en el apartado posterior.

6. Conclusiones

Las empresas de fabricación pueden aprovechar las arquitecturas orientadas a servicios para integrar nuevas técnicas y prácticas de negocio más rápidamente y a menor coste. Sin embargo, es necesario tener una metodología que defina el proceso a seguir, y que dirija las pruebas sobre dicho sistema. En este trabajo se ha propuesto una forma de llegar a dicha metodología, y se han ofrecido los primeros resultados.

Se ha seleccionado SOD-M, la metodología SOA propuesta por el grupo Kybele de la Universidad Rey Juan Carlos, y se han propuesto una serie de extensiones para que pueda modelar las pruebas del sistema. Parte de estas extensiones se han implementado dentro de la plataforma Eclipse de IBM, y se han validado con éxito dentro de un caso de negocio dedicado a una empresa del sector tabaquero.

En un futuro, se planea revisar la última versión de la metodología SOD-M, que ha cambiado en paralelo con el desarrollo de este trabajo, e integrar los nuevos aspectos que se consideren de interés. La herramienta se hará más cómoda y completa, permitiendo generar al menos parcialmente el código del sistema y sus pruebas, ejecutar las pruebas y analizar sus resultados sobre el modelo.

6. Referencias

- [1] M. Marcos, F. Aguayo, M. Sánchez Carrilero, L. Sevilla, y J.R. Lama, *Toward the Next Generation of Manufacturing Systems. Frabiho: a Synthesis Model for Distributed Manufacturing*, Proceedings of the First I*proms Virtual Conference, Elsevier (2005), págs. 35-40.
- [2] A. Tharumarajah, A. Wells, y L. Nemes, *Comparison of emerging manufacturing concepts*, Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, CA, USA (1998), págs. 325-331.
- [3] Z. Stojanović, *A Method for Component-Based and Service-Oriented Software Systems Engineering*, Delft University of Technology (2005).
- [4] S.G. A. Arsanjani y A. Allam, *SOMA: a method for developing service-oriented solutions*, IBM Systems Journal 47 (2008), págs. 377-396.
- [5] M.V. 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*, Universidad Rey Juan Carlos (2007).
- [6] Object Management Group, *MDA Guide version 1.0.1* (junio 2003), véase: <http://www.omg.org/mda/>.
- [7] A. García Domínguez, I. Medina Bulo, y M. Marcos Bárcena, *Hacia la Integración de Técnicas de Pruebas en Metodologías Dirigidas por Modelos para SOA*, Actas de las V Jornadas Científico-Técnicas en Servicios Web y SOA, Madrid (2009).
- [8] I. Medina Bulo, A. García Domínguez, F. Aguayo, L. Sevilla, y M. Marcos Bárcena, *Propuesta metodológica para la implementación de una arquitectura orientada a servicios en entornos de Sistemas de Fabricación Distribuida*, Actas del III Congreso Internacional de la Sociedad de Ingeniería de Fabricación, Alcoy, España (2009), págs. 346-353.
- [9] L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, y P. Spoletini, *A Timed Extension of WSCoL*, Web Services, 2007. ICWS 2007. IEEE International Conference on, (2007), págs. 663-670.
- [10] L. Burdy, Y. Cheon, y D.R. Cok, *An overview of JML tools and applications*, International Journal on Software Tools for Technology Transfer (STTT) 7 (junio 2005), págs. 212-232.
- [11] A. Paradkar, M.A. Vouk, y K.C. Tai, *Specification-based testing using cause-effect graphs*, Annals of Software Engineering 4 (enero 1997), págs. 133-157.
- [12] M. Lohmann, L. Mariani, y R. Heckel, *A Model-Driven Approach to Discovery, Testing and Monitoring of Web Services*, Test and Analysis of Web Services, Springer Berlin (2007), págs. 173-204.
- [13] J. Gordijn y H. Akkermans, *Value-based requirements engineering: exploring innovative e-commerce ideas*, Requirements Engineering 8 (julio 2003), págs. 114-134.