

PERSEO: Identificando servicios en sistemas heredados mediante un enfoque ADM

Ignacio García-Rodríguez de Guzmán¹, Ricardo Pérez-Castillo², Mario Piattini¹

Institute of Information Technologies and Systems¹
University of Castilla-La Mancha
Camino de Moledores s/n, 13071, Ciudad Real
{Ignacio.GRodriguez, Mario.Piattini}@uclm.es
Itestra GmbH²
Capitán Haya 1, Planta 15, 28020 Madrid
perez@itestra.com
+49 177 277 4558

Resumen. En la actualidad, las empresas se encuentran con el hecho de que sus sistemas de información empiezan a encontrarse obsoletos y sin apenas capacidad de maniobra para afrontar los cambios tanto tecnológicos como negocio que pueden surgir (y surgirán). El principal problema de esta obsolescencia es la cantidad de conocimiento embebido en el portafolio de sistemas de las empresas. Esto hace que la opción de desechar los sistemas actuales y sustituirlos por otros nuevos sea una opción que no resulta viable. La arquitectura orientada a servicios, también conocida como SOA, puede verse como otra fase dentro de la evolución del software, y que permite dotar a la infraestructura software de las empresas de esa flexibilidad de que en estos momentos adolece. Por ello, es posible que ésta sea la opción más adecuada ante la disyuntiva que se presenta con los sistemas heredados, permitiendo que los mismos evolucionen hacia este paradigma tecnológico. Este trabajo presenta una versión inicial de un entorno para la generación de especificaciones en SoaML a partir de sistemas heredados, facilitando así la migración de estos sistemas hacia el paradigma SOA.

Palabras clave. Architecture-Driven Modernization, service elicitation, legacy system, service migration, SoaML, KDM.

1 Introducción

La arquitectura orientada a servicios, también conocida como SOA, puede verse como otra fase dentro de la evolución del software. Sin embargo esta evolución supone un cambio muy significativo en cómo las empresas deben implementar y suministrar sus funcionalidades a los usuarios [1].

En la actualidad, las empresas se encuentran con el hecho de que sus sistemas de información se encuentran obsoletos y sin apenas capacidad de maniobra para afrontar los cambios tanto tecnológicos como de negocio que pueden surgir y surgirán. El principal problema de esta obsolescencia es la cantidad de conocimiento embebido

(procesos y reglas de negocio) en el portafolio de sistemas de las empresas. Esto hace que la opción de desechar los sistemas actuales y sustituirlos por otros nuevos sea una opción que no resulta viable.

SOA, como paradigma tecnológico, permite dotar a la infraestructura software de las empresas de esa flexibilidad de que en estos momentos adolece. Es por esto que se hace imprescindible hallar una solución de compromiso entre la adopción de los nuevos paradigmas tecnológicos y la conservación de los sistemas actuales, que embeben el conocimiento de negocio de las organizaciones.

Probablemente, la opción más adecuada ante esta disyuntiva es permitir que los sistemas existentes evolucionen desde sus estados actuales hacia los servicios, para lo cual, es necesario aplicar reingeniería de forma que se puedan obtener nuevos sistemas orientados a servicios y que conserven la funcionalidad, las reglas y los procesos de negocio actuales (esto último cobra aun más importancia en el contexto de los servicios).

Por este motivo, en este artículo los autores presentan PERSEO, una herramienta para la migración de sistemas heredados hacia servicios. PERSEO es una herramienta basada en estándares de OMG¹ como KDM [2], QVT [3] y SoaML [4]. PERSEO se encuentra en una versión beta, sin embargo, ya permite abordar la ingeniería inversa de sistemas para su representación mediante modelos PSM y PIM [5] y una identificación preliminar de servicios para su especificación mediante SoaML. Además, el desarrollo de PERSEO se ha estructurado de forma que se puedan admitir distintas estrategias de identificación de servicios a partir de sistemas heredados según la naturaleza del sistema heredado o los requisitos del proyecto de migración.

El resto del artículo se estructura de la siguiente manera: la sección 2 presenta una visión general del estado del arte sobre migración de servicios; la sección 3 describe el proceso de migración ilustrando los niveles de abstracción que se contemplan a la hora de migrar un sistema a servicios y las transformaciones involucradas; la sección 4 muestra la herramienta PERSEO; por último, en la sección 5 se da una visión general del artículo en forma de conclusiones y las tareas que restan acometer en la evolución inmediata del desarrollo de PERSEO.

2 Estado del arte

En esta sección se estudiarán, de forma resumida, algunas de las propuestas con más impacto en el ámbito de la migración de los LIS (*Legacy Information System*) a SOA

Uno de los primeros aspectos que deben tenerse en cuenta (no sólo en la migración a SOA, sino en cualquier proceso de reingeniería) es *qué se espera del sistema destino*. Esta fase puede entenderse como *el estudio de las características funcionales del sistema destino* [6-8]. En este sentido, el método SOAMIG [7] se centra en la importancia del diseño de los servicios como resultado de la ingeniería directa de los servicios identificados en un LIS durante la fase de ingeniería inversa. Por otro lado, el método SMART [8] realiza, en primer lugar, el diseño de los servicios del sistema

¹ Object Management Group - <http://www.omg.org/>

destino en base a las funcionalidades potenciales identificadas en el sistema origen, y por otro lado, realiza una verificación de dichos servicios con los *stakeholders*.

Casi tan importante como realizar la evolución de un LIS a SOA, es determinar la viabilidad de esta migración, o lo que es lo mismo, determinar si se hace o no. Las prácticas más conocidas para determinar la viabilidad de la evolución se resumen en la propuesta de análisis coste-beneficio para proyectos de reingeniería de [9]. Esta propuesta ha sido muy empleada dentro de las técnicas de evolución LIS a SOA [10-12]. El método SMART emplea la técnica *options analysis for re-engineering* (OAR) [13] para determinar la viabilidad de la migración. En [13], se presenta una herramienta de soporte para la toma de decisiones en la elección de la estrategia de modernización de LIS a SOA desarrollada a partir del método SMART y el *framework* de toma de decisiones [14].

Para resolver el problema de la identificación de servicios en el LIS existen dos grupos de técnicas importantes: (i) modelado de los requisitos de negocio (top-down), y (ii) código heredado hacia funcionalidades de negocio (bottom-up). En el primer enfoque, los procesos de negocio más importantes se diseñan a partir de la información identificada en la fase de comprensión del sistema heredado. Este proceso de negocio es dividido en actividades hasta que estas son directamente *mapeables* a funcionalidades identificables en el LIS. Por otro lado, cuando se parte del código heredado para identificar las funcionalidades de negocio, se emplea el código fuente como punto de inicio para el descubrimiento de la información de negocio [15-21].

Para la integración de un LIS en un contexto SOA pueden seguirse (principalmente) dos enfoques: integración de sistemas heredados o migración de sistemas heredados. En la integración de sistemas heredados, el código fuente de los sistemas no sufre grandes cambios (uso de técnicas como los *wrappers* y los *middlewares*). La integración de sistemas heredados se considera una estrategia rápida, con menos riesgos, más económica y sencilla, sin embargo el sistema sigue siendo tal y como es, sin ser modificado. En [22] [12] [23] pueden encontrarse algunas propuestas en este sentido.

Por otro lado, en el enfoque de migración de sistemas heredados (*una reestructuración y modificación interna del sistema heredado en un nuevo sistema*), el código fuente del LIS sí que es sometido a transformaciones. Esta técnica es considerada como la más costosa, aunque puede efectuarse paulatinamente y a lo largo del tiempo, obteniendo como resultado una nueva versión del sistema sin los problemas que suele arrastrar un LIS por efecto del envejecimiento[24]. Existen varias técnicas para la migración desde LIS a SOA, como pueden ser *program slicing* [10, 25-27], transformación de modelos [6, 28, 29].

3 Descripción del proceso de migración hacia servicios

PERSEO es una herramienta que permite identificar servicios a partir de sistemas heredados. PERSEO se centra en la fase de ingeniería inversa del modelo de reingeniería en herradura y se basa en KDM, el cual permite realizar representaciones conceptuales abstractas de las diferentes vistas de la arquitectura de los LIS. Para ello, PERSEO estructura la información que extrae del sistema heredado en cuatro niveles

de abstracción. El paso de un nivel a otro se realiza mediante transformaciones desarrolladas en QVT [3] y de forma programática. La Fig. 1 representa gráficamente la arquitectura de PERSEO. A continuación, se definen los niveles de abstracción que considera PERSEO:

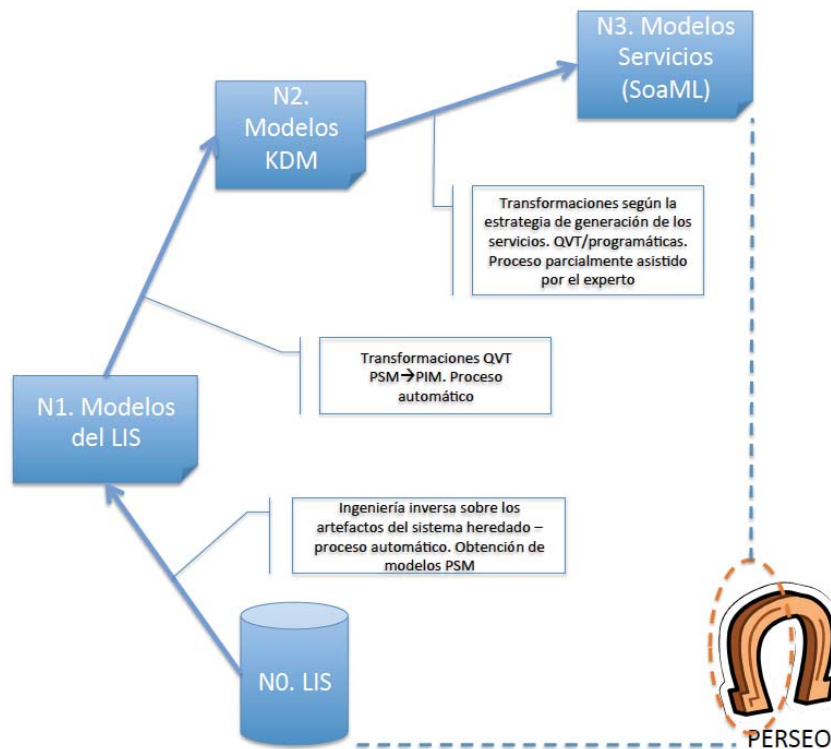


Fig. 1. Vista general de PERSEO

- Nivel 0 (N0). Este nivel representa al LIS de origen en el mundo real, del cual se pretende descubrir los servicios subyacentes.
- Nivel 1 (N1). Este nivel agrupa un conjunto de modelos que representan las diferentes vistas o aspectos de la arquitectura del LIS, es decir los diferentes activos software como código fuente, bases de datos, componentes, etc. La transformación N0-a-N1 obtiene modelos PSM desde cada activo software heredado. Para ello se recurre a la extracción de información del LIS por medio de técnicas de ingeniería inversa clásicas como análisis estático/dinámico, *slicing*, etc [30]. Los diferentes modelos PSM se construyen de acuerdo a metamodelos específicos según la tecnología del sistema origen o el artefacto del mismo que se esté procesando.
- Nivel 2 (N2). Este nivel se enfoca en la representación integrada de todos los activos software de un LIS de acuerdo al metamodelo de KDM. Durante la transformación N1-a-N2 los modelos específicos se transforman hacia un único modelo PIM basado en KDM. Estas transformaciones se formalizan mediante QVT. Dentro de este mismo nivel pueden establecerse transformaciones entre diferentes capas de la arquitectura KDM. Por ejemplo transformaciones entre la capa de ele-

mentos de programa o recursos que representan conocimiento explícito del LIS y la capa de abstracción que representa conocimiento implícito. De esta manera se reduce progresivamente la brecha conceptual entre el LIS y los servicios.

- Nivel 3 (N3). El último nivel representa los modelos de servicios que son inferidos a partir de los modelos KDM. El metamodelo que nos permite representar la información en este nivel es *SoaML* [4], un estándar de OMG para la especificación de sistemas basados en servicios. Cabe destacar, tal y como se ha mostrado en la sección 2, que existen distintas estrategias a la hora de migrar un sistema heredado hacia SOA. Por esta razón, el desarrollo de la transformación N2-a-N3 no se plantea como una transformación única. En la actualidad se está planteando el desarrollo de distintas transformaciones que permitan aplicar, según el contexto y las necesidades, una estrategia u otra para obtener un conjunto distinto de servicios.

La Tabla 1 resume cada una de las tres transformaciones de PERSEO. El objetivo final es completar el camino $N0 \rightarrow N1 \rightarrow N2 \rightarrow N3$ para generar los servicios necesarios que permitan exponer la funcionalidad más importante del sistema heredado.

Tabla 1. Transformaciones entre los niveles de PERSEO

Transformación	Artefactos	Metamodelos	Técnica de transformación
N0-a-N1	in Activos software del LIS: código fuente, bases de datos, interfaces de usuario, etc.	-	Técnicas de ingeniería inversa: análisis estático y dinámico, <i>program slicing</i> , etc.
	out Modelos PSM por cada activo software	Metamodelos específicos: MM_{JAVA} , MM_{SQL} , MM_{C++} , etc.	Transformaciones entre modelos establecidas mediante <i>QVT Relations</i> .
N1-a-N2	in Modelos KDM tipo PIM	Metamodelo de KDM	Transformaciones implementando estrategias de migración. Desarrolladas en QVT y de forma programática
	out Modelo de Servicios	Metamodelo de SoaML	Intervención del experto para ayudar en la configuración de los servicios.

4 PERSEO

PERSEO se presenta como un *plug-in* que se integra en la plataforma de desarrollos Eclipse (ver Fig. 2 (izda)). PERSEO, que se muestra como una opción de menú en la que se implementan las diferentes etapas del proceso.

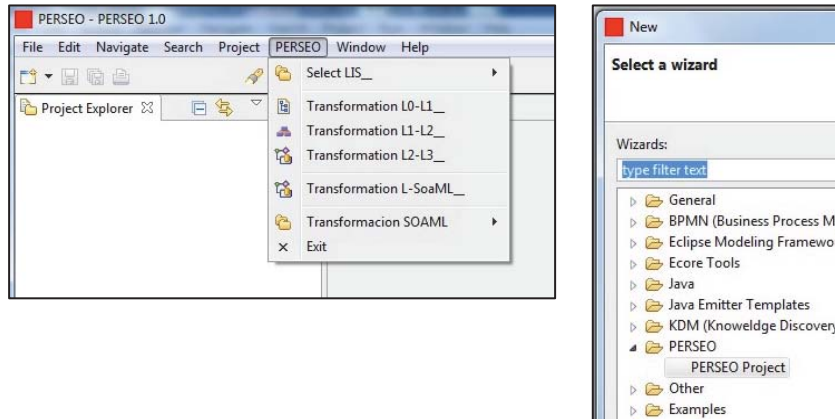


Fig. 2. Menú principal del *plug-in* PERSEO (izda) y creación de un nuevo proyecto PERSEO (dcha)

Para comenzar la identificación de servicios en un sistema heredado, se creará un nuevo proyecto del tipo “PERSEO Project” (ver Fig. 2 (dcha)), con lo que se dispondrá de un proyecto vacío con la arquitectura que se ha definido anteriormente. A continuación, se realizan los siguientes pasos previos a la identificación de los servicios:

1. Se carga el sistema heredado en el nivel L0. Todo el código fuente del sistema origen se ubica en este directorio para su procesamiento (ver Fig. 3(a)).
2. Se obtienen los modelos PSM del sistema heredado en el nivel L1 (ver Fig. 3(b)). Se ejecuta la transformación L0-a-L1 (ver Fig. 2 (dcha)).
3. Se obtienen los modelos KDM en el directorio L2 (ver Fig. 3(c)) mediante la transformación L1-a-L2 (ver Fig. 2 (dcha)).

La versión actual de PERSEO soporta la generación de especificaciones *SoaML* basada en el *wrapping* de funcionalidades del sistema identificadas a partir de los modelos KDM. La Fig. 4 muestra cómo, a partir de los modelos KDM, se pueden elegir qué funcionalidades serán presentadas como servicios y cuya información se empleará en la generación de los modelos *SoaML*.

Como se indicaba, la versión actual genera especificaciones básicas de *SoaML* sin embargo (y tal y como se exponía en la sección anterior) la transformación para pasar de los niveles N2 a N3 (de KDM a *SoaML*) no es una única, sino que dependerá de la estrategia de migración que se decida aplicar según el contexto del proyecto. Aunque la versión actual de la transformación es una versión “exploratoria”, ya está en marcha el diseño de las implementaciones de la misma para otras estrategias de identificación de los servicios y su consecuente especificación.

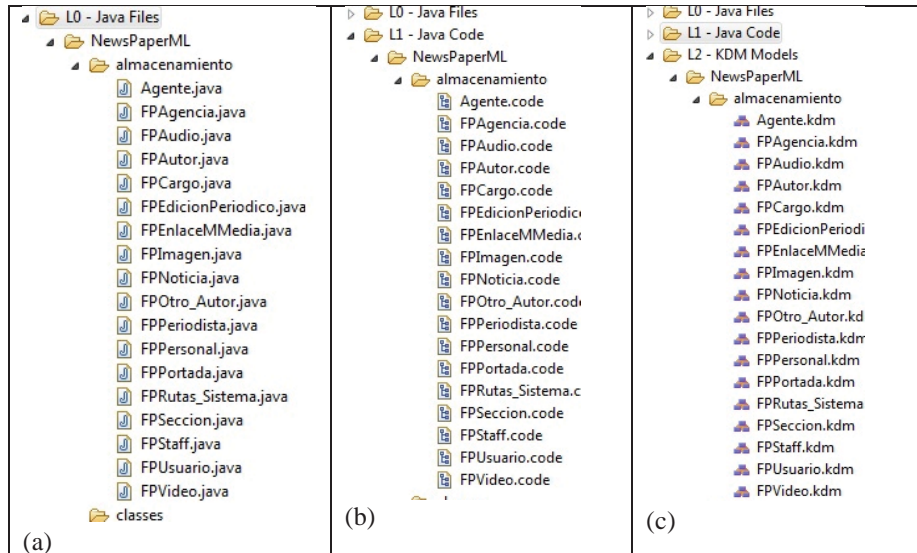


Fig. 3. Aplicación de las sucesivas transformaciones a partir de código (a) para obtener los modelos PSM (b) y PIM (c)

5 Conclusiones

La migración de sistemas heredados hacia servicios ha dejado de ser una promesa de futuro para convertirse en una realidad para las organizaciones actuales. La integración de sistemas y la apertura a nuevas oportunidades de negocio hacen que los Sistemas de Información deban situarse en la nube en forma de servicios, y para esto hay dos opciones (*grosso modo*): (i) los sistemas se desarrollan desde cero con esta orientación; o (ii) los sistemas existentes (la inmensa mayoría de los casos) que soportan la operativa de las empresas deben ser migrados o adaptados a SOA.

Como demuestra el estado del arte, existen múltiples situaciones y circunstancias a la hora de migrar un sistema heredado hacia un nuevo paradigma, y en función de las circunstancias y el contexto, unas opciones de migración serán más adecuadas que otras. Por este motivo, en este artículo se presenta PERSEO, una propuesta de herramienta que pretende ofrecer soporte al proceso de migración de sistemas heredados hacia servicios. La versión actual de PERSEO aborda la generación de especificaciones *SoaML* de forma simple, sin embargo, ya se está trabajando en el desarrollo de otras opciones y estrategias (o patrones) para la generación de servicios.

Principalmente, el trabajo futuro de PERSEO se mueve en dos direcciones: (i) la implementación de nuevas transformaciones para la generación de servicios en distintos modos, y (ii) el desarrollo de un módulo para la generación de código que permita obtener los servicios funcionales listos para desplegar en la nube, ya que la versión actual se centra en la identificación de servicios y su especificación.

Otro de los aspectos que se deberán considerar en PERSEO como parte y complemento al proceso de migración hacia servicios son los procesos de negocio. Tal y

como se puede ver en [6, 7] los procesos de negocio modelados en BPMN son también un complemento necesario para llevar a cabo una correcta migración.

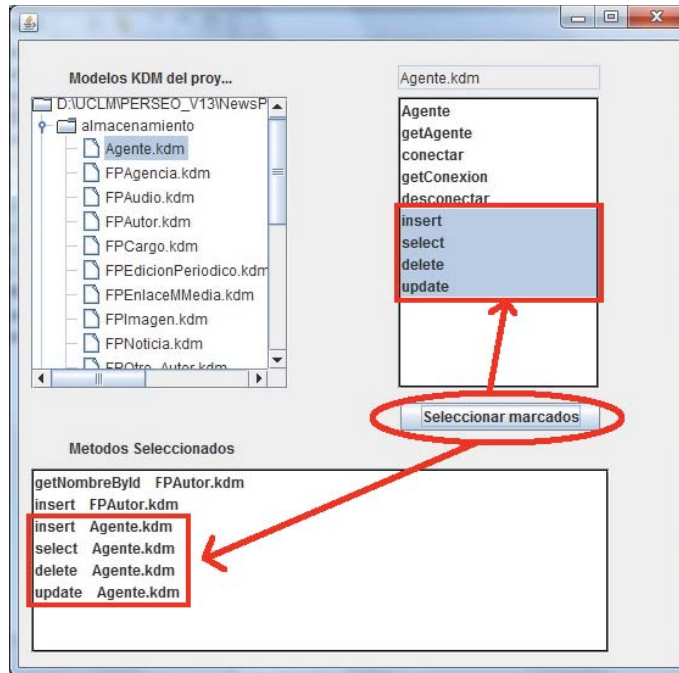


Fig. 4. Panel de configuración de los *MethodUnit* que se migrarán como servicios

6 Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto PISCIS: Paradigmas de Innovación, Social y Colaborativo aplicados a Ingeniería del Software (CDTI, FEDER - INNTERCONECTA – ANDALUCÍA, ITC- 20131007).

7 Referencias

1. Hutchinson, J., et al., *Migrating to SOAs by Way of Hybrid Systems*. T Professional. IEEE Computer Society, 2008. **10**(1): p. 34-42.
2. ISO/IEC, *ISO/IEC DIS 19506. Knowledge Discovery Meta-model (KDM), v1.1 (Architecture-Driven Modernization)*. 2009, ISO/IEC. p. 302.
3. OMG, *QVT. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*. <http://www.omg.org/spec/QVT/1.0/PDF>, 2008, OMG.
4. OMG, *Service Oriented Architecture Modeling Language (SoaML)*, 2012, Object Managent Group.
5. OMG, *MDA. Model Driven Architecture Guide Version 1.0.1*, 2003.

6. Fuhr, A., et al., *Model-driven software migration into service oriented architectures*. Computer Science - Research and Development, 2011. **28**(1): p. 65-84.
7. Zillmann, C., et al. *The SOAMIG Process Model in Industrial Applications*. in *In proceedings of the 15th European Conference on Software Maintenance and Reengineering*. 2011.
8. Lewis, G., E. Morris, and D. Smith. *Analyzing the reuse potential of migrating legacy components to a service-oriented architecture*. in *In Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 2006. CSMR 2006*. 2006.
9. Sneed, H.M., *Planning the reengineering of legacy systems*. IEEE Software, 1995. **12**(1): p. 24-34.
10. Khadka, R., et al. *A Method Engineering based Legacy to SOA Migration Method*. in *In Proceedings of the the 27th IEEE International Conference on Software Maintenance (ICSM'11)*. 2011.
11. Sneed, H.M. *COB2WEB a toolset for migrating to web services*. in *In Proceedings of the 10th International Symposium on Web Site Evolution (WSE'08)*. 2008.
12. Sneed, H.M., I.J.S. Tools, and Technol. Transf., 441-451., *A pilot project for migrating COBOL code to web services*. International Journal on Software Tools for Technology Transfer, 2009. **11**(6): p. 441-451.
13. Smith, H., et al., *The Emergence of Business Process Management*, 2002, CSC's Research Services Report.
14. Erradi, A., S. Anand, and N. Kulkarni. *Evaluation of Strategies for Integrating Legacy Applications as Services in a Service Oriented Architecture*. in *In Proceedings of the IEEE International Conference on Services Computing (SCC 06)*. 2006. Chicago, EEUU: IEEE Computer Society.
15. Aversano, L., L. Cerulo, and C. Palumbo. *Mining candidate web services from legacy code*. in *In Proceedings of the 10th International Symposium on Web Site Evolution, (WSE'08)*. 2008. Beijing: IEEE Computer Society.
16. Zhang, Q., R. Chen, and Y. Zou. *Reengineering User Interfaces of E-Commerce Applications Using Business Processes*. in *Proceedings of the 22st IEEE International Conference on Software Maintenance (ICSM'06)*. 2006. Philadelphia, Pennsylvania: IEEE Computer Society.
17. Zhang, Z. and H. Yang. *Incubating Services in Legacy Systems for Architectural Migration*. in *In Proceedings of the 11th Asia-Pacific Software Engineering Conference (ASPEC'04)*. 2004.
18. Marchetto, A. and F. Ricca, *From objects to services: toward a stepwise migration approach for Java applications*. International Journal on Software Tools for Technology Transfer (STTT), 2009. **11**(6): p. 427-440.
19. García-Rodríguez de Guzmán, I., M. Polo, and M. Piattini. *An ADM Approach to Reengineer Relational Databases Towards Web Services*. in *14th Working Conference on Reverse Engineering (WCRE 2007)*. 2007. Vancouver, British Columbia, Canada: IEEE Computer Society.

20. Jiang, Y. and E. Stroulia. *Towards Reengineering Web Sites to Web-services Providers*. in *Proceedings of the Eighth Euromicro Working Conference on Software Maintenance and Reengineering (CSMR'04)*. 2004. Tampere, Finland: IEEE Computer Society.
21. Zhang, Z., et al., *A service composition approach based on sequence mining for migrating e-learning legacy system to SOA*. *International Journal of Automatic Computing*, 2010. 7(4): p. 584-595.
22. Almonaies, A.A., J.R. Cordy, and T.R. Dean. *Legacy system evolution towards service-oriented architecture*. in *In Proceedings the International Workshop on SOA Migration and Evolution (SOAME 2010)*. 2010. Madrid (España).
23. Zhang, B., et al. *A Black-Box Strategy to Migrate GUI-Based Legacy Systems to Web Services*. in *In Proceedings of the 2008 IEEE International Symposium on Service-Oriented System Engineering (SOSE '08)*. 2008. IEEE Computer Society.
24. Visaggio, G., *Ageing of a data-intensive legacy system: symptoms and remedies*. *Journal of Software Maintenance and Evolution: Research and Practice*, 2001. 13: p. 281-308.
25. Bao, L., et al. *Extracting Reusable Services from Legacy Object-Oriented Systems*. in *In Proceedings of the 2010 IEEE International Conference on Software Maintenance (ICSM)*. 2010. Timisoara: IEEE Computer Society.
26. Chen, F., et al. *Service Identification via Ontology Mapping*. in *In Proceedings of the 33th IEEE International Computer Software and Applications Conference*. 2009. Seattle, Washington (EEUU): IEEE Computer.
27. Marchetto, A. and F. Ricca. *Transforming a java application in an equivalent web-services based application: toward a tool supported stepwise approach*. in *In Proceedings of the 10th International Symposium on Web Site Evolution (WSE'08)*. 2008. Beijing: IEEE Computer Society.
28. Chen, F., et al. *A Formal Model Driven Approach to Dependable Software Evolution*. in *In Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC '06)* 2006. Chicaco, EEUU: IEEE Computer Society.
29. Hoyer, P., et al. *A Model-Driven Development Approach for Service-Oriented Integration Scenarios*. in *In Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*. 2009. IEEE Computer Society.
30. Canfora, G. and M. Di Penta. *New Frontiers of Reverse Engineering*. in *2007 Future of Software Engineering*. 2007. IEEE Computer Society.