

Despliegue semántico de servicios en entornos heterogéneos y distribuidos

Laura Díaz-Casillas, Mercedes Garijo
Departamento de Ingeniería de Servicios Telemáticos
Universidad Politécnica de Madrid
Avenida Complutense 30, Madrid (España)
{ldcasillas, mga}@dit.upm.es

Resumen—En la actualidad, numerosas aplicaciones empresariales se desarrollan siguiendo una arquitectura orientada a servicios. Una de las actividades asociadas a este proceso es el despliegue, el cual debe realizarse sobre varios entornos, adecuados a las necesidades de cada fase del desarrollo. Dichos entornos se caracterizan por ser heterogéneos y distribuidos. En la mayor parte de los casos, el despliegue se realiza de forma manual, siendo una tarea costosa y que genera numerosos errores, existiendo una necesidad de encontrar alguna solución que facilite dicha labor. En este artículo, se propone emplear una ontología mediante la cual se caractericen los elementos involucrados en el despliegue de servicios, considerando sus propiedades funcionales y no funcionales, con el objetivo de realizar un despliegue óptimo, adecuado a las características del entorno de desarrollo concreto sobre el que se trabaja.

Palabras Clave—ontología despliegue servicios

I. INTRODUCCIÓN

El uso de servicios web [1] y arquitecturas orientadas a servicios (o SOA, del inglés *Service Oriented Architecture*) [2] favorece el desarrollo de aplicaciones empresariales, facilitando la interoperabilidad entre sistemas heterogéneos, al definir una manera estándar de anunciar e invocar servicios que actúan de forma independiente, mediante los cuales se alcanzan los objetivos de negocio deseados. Su uso fomenta además la disponibilidad, fiabilidad y escalabilidad de dichas aplicaciones, lo que ha llevado a su implantación en un gran número de empresas.

Una de las actividades a llevar a cabo durante el desarrollo de este tipo de aplicaciones es el despliegue de servicios, la cual consiste en realizar todas las acciones necesarias para poder poner los servicios en funcionamiento. Un servicio es un sistema software capaz de proporcionar una función a un usuario o a otro sistema software. Es habitual que los servicios cooperen entre sí para poder alcanzar el objetivo deseado, siendo necesario comprobar su disponibilidad. Por otro lado, también se deben analizar los requisitos demandados por los servicios para su correcta ejecución, y seleccionar en base a éstos los nodos más adecuados para su despliegue. Es además necesario tener en cuenta que durante el desarrollo de los servicios es habitual que se produzcan modificaciones en sus requisitos de despliegue, variándose las características de los componentes requeridos para su correcta ejecución, lo que da lugar a que las actividades de despliegue cambien a lo largo del desarrollo.

Pero aparte de las requisitos específicos del servicio que se desea poner en funcionamiento, es necesario considerar las características del entorno sobre el que se va a realizar el despliegue. En concreto, los entornos de despliegue de apli-

caciones empresariales se caracterizan por ser heterogéneos, estando constituidos por componentes de diversas características, y fuertemente distribuidos, es decir, los servicios suelen ejecutarse sobre distintos nodos, los cuales se encuentran conectados mediante enlaces de red. Por otro lado, se debe tener en cuenta que durante el desarrollo de las aplicaciones empresariales se suelen utilizar varios entornos de despliegue, para de esta forma adecuarse a las necesidades concretas de cada fase del proceso de desarrollo. Así, el número de recursos empleados en las fases iniciales varía con respecto a los utilizados en producción.

La complejidad de las actuales aplicaciones empresariales hace que el despliegue sea una actividad complicada. Sin embargo, es una tarea que se realiza normalmente de forma manual, siendo por ello una de las principales fuentes de error durante el desarrollo de los sistemas, y que por tanto presenta un alto coste asociado. Este hecho ha dado lugar a la investigación en este campo, con el objetivo de facilitar la realización de dicha tarea, proponiéndose diversas soluciones para poder llevar a cabo el despliegue de los servicios en distintos entornos. En concreto, el despliegue puede realizarse de forma manual o a través de herramientas basadas en *scripts*, lenguajes o modelos. En [3] se realiza un análisis de dichas opciones, llegándose a la conclusión de que la opción manual es la que presenta menos barreras de entrada y mayor facilidad de uso, pero se encuentra muy limitada a la hora de escalar y realizar modificaciones en el sistema, recomendándose las técnicas basadas en modelos.

En este artículo se propone el uso de tecnologías semánticas para realizar el despliegue de servicios de aplicaciones empresariales sobre entornos heterogéneos y distribuidos. En concreto, se propone emplear una ontología mediante la cual modelar los elementos involucrados en dicho proceso, es decir, tanto los recursos físicos como los recursos lógicos. En ella, se tendrán en cuenta requisitos funcionales y no funcionales, para llevar a cabo el proceso de despliegue de la forma más adecuada posible al entorno sobre el que se trabaje.

El resto del artículo se estructura de la siguiente manera. En el capítulo II se muestra un escenario de ejemplo, en el cual se explica una posible aplicación del despliegue semántico de los servicios. En el capítulo III se realiza un estudio previo de las tecnologías empleadas en la solución propuesta, descrita en el capítulo IV. A continuación, en el capítulo V se presentan trabajos relacionados y por último, en el capítulo VI, se exponen las conclusiones y líneas futuras de trabajo.

II. ESCENARIO DE EJEMPLO

Se expone a continuación un escenario de aplicación, enmarcado dentro del desarrollo del proyecto ITECBAN (Infraestructura Tecnológica y Metodológica de Soporte para un Core Bancario), cuyo principal objetivo es la elaboración de una plataforma que sirva como base para la creación de sistemas de gestión destinados al sector bancario, eliminándose las actuales limitaciones de los sistemas de información empleados en entornos financieros.

Las aplicaciones empresariales empleadas en este tipo de áreas se caracterizan por presentar en la mayor parte de los casos una arquitectura orientada a servicios, lo que las permite adaptarse de manera continua y flexible a los cambios que se producen en su alrededor, en respuesta a la demanda del mercado. Durante el desarrollo de este tipo de aplicaciones es necesario el empleo de varios entornos de despliegue, a través de los cuales se evoluciona hasta alcanzar el entorno de producción. Los motivos que conducen a utilizar distintos entornos son diversos, entre ellos destacan el control de acceso, el cual debe ir aumentando a medida que se avanza en el proceso de desarrollo, y la topología del entorno, la cual va cambiando a lo largo del ciclo de desarrollo, haciéndose más compleja en la fase de producción, cuando se requiere ampliar el número de recursos físicos empleados, siendo necesario utilizar sistemas de respaldo. A esta situación se añade el hecho de que dichos entornos son heterogéneos y distribuidos, siendo frecuente que se produzcan cambios en sus topologías, muchas veces debidos al mantenimiento o a la mejora de sus prestaciones.

Surge entonces la necesidad de disponer de una solución que facilite el despliegue de los servicios, adecuándolos a las necesidades y características de cada entorno en concreto. Para ello, se propone utilizar una ontología mediante la cual modelar los recursos presentes en el entorno, y así poder determinar los nodos más adecuados sobre los que desplegar cada uno de los servicios.

En la base de conocimiento de la ontología se almacenará la información de los recursos disponibles en el sistema, para en base a la selección del usuario de los servicios a desplegar sobre un entorno concreto, analizar las necesidades demandadas por éstos junto con las características del entorno sobre el que se desea realizar el despliegue, y así poder determinar la ubicación más adecuada de los servicios y conseguir que éstos funcionen correctamente.

III. ESTUDIO PREVIO

Una ontología es una representación formal de los conceptos pertenecientes a un dominio concreto, en la que se indican sus características y las relaciones entre dichos conceptos.

La definición de los elementos presentes en un dominio mediante una ontología facilita el intercambio de información, permitiendo la integración de datos procedentes de diversas fuentes, como son modelos de información, servicios o comportamiento, posiblemente representados en distintos lenguajes, al pertenecer a sistemas diferentes. Por otro lado, una ontología se caracteriza por ser una representación del conocimiento, permitiendo implementar capacidades de razonamiento sobre ella.

Pero además, la aplicación de ontologías al despliegue de servicios aporta diversas ventajas:

- La representación de los conceptos asociados al despliegue de servicios mediante una ontología posibilita su evolución, añadiendo nuevos elementos e incluso adaptando las propiedades de los elementos existentes a las nuevas características que puedan aparecer en un futuro, mientras no se varíe el modelo original, facilitando con ello la reutilización de conceptos existentes.
- El uso de una ontología permite realizar una verificación del modelo, comprobando su consistencia y detectando redundancias, evitando con ello posibles errores antes de realizar las actividades asociadas al proceso de despliegue.
- La descripción semántica de los recursos empleados facilita su búsqueda, favoreciendo la selección de los más apropiados a cada caso.
- Una ontología representa la información de manera jerárquica, lo que facilita su interpretación al mostrarse de forma más clara las asociaciones y dependencias entre los elementos involucrados en el despliegue que si la representación se realizara de manera textual.

En [4] se analiza cómo el uso de ontologías en sistemas de gestión permite representar conceptos, tales como conjuntos, relaciones de similitud o enlaces entre distintas representaciones de una entidad, que no pueden implementarse con otras tecnologías empleadas en la actualidad para modelar este tipo de sistemas, como es por ejemplo UML (*Unified Modeling Language*).

El lenguaje de ontologías web (u OWL, del inglés *Web Ontology Language*) [5] es un lenguaje de marcado que permite definir clases, mediante las cuales representar los conceptos, y sus propiedades asociadas, a través de las cuales determinar las relaciones y restricciones existentes entre ellas. De este modo, OWL posibilita la construcción de una ontología en la que se modelen todos los elementos partícipes en el despliegue de los servicios.

Las posibilidades de OWL pueden ampliarse mediante SWRL (*Semantic Web Rule Language*) [6], un lenguaje propuesto por el W3C (*World Wide Web Consortium*) que permite definir reglas y combinarlas con la base de conocimiento de OWL. Las reglas proporcionadas se corresponden con cláusulas de Horn, estableciendo una relación entre un antecedente y un consecuente, de manera que cuando se cumplan las condiciones expuestas en el antecedente, las condiciones indicadas en el consecuente deberán cumplirse también. De esta forma, se consigue enriquecer el modelo representado mediante OWL a través de una serie de reglas que ayudan a determinar las actividades a realizar durante el proceso de despliegue.

IV. ONTOLOGÍA PROPUESTA

A continuación, se describe la ontología propuesta, mediante la cual se pretende representar los recursos involucrados en el despliegue de servicios, incluyéndose sus propiedades y las relaciones entre ellos, con el objetivo de facilitar dicho proceso, adecuándolo a las características de cada caso concreto.

Considerando la rápida evolución de los sistemas presentes en la industria, se ha optado por proponer una ontología básica, en la que se consideran las características más representa-

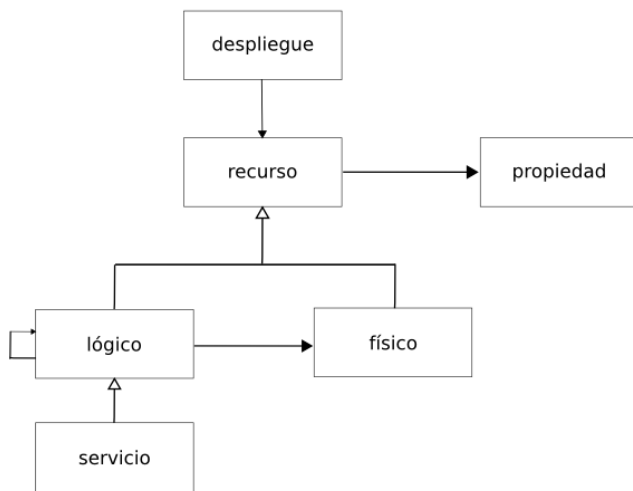


Figura 1. Despliegue del sistema

tivas de los elementos partícipes del despliegue, facilitándose su ampliación y adaptación a necesidades futuras.

Si se analizan los elementos que componen un entorno de despliegue, se observa como éste se encuentra formado por una serie de recursos, físicos y lógicos. Los recursos físicos representan los nodos sobre los que se realizará el despliegue, mientras que los recursos lógicos identifican los sistemas software a desplegar, entre los cuales están los servicios. Los recursos lógicos se distinguen además porque pueden requerir otros recursos lógicos para poder funcionar.

Por otro lado, es necesario establecer una relación directa entre los recursos lógicos y físicos, debido a que cada recurso lógico se desplegará sobre un determinado recurso físico, dotando a éste de un valor añadido, que en algunos casos determinará el despliegue de otros recursos que dependan de él.

Cada uno de los recursos se encuentra identificado por medio de un nombre único y definido mediante una serie de propiedades, caracterizadas a través de uno o más atributos, que llevan asociada una medida para poder cuantificarlos.

Esta información queda reflejada en la figura 1, en la que se muestra una visión general de la ontología propuesta.

IV-A. Descripción de las propiedades de los recursos

Los recursos presentan una serie de propiedades funcionales y no funcionales que los definen.

Cada propiedad se encuentra caracterizada mediante un nombre y uno o más atributos, los cuales a su vez tienen asociada una medida, que permite valorarlos de manera objetiva o subjetiva, mediante un valor y un tipo que determina cómo tratar dicho valor.

Si las propiedades funcionales del nodo se corresponden con las demandadas por los servicios a desplegar sobre él, se consigue una correcta ejecución de los servicios, pero si además se tienen en cuenta características no funcionales, como la disponibilidad o la seguridad de los servicios, se obtiene un despliegue más eficiente y adecuado a las características del entorno de despliegue concreto sobre el que se actúa. Es por ello que se han considerado ambas a la hora de caracterizar los recursos en la ontología.

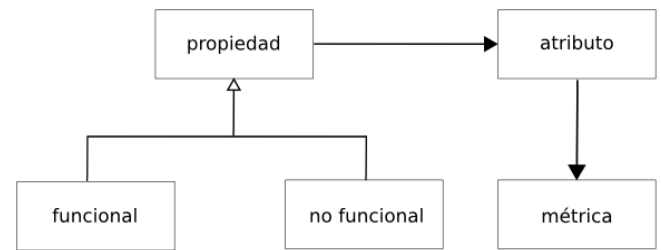


Figura 2. Propiedades

IV-A1. Propiedades funcionales: Para poder realizar un despliegue válido de los servicios es necesario analizar en primer lugar las propiedades funcionales de los recursos involucrados en el proceso. Tras el estudio de las necesidades demandadas por los servicios a desplegar, es necesario conocer los recursos disponibles en el entorno sobre el cual tendrá lugar el proceso, de manera que los servicios se alberguen en los nodos adecuados. Para lograr este propósito, se han considerado cuatro propiedades funcionales:

- Capacidad de procesamiento.
- Disco duro.
- Memoria.
- Conexión en red.

IV-A2. Propiedades no funcionales: Además, en la ontología se han tenido en cuenta propiedades no funcionales, con el objetivo de obtener una calidad de servicio adecuada a la demandada por la aplicación sobre el entorno de despliegue empleado. Teniendo en cuenta que éstos se caracterizan por ser heterogéneos y distribuidos, los aspectos que se han considerado más relevantes son:

- Coste asociado al despliegue del servicio, lo que dependerá del número de recursos requerido para su puesta en funcionamiento y de si éstos se encuentran actualmente en el sistema o es necesario desplegarlos.
- Disponibilidad o probabilidad de que un servicio responda a las peticiones de los clientes, lo que se encuentra directamente relacionado con la capacidad de respuesta del servicio frente a dichas peticiones.
- Fiabilidad o probabilidad de que un servicio se ejecute correctamente, lo que dependerá de las operaciones que realice, si éstas son locales o remotas o si dependen de otros servicios.
- Seguridad, en función de si un servicio requiere autenticación para su uso y codifica los datos que maneja o no.
- Interoperabilidad o facilidad que presenta un servicio para comunicarse con otros.
- Escalabilidad o capacidad de ampliación del servicio, en función de los recursos requeridos para su ejecución.

En la figura 2 se puede observar una visión más detallada de la parte de la ontología dedicada a la definición de las propiedades de los recursos.

IV-B. Descripción de los tipos de recursos

Por otro lado, los recursos se clasifican en dos grandes grupos: físicos y lógicos.

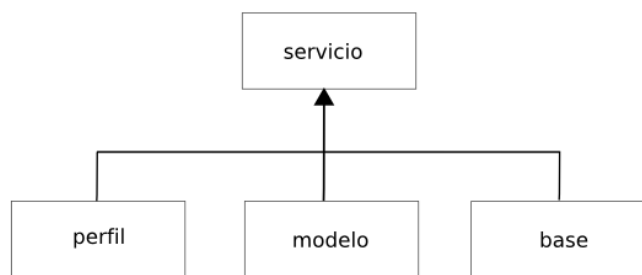


Figura 3. Servicios

IV-B1. Recursos físicos: Los recursos físicos se corresponden con los nodos sobre los cuales se despliegan los servicios. Éstos se caracterizan por tener una ubicación concreta, por tanto deben tener asociados aparte de un nombre, una dirección IP que permita localizarlos.

IV-B2. Recursos lógicos: Los recursos lógicos definen los recursos software a desplegar o necesarios para el despliegue de determinados recursos. Así por ejemplo, para el despliegue de un servicio que gestione una base de datos, será necesario instalar un sistema de gestión de bases de datos en el nodo correspondiente.

Los recursos lógicos se encuentran caracterizados por un nombre único, aunque además es importante indicar su estado: activo o inactivo, según hayan sido desplegados o no en el sistema. Por otro lado, también es necesario indicar su versión para evitar posibles incompatibilidades y facilitar su gestión durante las fases iniciales de desarrollo [7].

Dentro de los recursos lógicos se encuentran los servicios, descritos mediante OWL-S [8]. Ésta es una ontología definida por el W3C que permite incorporar semántica a los servicios web. Su objetivo es automatizar las tareas de descubrimiento, invocación, composición e interoperabilidad de servicios. En OWL-S, los servicios se describen mediante tres tipos de conocimiento:

- El perfil de servicio (*service profile*), que contiene una descripción semántica. Se emplea para anunciar el servicio, indicando su función a los posibles clientes.
- El modelo de servicio (*service model*), mediante el cual se define cómo se utiliza, indicando cómo debe invocarse el servicio y qué sucede durante su llamada.
- La base del servicio (*service grounding*), a través de la cual se define cómo se interacciona con él, proporcionando información acerca de los protocolos de comunicación, formatos de mensajes y otros detalles específicos que permiten conocer cómo acceder al servicio.

En la figura 3 se muestran los elementos que forman parte de OWL-S.

La descripción detallada que se realiza de los servicios al emplear OWL-S permite optimizar el proceso de despliegue. Al disponer de información detallada de cada uno de los servicios, es posible determinar con mayor precisión los nodos más adecuados para su despliegue. Además, el empleo de semántica para la descripción de los servicios permite la posibilidad de realizar un descubrimiento automático, y así por ejemplo, en el caso de que un servicio fallara, sería posible implementar un buscador semántico capaz de encontrar un servicio con características similares que lo sustituyera.

```
<!-- Service description -->
```

```
<service:Service rdf:ID="PersonalInformationService">
  <service:presents rdf:resource="#PersonalInformationProfile"/>
  <service:describedBy rdf:resource="#PersonalInformationProcess"/>
  <service:supports rdf:resource="#PersonalInformationGrounding"/>
</service:Service>
```

Figura 4. Descripción del servicio

```
<!-- Profile description -->
```

```
<profile:Profile rdf:ID="PersonalInformationProfile">
  <service:presentsBy rdf:resource="#PersonalInformationService"/>
  <service:has_process rdf:resource="#PersonalInformationProcess"/>
  <profile:serviceName xml:lang="en">Personal information</profile:serviceName>
  <profile:textDescription xml:lang="en">This service returns the information
  of a person whose ID is given.
</profile:textDescription>
  <profile:hasInput rdf:resource="#ID"/>
  <profile:hasOutput rdf:resource="#PersonalInformation"/>
</profile:Profile>
```

Figura 5. Perfil del servicio

A modo de ejemplo, en las figuras 4 y 5 se muestra parte de la descripción semántica de un servicio realizada mediante OWL-S, en la que se observa como asociados a cada servicio se definen un perfil, un modelo y una base. En el perfil se muestra el anuncio del servicio y los tipos de datos de entrada y salida. En este caso, el servicio proporciona la información de una persona a partir de su DNI.

IV-C. Validación

El uso de una ontología para describir los recursos permite realizar una validación de la información contenida en la base de conocimiento antes de llevar a cabo el proceso de despliegue, eliminando redundancias e incoherencias en los datos. De esta forma, se consiguen evitar posibles fallos en las actividades de despliegue.

Gracias a las restricciones indicadas en la definición de los recursos del sistema, es posible por ejemplo analizar las dependencias de los servicios a desplegar. Siendo necesario comprobar además que los recursos requeridos se encuentran disponibles en el sistema antes de realizar el despliegue, en caso contrario, será necesario ponerlos en funcionamiento.

IV-D. Restricciones adicionales

Se propone emplear SWRL como lenguaje de definición de reglas para complementar las restricciones indicadas mediante la ontología. SWRL permite definir reglas a través de las cuales analizar las propiedades de los recursos involucrados en el despliegue, adecuando la ubicación de los servicios a los nodos disponibles, para lograr un correcto funcionamiento. Como hemos visto en apartados anteriores, se consideran tanto las características funcionales como las no funcionales, con el fin de poder llevar a cabo un despliegue de servicios válido y adecuado al entorno de desarrollo sobre el que se trabaja.

El uso de SWRL permite, además de completar las restricciones impuestas por OWL, aplicar políticas mediante las cuales optimizar el proceso de despliegue. Por ejemplo, analizando los recursos empleados por un servicio y buscando otro servicio con la misma funcionalidad pero que requiera menos recursos. Además, también es posible la aplicación de políticas para realizar un reparto equitativo de carga entre los nodos disponibles o incluso dedicar nodos en exclusiva a determinados servicios o replicar un servicio en múltiples nodos de manera simultánea debido a las características de éste.

```

service (?a) ∧ node(?b) ∧ isLocated(?a, ?b)
⇒ swrlb:subtract(memory(?b), memory(?a))

```

Figura 6. Uso de SWRL I

```

service (?a) ∧ node(?b) ∧ swrlb:lessThan(memory(?a), memory(?b))
∧ node(?c) ∧ swrlb:lessThan(memory(?b), memory(?c))
⇒ deploy(?a, ?c)

```

Figura 7. Uso de SWRL II

En la figura 6 se muestra una regla implementada con SWRL, mediante la cual se permite añadir una restricción en la que el valor de una variable depende del valor de otra variable, limitación que no es posible implementar en OWL. En este caso, el servicio a está desplegado sobre el nodo b, por tanto, el valor de la memoria disponible en el nodo b depende de la memoria requerida por el servicio a. En el ejemplo de la figura 7, se emplea SWRL para definir una regla más compleja, cuyo objetivo es realizar un reparto equitativo del uso de memoria sobre los distintos nodos involucrados en el despliegue. En concreto, se compara la memoria disponible en los nodos b y c, realizándose el despliegue del servicio a sobre el nodo c, en el caso de que éste disponga de más memoria que el nodo b.

Se muestran sólo dos ejemplos del uso de SWRL en el proceso de despliegue de los servicios, siendo necesaria la aplicación de muchas otras reglas para poder obtener un resultado óptimo.

V. TRABAJOS RELACIONADOS

El despliegue de servicios en aplicaciones empresariales es una actividad compleja, que requiere tiempo y esfuerzo, y que normalmente se realiza de forma manual, siendo una de las principales fuentes de error y que por tanto más coste supone en el desarrollo de las aplicaciones.

Este hecho ha originado la propuesta de diversas ideas para solventar esta situación, dichas soluciones se encuentran basadas en *scripts*, lenguajes o modelos. Aunque la última opción es la que puede resultar más complicada en un principio, es también la que presenta mayores posibilidades a la hora de escalar el sistema y por ello es la empleada en este artículo. Pero existen otras propuestas en las que se emplean modelos para representar los elementos involucrados en el proceso de despliegue con el objetivo de facilitar las actividades requeridas para llevar a cabo dicho proceso, aunque ninguna de ellas se ha establecido como estándar. Así por ejemplo, en [9] se desarrolla un metamodelo mediante el cual se describen los parámetros de configuración requeridos para el despliegue de servicios, y en [10] se presenta otro metamodelo que trabaja a más alto nivel, centrado en la gestión de los servicios empleados por las aplicaciones, facilitando su compartición y reutilización. Los modelos suelen encontrarse integrados en sistemas más amplios, con el objetivo de automatizar las tareas de despliegue, como es por ejemplo [11], donde se propone una arquitectura de despliegue dirigida por políticas que permite adaptarse a entornos de despliegue distribuidos y heterogéneos, o [12], donde se define un entorno para

el despliegue de servicios asociados a sesiones sobre un entorno distribuido. En dichas iniciativas se emplean modelos implementados con UML, lo que limita sus posibilidades.

Dicha limitación ha dado lugar a propuestas en las que se enriquecen los modelos mediante el uso de ontologías. Así en [13] se emplea OWL-S para describir los servicios, el entorno de despliegue y el propio proceso de despliegue, con el objetivo de obtener un resultado óptimo en entornos dinámicos. En [14] se propone una ontología extensible implementada en OWL que permite crear infraestructuras adaptables al contexto en el que se encuentren, adaptando las necesidades de los usuarios a las características concretas de los dispositivos y el entorno sobre el que se despliegan los servicios. En [15] y [16] se emplea OWL para definir ontologías mediante las cuales se modela el contexto para facilitar la autogestión de sistemas distribuidos orientados a objetos. En estos trabajos, se puede observar como el uso de ontologías mejora el proceso de despliegue, pero no tienen en cuenta las características no funcionales, con lo que limitan las posibilidades de obtener una despliegue óptimo.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

El desarrollo de aplicaciones empresariales requiere el empleo de varios entornos de despliegue, heterogéneos y distribuidos, adecuados a las necesidades de cada fase del desarrollo. El despliegue es un proceso que se realiza normalmente de forma manual, dando lugar a numerosos errores, siendo necesario el desarrollo de una solución que facilite el correcto despliegue de los servicios sobre dichos entornos.

En este artículo, se propone una ontología mediante la cual se representan los recursos involucrados en el despliegue de los sistemas, considerándose tanto sus propiedades funcionales como funcionales, y sobre la que se aplican reglas, para obtener una correcta adecuación a las necesidades de cada entorno concreto.

Siguiendo el modelo de computación autonómica propuesto por IBM [17], como línea futura se propone ampliar la solución presentada para que el sistema sea capaz de reaccionar ante los cambios que se produzcan en el entorno tras su despliegue, determinando las acciones más adecuadas a realizar en cada caso para mantener el sistema en un estado óptimo de funcionamiento, solventando los posibles errores que puedan producirse.

AGRADECIMIENTOS

Este trabajo ha sido cofinanciado por el Ministerio de Industria, a través del programa de Consorcios Estratégicos Nacionales en Investigación Técnica (CENIT) mediante el proyecto ITECBAN, y por el Ministerio de Ciencia e Innovación, a través del Plan Nacional de I+D+I, por medio del proyecto T2C2 (Tecnologías Telemáticas para la Colaboración Ciudadana - TIN2008-06739-C04-01).

REFERENCIAS

- [1] W3C, "Web Services Activity," <http://www.w3.org/2002/ws/>, 2009.
- [2] S. Hashimi, "Service-Oriented Architecture Explained," O'Reilly Media, Inc., 2003.
- [3] V. Talwar, D. Milojevic, Q. Wu, C. Pu, W. Yan, and G. Jung, "Approaches for Service Deployment," *IEEE Internet Computing - Service-Oriented Computing Track*, 2005.

- [4] J. Strassner, D. O'Sullivan, and D. Lewis, "Ontologies in the Engineering of Management and Autonomic Systems: A Reality Check," *Journal of Network and Systems Management*, 2007.
- [5] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C Recommendation, 2004.
- [6] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Member Submission, 2004.
- [7] H. H. Shahri, J. A. Hendler, and A. A. Porter, "Software Configuration Management Using Ontologies," *3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2007)*, 2007.
- [8] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic Markup for Web Services," W3C Member Submission, 2004.
- [9] Y. Li, J. Qiu, K. Sun, and Y. Chen, "Modeling and Verifying Configuration in Service Deployment," *IEEE International Conference on Services Computing (SCC'06)*, 2006.
- [10] A. Chazalet and P. Lalanda, "A Meta-Model Approach for the Deployment of Services-oriented Applications," *IEEE International Conference on Services Computing (SCC' 2007)*, 2007.
- [11] J. L. Revuelta, "A policy-driven, model-based software and services deployment architecture for heterogeneous environments," Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Tech. Rep., 2007.
- [12] T. Villemur and E. Hammami, "Design and evaluation of a context-aware service deployment for collaborative sessions," *Computer Communications*, 2008.
- [13] H. Chamekh and F. le Mouel, "An Ontology-based Approach to Semantically Deploy Services in Pervasive Environments," *IEEE International Conference on Pervasive Services*, 2007.
- [14] D. Preuveneers, J. V. den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. D. Bosschere, "Towards an Extensible Context Ontology for Ambient Intelligence," in *Second European Symposium on Ambient Intelligence*, 2004.
- [15] A. R. Haydarlou, M. A. Oey, B. J. Overeinder, and F. M. T. Brazier, "Using Semantic Web Technology for Self-Management of Distributed Object-Oriented Systems," *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.
- [16] Y. Zhou, J. Pan, X. Ma, B. Luo, X. Tao, and J. Lu, "Applying ontology in architecture-based self-management applications," in *2007 ACM Symposium on Applied Computing*, 2007.
- [17] J. O. Kephart and D. M. Chess, "The Vision of autonomic computing," *IEEE Computer Society*, 2003.