

Encadenamiento de servicios web: Hacia IDEs basadas en servicios

Carlos Granell

Michael Gould

Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I (Castellón), {carlos.granell, gould}@uji.es

Resumen: *Presentamos una metodología para la creación de servicios integrados de información espacial mediante el encadenamiento de servicios web, asegurándonos que en todo momento nuestra propuesta es conforme a la recomendación europea INSPIRE en términos de interoperabilidad e integración. El modelo propuesto de encadenamiento de servicios gira en torno al concepto de componente integrado, visto como una pieza básica de reutilización, y en una metodología para la creación, composición, reutilización y transformación de tales componentes integrados en procesos ejecutables para las aplicaciones de usuario. Finalmente, demostramos la viabilidad del modelo propuesto en un escenario real relacionado con la creación de sistemas ad hoc de información geoespacial en caso de situaciones de emergencia.*

INTRODUCCIÓN

Las Infraestructuras de Datos Espaciales (IDE) engloban políticas, acuerdos institucionales, bases datos espaciales, servicios sobre estos datos, y tecnologías que facilitan la disponibilidad y el acceso a los servicios y a los datos [7]. Esta definición está ampliamente aceptada pero todavía está centrada en el intercambio y acceso a los datos y servicios de tipo geoespacial y los que siguen ciertas especificaciones OGC. Avanzando un paso más allá, en [2] los autores se preguntan cuáles serán los siguientes retos relativos a la investigación de IDEs. Existen diversos problemas aún por resolver (como por ejemplo la interoperabilidad semántica) pero nosotros nos centraremos en cómo es posible la encadenación de diferentes servicios de información geográfica que soporten tareas complejas y específicas, con otros servicios relacionados pero no servicios OGC per se. De cara al usuario, la encadenación de servicios resulta interesante porque no solo permite acceder a los datos geoespaciales disponibles por los servicios primarios (geoportales) sino que permite la encadenación ad hoc de servicios simples disponibles para formar nuevos servicios compuestos. De esta forma, la clásica definición de IDE como facilitador de datos geoespaciales se convierte en una visión mucho más amplia, proporcionando geo-servicios (o servicios en general) que envuelven explícita o implícitamente datos geoespaciales [1]. Esta nueva visión implica que las IDEs necesitan apoyarse mucho más en la funcionalidad de los servicios que en los propios datos geoespaciales, ya que la clave reside en la combinación de funcionalidades ofrecidas por diferentes servicios que acceden a su vez a sus respectivos datos geoespaciales.

Recientemente se está prestando mucha atención a la iniciativa europea para el desarrollo de una Infraestructura de Datos Espaciales en Europa (INSPIRE) [6], debido en gran parte a su adopción (en primera instancia) por parte del Parlamento Europeo en junio del 2005. El objetivo que persigue INSPIRE es crear una directiva marco que guía y instruye a los estados miembros sobre la creación de infraestructuras de información espacial a nivel nacional y local que proporcione a usuarios finales (ya sea administración, empresas, otras organizaciones, así como a ciudadanos) servicios integrados de información geoespacial. Es interesante destacar este último concepto de *servicios integrados* que aparece en los borradores de INSPIRE. Los usuarios finales buscan servicios que se adapten a sus requisitos. Normalmente, no existe un único servicio que satisfice completamente sus demandas, sino que servicios individuales satisfacen distintos aspectos de los requisitos del usuario. La idea, pues, consiste en proporcionar al usuario una cadena de servicios formada por la composición de otros servicios más simples o genéricos. Solo que dicha cadena de servicios debe ser finalmente entregada al usuario con la apariencia de un solo servicio integrado de información geoespacial (un servicio opaco según la terminología de la norma ISO 19119). Podríamos equiparar entonces la cadena de servicios resultante por la composición de servicios geoespaciales con el servicio integrado de información espacial propuesto por la recomendación de INSPIRE.

Ya existen diversos trabajos en el área de los servicios web aplicados al contexto geográfico que revelan la importancia de este campo en el desarrollo y evolución de las IDEs. El *Open Geospatial Consortium* (OGC) ha dedicado varias iniciativas consecutivas para abordar y promover la especificación de los clásicos servicios OpenGIS (WMS, WFS, WCS, etc.) según la arquitectura de servicios web (www.w3.org/2002/ws/arch/). La iniciativa más reciente, *OGC Web Services Phase 3* (OWS-3, www.opengeospatial.org/initiatives/?iid=162), está encargada de la definición de una arquitectura que permita la integración e interoperabilidad de diferentes servicios OGC (sensores, LBS) descritos o expuestos como servicios web. También el OGC está impulsando diversos experimentos en el marco de los servicios de

geo-procesamiento como el experimento *Web Process Service Interoperability Experiment* (WPS IE, www.opengeospatial.org/initiatives/?iid=148), llevado a cabo para testear la viabilidad de las especificaciones de interfaces OWS para permitir servicios de geo-procesamiento a través de Internet.

Por otra parte, en el área de encadenación de servicios, la Agencia Espacial Europea (ESA, www.esa.int) está desarrollando un ambicioso proyecto denominado *Service Support Environment* [3] que proporciona una arquitectura para la integración de servicios tanto de la Observación de la Tierra como de SIG proporcionados por diferentes *partners* distribuidos en nueve países europeos. Las composiciones de servicios resultantes son transformadas en procesos WS-BPEL [10]. En el OGC también se han finalizado recientemente los primeros experimentos encadenando diversos WCS mediante el uso de los lenguajes WSDL y WS-BPEL (véase el documento OGC 04-078, *OWS 2 Image Handling for Decision Support: Service Chaining with BPEL 1.0*). Parece lógico pensar que uno de los pilares básicos para el avance de las IDEs y, en definitiva, de INSPIRE, resida en la capacidad de ofrecer nuevos servicios encadenando otros ya existentes.

En este artículo presentamos un modelo para la creación de servicios integrados de información espacial mediante el encadenamiento de servicios web, asegurándonos que en todo momento nuestra propuesta es conforme a la recomendación europea INSPIRE [6], por lo menos a los borradores disponibles hasta el momento. Para eso, antes de introducirnos en nuestro modelo, en la siguiente sección definimos el papel de nuestro modelo (y en general la encadenación de servicios) dentro del contexto de INSPIRE. El resto del artículo presenta nuestro modelo de encadenamiento de servicios basado principalmente en el concepto de componentes integrados y en una metodología para gestionar dichos componentes integrados. Luego mostraremos cómo encadenar servicios mediante componentes integrados en un escenario de gestión de emergencias, demostrando que la salida de nuestra propuesta se ajusta perfectamente al servicio integrado de información espacial enunciado por INSPIRE. Finalmente, terminaremos con conclusiones acerca de la composición de servicios para el desarrollo de una IDE junto con nuestro plan de trabajo para el futuro inmediato.

INFRAESTRUCTURAS DE SERVICIOS ESPACIALES

Uno de los principales retos de INSPIRE se centra en la interoperabilidad de servicios para la documentación, publicación, descubrimiento y consumición de información geográfica tanto a nivel europeo, nacional, regional y local. La Figura 1 muestra la arquitectura de referencia (conceptual) de INSPIRE para permitir la interoperabilidad entre estos servicios [8]. Básicamente, esta arquitectura está compuesta de cuatro componentes bien diferenciados. Las aplicaciones de usuario (o cliente) que consumen tanto datos como servicios geográficos. Típicamente, los datos geográficos están almacenados en repositorios cuyos metadatos son publicados en catálogos para que puedan ser buscados tanto por las aplicaciones de usuario como por los servicios. La pieza central de la Figura 1 que une aplicaciones de usuario, catálogos y repositorios representan los servicios (middleware, de transformaciones, de geoprocésamiento, etc.). Los servicios permiten, por ejemplo, que las aplicaciones de usuario accedan a servicios de búsqueda de catálogo o bien que los datos contenidos en los repositorios sean procesados antes de ser consumidos por la aplicación usuario.

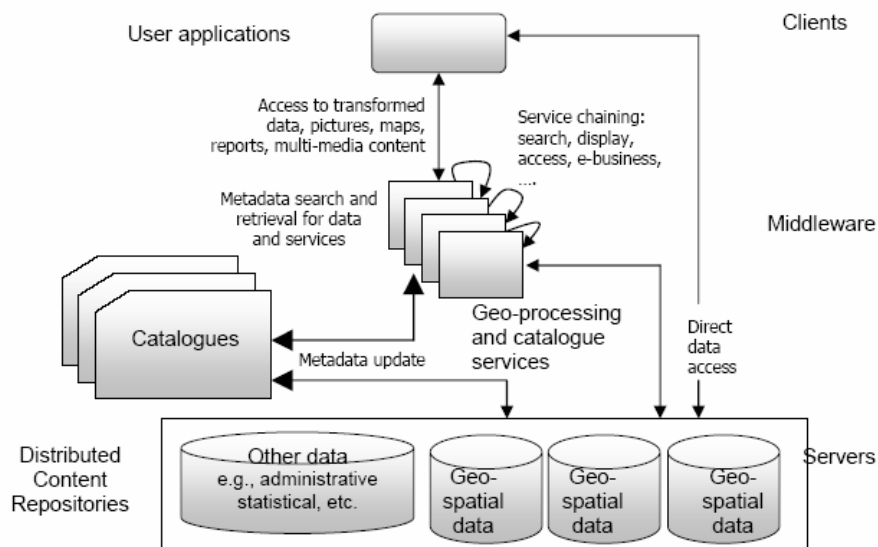


Figura 1: Arquitectura de referencia de INSPIRE (extraído de [8])

Sin embargo, los catálogos no mantienen únicamente metadatos de los datos geospaciales sino también de los servicios disponibles. De esta forma, ante peticiones complejas de las aplicaciones de usuario, los servicios de búsqueda recuperan otros servicios más simples que, encadenados, ofrecen los requisitos demandados. Nuestro trabajo se centra justo en el aspecto de la encadenación de servicios, operación ya contemplada en la Figura 1, en la sección de middleware. La idea consiste en ofrecer un modelo para componer servicios existentes basados en los principios de simplicidad, reutilización y flexibilidad. La reutilización es un objetivo prioritario en nuestro modelo y, en cierta manera, se corresponde con la flecha “*metadata update*” que aparece en la Figura 1. Por ejemplo, a existencia de servicios A y B será documentada en los metadatos de un registro o catálogo de servicios, pero en caso de crear un macro-servicio C, resultado de encadenar a A y B, también los metadatos del nuevo servicio serán enviados al catálogo. Nuevos servicios compuestos quedan almacenados en el catálogo para que sean reutilizadas en peticiones posteriores. En definitiva, como presentamos en la siguiente sección, nuestro modelo genera cadenas de servicios reutilizando otras ya existentes. Como se comentaba en la introducción, la reutilización de servicios es esencial ya que permite tanto la encadenación de servicios sin tener que empezar desde servicios básicos cada vez, así como también disponer de conocimiento previo sobre cadenas de servicios existentes aplicadas a problemas similares.

ENCADENAMIENTO DE SERVICIOS MEDIANTE COMPONENTES INTEGRADOS

El objetivo en esta sección es introducir nuestro modelo basado en componentes integrados que soporta la reutilización y composición de servicios. En primer lugar daremos una visión general de la arquitectura del modelo, sus principales componentes y las entradas y salidas, para luego describir el concepto de componente integrado y la metodología de composición propuesta.

Arquitectura

El modelo propuesto hereda directamente ciertas características de la ingeniería del software basada en componentes según la definición de Syzperski [9]. Este tipo de sistemas se caracteriza por tener bien definido un modelo de componente y una metodología de composición. El modelo de componente define cómo deben ser descritos los componentes para mejorar su reutilización. La metodología de composición describe los mecanismos utilizados para combinar tales componentes. Así pues, en nuestro modelo, el modelo de componente se corresponde con el concepto de *componente integrado* y la metodología de composición con la *metodología de composición de servicios*.

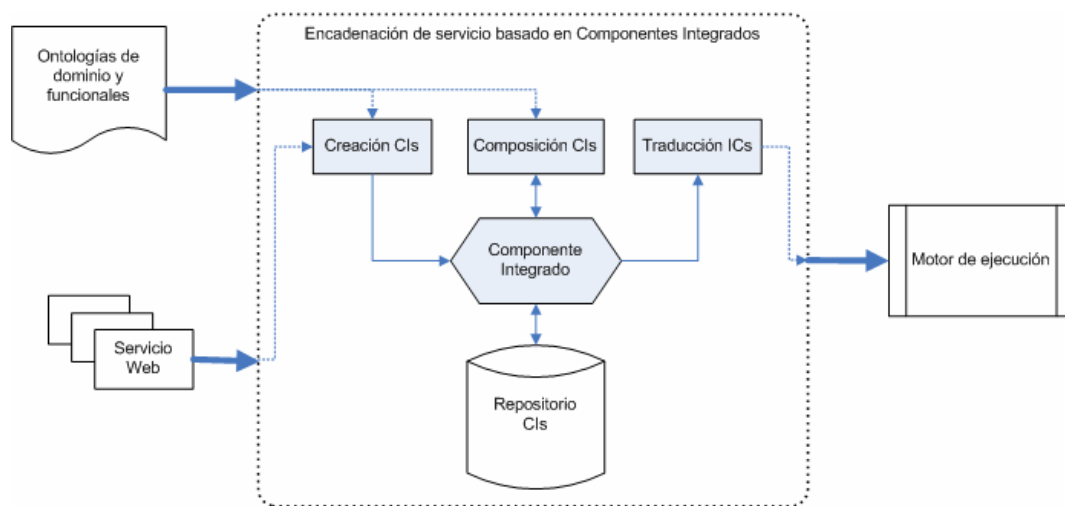


Figura 2: Interacciones entre los componentes del modelo.

La Figura 2 muestra las relaciones entre la metodología propuesta y los componentes integrados con respecto a las entradas y salidas de nuestro modelo. Los servicios web existentes y las ontologías de dominio forman las entradas de nuestro modelo. En este momento vale la pena comentar ciertas restricciones asumidas en este trabajo respecto a las ontologías. Nos basamos en ontologías dominio, generalmente definidas por expertos, en el contexto geográfico con el objetivo de anotar semánticamente ciertos aspectos de un componente integrado. La idea es experimentar el uso de la semántica en la descripción y composición de servicios de un modo gradual, proporcionando una herramienta complementaria que ayude al usuario en el proceso de composición. En este trabajo, la semántica es un aspecto minoritario ya que el proceso de composición expuesto en las siguientes secciones sigue siendo dirigido

fundamentalmente por la “semántica” del usuario. Pues bien, volviendo a la Figura 2, los componentes integrados son creados y combinados por medio de los diferentes procesos que forman la metodología de composición de servicios. Así, el primer proceso consiste en crear componentes integrados a partir de los servicios web y las ontologías. Estos componentes integrados son almacenados en un catálogo para su posterior reutilización. El modo de crear servicios de cierta complejidad es combinando dichos componentes integrados produciendo gradualmente otros nuevos. Los nuevos componentes integrados resultado de la composición son igualmente registrados en el repositorio para su futuro uso. Finalmente, cuando se desea disponer de un proceso ejecutable el correspondiente componente integrado es transformado en un documento WS-BPEL [10], la salida de nuestro modelo, listo para ser ejecutados por un motor de ejecución de procesos WS-BPEL.

Componentes integrados

Un componente integrado se define como un servicio que adopta algunos atributos y aspectos de la tecnología de componentes, proporcionando una pieza autónoma de código, reutilizable e independiente, junto con la utilización de los patrones de workflow que indican cómo deben combinarse los componentes integrados contenidos en cierta composición. El resultado de combinar componentes integrados se considera un nuevo componente integrado. Para conseguir este comportamiento es necesario que los componentes integrados sean unidades independientes, reutilizables y ofreciendo cierta encapsulación sobre detalles de su estructura. Nuestra estrategia consiste en capturar mediante descripciones abstractas todos los aspectos relevantes que definen a un componente integrado. Por lo tanto, en el diseño de un componente integrado consideramos los siguientes aspectos clave que nos ayudarán a perfilar la estructura de un componente integrado:

- Los aspectos descriptivos como metadatos referidos al contexto de la funcionalidad de un componente integrado.
- Los aspectos funcionales detallan la funcionalidad o las capacidades de un componente integrado en términos de la operación que ofrece y sus parámetros de entrada y salida.
- Los aspectos estructurales muestran como un componente integrado está internamente estructurado como combinación de servicios u otros componentes más simples, utilizando para ello los patrones de workflow.
- Los aspectos de enlace definen el flujo de datos tanto entre instancias de servicios web y componentes integrados como entre los propios componentes integrados que forman una composición.

Estos aspectos son encapsulados en un componente integrado combinado mediante dos interfaces funcionales. La interfaz pública expresa al mundo exterior los aspectos descriptivos y funcionales del servicio. La interfaz privada representa una vista interna del componente integrado encapsulando las características estructurales como el flujo de control y de datos así como las transformaciones necesarias para los aspectos de enlaces. De esta forma, el acceso a un componente integrado es controlado por la interfaz pública, proporcionando la necesaria encapsulación para permitir componentes integrados reutilizables e independientes de la composición en la que están contenidos.

Metodología de composición de servicios

La creación, composición y transformación de componentes integrados es realizada por los tres procesos que integran la metodología de composición de servicios. El Proceso de Abstracción de Servicios (PAS) está encargado de la creación de componentes integrados; el Proceso de Composición de Servicios (PCS) combina (y reutiliza) componentes integrados existentes para crear otros nuevos con la funcionalidad requerida; el Proceso de Transformación de Servicios (PTS) transforma las descripciones de los componentes integrados en procesos ejecutables. La Figura 3 ilustra el ciclo de vida de los componentes integrados, descrito en más detalle en las siguientes secciones. La descripción completa de los tres procesos que componen la metodología de composición de servicios puede encontrarse en [4, 5].

Proceso de abstracción de servicios

El proceso PAS (parte izquierda de la Figura 3) consiste en crear componentes integrados a partir de servicios web ya existentes en el catálogo. Para este proceso suponemos que no existe ya un componente integrado con la funcionalidad requerida por el usuario. Si no es así, evidentemente, se debe reutilizar dicho componente integrado en cuestión. La Figura 3 muestra los dos pasos (1-2) necesarios para la creación de componentes integrados.

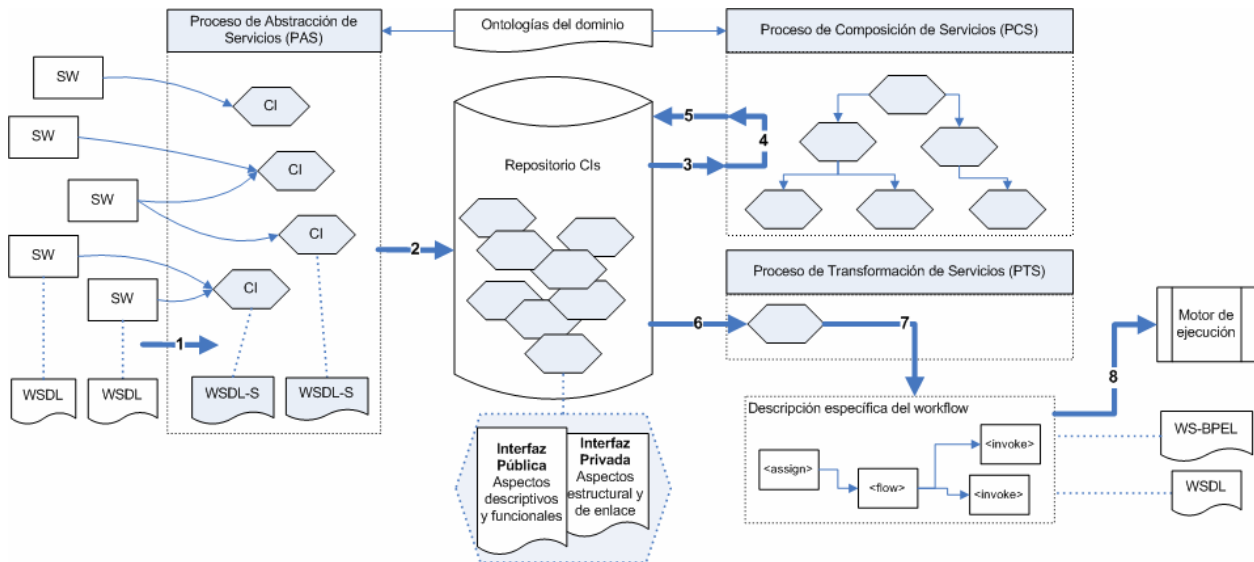


Figura 3: Arquitectura detallada para la creación, composición y transformación de componentes integrados.

El proceso de abstracción, marcado como 1 en la Figura 3, permite especificar los aspectos que describen a un componente integrado: descriptivos, funcionales, estructurales y de enlace. En este momento se describe la funcionalidad del componente (aspectos funcionales) y los patrones de selección¹ utilizados (aspectos estructurales). Una vez definido el nuevo componente integrado, el siguiente paso (2, Figura 3) actualiza el catálogo para que el nuevo componente éste disponible para ser reutilizado. Aunque se trate de un paso sencillo, es de vital importancia para maximizar la reutilización del modelo. Volviendo a la arquitectura de referencia mostrada en la Figura 1, el registro del componente integrado se corresponde con la acción de “*metadata update*”, ilustrando como los propios servicios actualizan el catálogo con sus metadatos. En nuestro caso, ya que los componentes integrados tienen realmente la interfaz de un servicio web, la descripción WSDL del componente integrado (que se corresponde con su interfaz pública) son los metadatos que se publican en el catálogo de servicios.

Proceso de composición de servicios

El PCS, representado por el ciclo 3-4-5 en la Figura 3, es el encargado de construir aplicaciones complejas como componentes integrados, resultado de gradualmente agregar y reutilizar otros componentes integrados ya disponibles en el repositorio. Esto nos proporciona un nivel añadido de simplicidad, independencia y reutilización durante el proceso de composición necesario con el fin de obtener una respuesta favorable a la pregunta: ¿es posible maximizar el nivel de reutilización de composiciones de servicios ya existentes?.

Al igual que el proceso de abstracción (1, Figura 3), el proceso de composición (4, Figura 3) crea nuevos componentes integrados a partir de otros ya existentes definiendo sus aspectos básicos (descriptivos, funcionales, estructurales, y de enlace). Esta vez, los componentes integrados que van a formar parte de la composición son recuperados del catálogo (3, Figura 3), siendo reutilizados en la nueva composición. Como antes, el nuevo componente integrado es publicado en el catálogo como un nuevo componente listo para ser reutilizado (5, Figura 3).

Proceso de transformación de servicios

Una vez creado el componente integrado siguiendo los dos procesos anteriores, utilizamos el proceso PTS para transformar la descripción de dicho componente integrado en un lenguaje de procesos ejecutable. Concretamente, en nuestro modelo hemos optado por el lenguaje WS-BPEL [10] por disponer de múltiples motores de ejecución.

El proceso de transformación se corresponde con los pasos 6-7-8 de la Figura 3. En primer lugar, el componente integrado que va a ser ejecutado es seleccionado del catálogo (6, Figura 3). Una vez seleccionado, se lleva a cabo el proceso de transformación propiamente dicho. Todos los componentes integrados que forman parte de la composición seleccionada son transformados a un único proceso WS-BPEL, que finalmente puede ser ejecutado por el motor de ejecución correspondiente (8, Figura 3).

¹ En [5] se analiza la distinción entre patrones de selección y de composición dentro de nuestro modelo. Básicamente, los patrones de selección son utilizados durante el proceso PAS mientras que los de composición durante el proceso PCS.

APLICACIÓN DEL MODELO DE COMPONENTES INTEGRADOS

La demostración de nuestro modelo consiste en un escenario relacionado con la mitigación de desastres frente a situaciones de emergencia. Con este escenario ya estamos familiarizados ya que se trata de uno de los pilotos desarrollados durante el proyecto europeo ACE-GIS (www.acegis.net). Suponemos pues, que se produce un escape de gas tóxico en una planta química en los alrededores de un área poblada. Es evidente que los responsables de gestionar la seguridad ciudadana activen planes de emergencias para evacuar dicha área si fuera necesario, con el fin de ofrecer una respuesta lo más rápida posible al desastre ocurrido.

Antes de dar luz verde al plan de evacuación, los responsables de la seguridad deben cerciorarse si, definitivamente, la nube de gas tóxica se dirige hacia la zona urbana. Deciden entonces monitorizar la nube de gas tóxica cada cierto tiempo para verificar su tamaño y localización exacta. Tras una breve discusión, se propone poner en marcha un servicio compuesto que monitorice la nube de gas tóxica, *servicio de monitorización*, el cual tomará como entrada el nombre de la planta química y devolverá un mapa del área afectada mostrando la situación y tamaño actual de la nube de gas tóxica. La cadena de servicios que forma el *servicio de monitorización* es la siguiente. En primer lugar utilizan un servicio de *gazetteer* que toma como entrada un topónimo o nombre de lugar (como el nombre de nuestra planta química) y devuelve sus coordenadas geográficas (latitud/longitud). El siguiente paso consiste en recuperar las condiciones meteorológicas en dicha localización. Tanto la dirección como velocidad del viento son parámetros necesarios para el cálculo de la nube de gas tóxica. Como es habitual, no se encuentran estaciones meteorológicas en cada localización terrestre y deciden entonces utilizar un servicio de proximidad para localizar el aeropuerto más cercano a la planta química (todos los aeropuertos disponen de estación meteorológica). Así, el servicio *aeropuerto más cercano* devuelve el código de dicho aeropuerto dada una determinada localización, en nuestro caso la localización de la planta química. Con el código de aeropuerto ya pueden interrogar el servicio meteorológico para obtener los parámetros de viento necesarios. El equipo de responsables ya tiene todos los datos necesarios (localización, dirección y velocidad del viento, y la densidad del gas) para invocar el servicio *calcula dispersión nube de gas* que devuelve un polígono representando la nube de gas tóxica (de hecho este servicio envuelve a un WFS). El último paso consiste en superponer el polígono sobre un mapa centrado en el área del desastre utilizando para ello un WMS. En la figura 4 aparece representada la descripción lógica de la cadena de servicios que forma el servicio *monitorización* mediante las flechas discontinuas en negro que unen los componentes integrados del nivel inferior (*CI gazetteer*, *CI proximidad*, *CI meteorológico*, *CI calcula dispersión nube de gas*, y *CI visualiza mapa*).

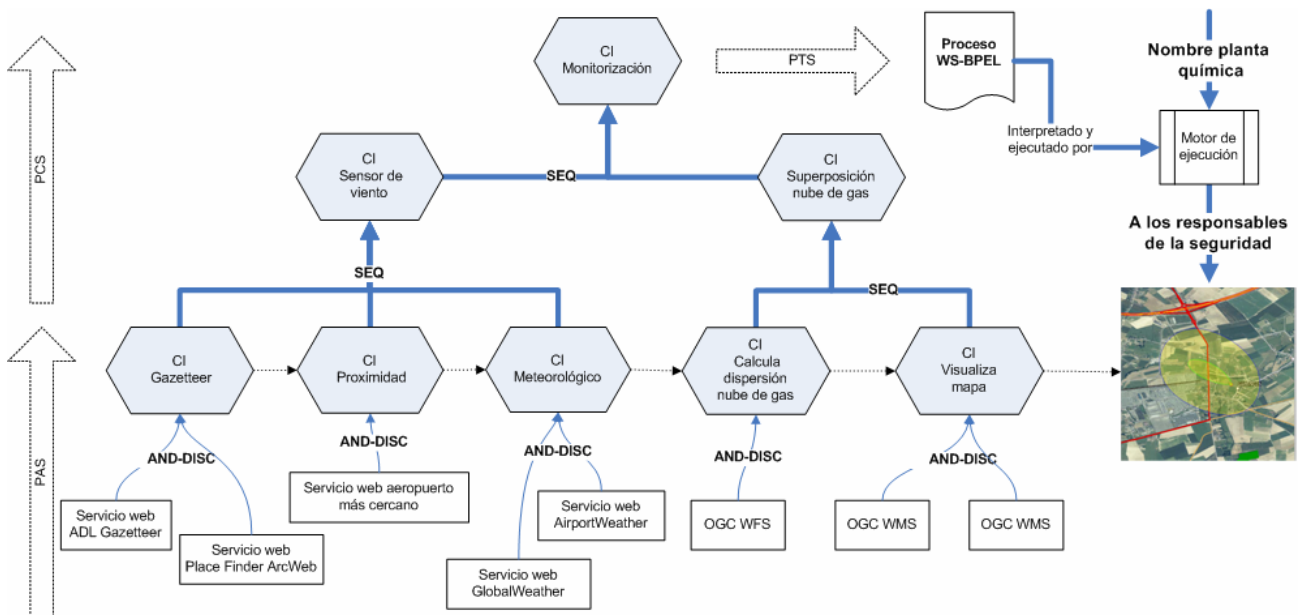


Figura 4: Componente integrado *monitorización* encadenando componentes integrados.

Pues bien, ¿cómo se diseña el servicio de *monitorización* mediante el modelo de componente integrados?. El primer paso consiste en la creación de los correspondientes componentes integrados a partir de los servicios web disponibles en el catálogo. Este paso consiste en aplicar el Proceso de Abstracción de Servicios (PAS). La parte inferior de la Figura 4

muestra los servicios web disponibles. Por ejemplo, para crear el correspondiente *gazetteer* como componente integrado podemos hacer uso de dos servicios ya disponibles como el *ADL gazetteer* (*Alexandria Digital Library gazetteer*, www.alexandria.ucsb.edu/gazetteer/) y el servicio *Place Finder* proporcionado por ESRI (arcweb.esri.com/arcwebonline/). En este momento vale la pena señalar que aunque nuestro objetivo es potenciar la integración de servicios OGC dentro de una IDE, el servicio OGC *gazetteer* (véase el documento OGC 05-035 *Gazetteer Profile of the Web Feature Service Implementation Specification*) todavía se encuentra en discusión y, hasta nuestro conocimiento, no conocemos de ninguna implementación realmente funcionando². Tanto *ADL gazetteer* como *Place Finder* no son servicios OGC. Sin embargo este hecho demuestra que nuestro modelo es independiente del contexto, siendo lo bastante general para ser aplicado a distintos contextos.

Volviendo a nuestro escenario, el componente integrado *gazetteer* consiste de hecho en dos servicios modelados mediante el patrón de selección *AND-DISC* [5]: *ADL gazetteer* y *Place Finder*. Tal como se ha comentado anteriormente, durante el proceso de abstracción y composición se utilizan patrones de selección y composición para definir los atributos estructurales de un componente integrado. El patrón de selección *AND-DISC* es uno de los patrones disponibles. La utilización del patrón de selección *AND-DISC* para el componente integrado *gazetteer*, que contiene dos servicios concretos (*ADL gazetteer* y *Place Finder*), no significa que en tiempo de ejecución los resultados de ambos *gazetteer* serán considerados, sino que el primer resultado obtenido será tenido en cuenta en el flujo de datos de la cadena. Por ejemplo, si *ADL gazetteer* responde antes que *Place Finder*, los resultados de éste último son ignorados por el motor de ejecución. Los patrones de selección aportan gran flexibilidad al modelo ya que un componente integrado no está ligado en tiempo de diseño a un determinado servicio web sino a un conjunto de servicios web candidatos. El resto de componentes integrados del mismo nivel son creados del mismo modo.

El siguiente paso es componer y reutilizar, si es posible, componentes integrados. Como ilustra la figura 4, el nuevo componente integrado *Sensor de viento* es una composición de los tres componentes integrados justo debajo. Los componentes integrados *gazetteer*, *proximidad*, y *meteorológico* son combinados mediante el patrón de composición secuencia (SEQ en la Figura 4). De hecho, tales componentes integrados son realmente reutilizados porque ya están disponibles en el catálogo desde el momento en que son creados. De la misma forma, tanto *sensor de viento* como *superposición nube de gas* son incluidos en el catálogo para que puedan ser reutilizados de nuevo, creando de este modo el componente integrado *monitorización* buscado por los responsables de la seguridad en nuestro escenario. Tan sólo queda transformar el componente integrado *monitorización* en un proceso WS-BPEL ejecutable mediante el proceso PTS, para que sea finalmente interpretado y ejecutado por un motor de ejecución compatible.

CONCLUSIONES

En este artículo hemos presentado un modelo para el desarrollo de aplicaciones web para la composición y reutilización de servicios centrado alrededor del concepto de componente integrado. El componente integrado se convierte en la pieza básica y modelo de reutilización en otras composiciones. Para manejar y gestionar componentes integrados, hemos presentado una metodología de composición basada en tres procesos: el Proceso de Abstracción de Servicios para la creación de componentes integrados; el Proceso de Composición de Servicios para reutilizar componentes integrados existentes en nuevas composiciones; y el Proceso de Traducción de Servicios para convertir las descripciones de los componentes integrados en procesos WS-BPEL ejecutables.

Sin embargo, creemos que todavía queda un largo camino para alcanzar un nivel maduro en la encadenación de servicios personalizados en el marco de la IDE. Varios autores [1, 2] ya han identifican la necesidad de desplazar la visión actual de la IDE centrada en los datos a un visión mucho más amplia que abarque además servicios distribuidos geográficamente a diferentes niveles (nacional, local, etc.) y, como último fin, la visión de una infraestructura espacial dirigida completamente por servicios. Por lo tanto, el continuo desarrollo de soluciones innovadores para la encadenación de servicios en el marco de la IDE e INSPIRE, acercará cada vez más la visión de una infraestructura de *servicios* espaciales.

Nuestro trabajo más inmediato está siendo el testeo de nuestro modelo de componentes integrado en otros contextos como el reciente proyecto europeo AWARE (www.aware-eu.info), aplicando nuestro modelo en el encadenamiento de servicios para proporcionar mapas sobre previsiones de recursos hídricos en entornos de alta montaña. Además, otra de nuestras líneas futuras sigue siendo la integración de nuestro modelo en IDEs [4] con el fin de contribuir un poco más al desarrollo y evolución de las infraestructuras de datos y servicios espaciales en su faceta más tecnológica.

² Existe un OGC *gazetteer* en http://ogc.compusult.nf.ca/OGC/gaz_get_search.html, aunque no se trata de una versión estable.

REFERENCIAS

1. Bernard, Lars; Craglia, Max (2005): SDI - From Spatial Data Infrastructure to Service Driven Infrastructure. En *Proceedings of the First Research Workshop on Cross-learning on Spatial Data Infrastructures and Information Infrastructures*, Enschede (The Netherlands). <http://gi-gis.jrc.it/ws/crosslearning/papers/PP> Lars Bernard - Max Craglia.pdf (último acceso: Octubre 2005)
2. Bernard, Lars; Craglia, Max; Gould, Michael; Kuhn, Werner (2005): Towards an SDI Research Agenda. *Proceedings of the 11th EC-GIS Workshop*, Sardinia. <http://www.ec-gis.org/Workshops/11ec-gis/presentations/07kuhn.pdf> (último acceso: Octubre 2005)
3. D'Elia, Sergio; Marchetti, Pier G. (2005): Service Support Environment – Web-Services based ESA Infrastructure for Geospatial Services. En *Proceedings of the IGARSS 2005 Conference*. Seoul, Korea.
4. Granell, Carlos; Gould, Michael; Ramos, Francisco (2005): Service Composition for SDIs: integrated components creation. En *Proceedings of the DEXA Workshops 2005*, Copenhagen (Dinamarca). IEEE CS Press, 475-479
5. Granell, Carlos; Gould, Michael; Grønmo, Roy; Skogan, David (2005): Improving Reuse of Web Service Composition. En *Proceedings of the EC-Web 2005*, Copenhagen (Dinamarca). Springer LNCS 3590, 358-367
6. INSPIRE Initiative. <http://inspire.jrc.it> (último acceso: Septiembre 2005)
7. Nebert, Doug (ed.) (2004): Developing Spatial Data Infrastructures: The SDI Cookbook. GSDI Cookbook, version 2, GSDI. <http://www.gsd.org/docs2004/Cookbook/cookbookV2.0.pdf> (último acceso: Octubre 2005)
8. Smits, Paul et al (2002): INSPIRE Architecture and Standards Position Paper. Architecture and Standards Working Group. http://inspire.jrc.it/documents/inspire_ast_pp_v4_3_en.pdf (último acceso: Octubre 2005)
9. Szyperski, Clemens (1988): *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley
10. WS-BPEL (2005): OASIS Web Services Business Process Execution Language 2.0 – 1 septiembre 2005, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wspml (último acceso: Octubre 2005)