

Modelado Arquitectónico Orientado a Servicios de Estrategias de Coordinación Inspiradas en Tácticas Deportivas

Marcos López-Sanz, Carlos E. Cuesta, Esperanza Marcos, Jesús Domínguez

Departamento de Lenguajes y Sistemas II, Universidad Rey Juan Carlos
C/ Tulipán, S/N – 28933 Móstoles (Madrid)

{marcos.lopez, carlos.cuesta, esperanza.marcos, jesus.dominguez}@urjc.es

Resumen. El paradigma orientado a servicios ha sido considerado tradicionalmente como la solución a los problemas de integración entre aplicaciones de distintas organizaciones. Uno de los aspectos que más influyen a la hora de integrar diferentes aplicaciones es la forma de coordinar los diferentes servicios que participan en la solución de integración. Las coreografías de servicios permiten coordinarlos sin que ninguno de ellos dirija el proceso de coordinación, sino que todos son considerados como iguales. Este artículo expone el desarrollo de estrategias de coordinación basadas en coreografías de servicios utilizando un enfoque dirigido por modelos y centrado en la arquitectura. La validez de la propuesta de modelado se demuestra utilizando una metáfora deportiva en un entorno de cooperación real: una situación de juego puntual en el transcurso de un partido de baloncesto.

Palabras clave. SOA, Desarrollo Dirigido por Modelos, Estrategias de Coordinación, Situaciones Tácticas Deportivas, Coreografías de Servicios

1 Introducción

El paradigma de la orientación a servicios (*Service-Oriented Computing*, SOC) [26] se relaciona habitualmente con la posibilidad de integrar e interconectar aplicaciones pertenecientes a diferentes organizaciones o empresas [14]. Es por ello que uno de los aspectos que más importancia tiene durante el desarrollo de soluciones orientadas a servicios sea, junto con el dinamismo, la composición de servicios [27]. Desde un punto de vista conceptual, la composición de servicios ha de entenderse como la coordinación de las funcionalidades ofrecidas por cada uno de ellos con el fin de alcanzar una meta común.

La coordinación entre servicios puede darse, principalmente, de dos formas: como orquestación o como coreografía [28]. En el primero de los casos uno de los participantes actúa de coordinador definiendo qué funcionalidad de qué servicio es la que debe ejecutarse en cada momento. Esta opción es la que tradicionalmente se utiliza para la implementación de procesos de negocio basados en servicios que realizan tareas pertenecientes a flujos de trabajo complejos [5]. En otros casos, sin embargo, la consecución de un objetivo o de una funcionalidad concreta no se basa en la ejecución secuencial de una serie de operaciones, sino que se obtiene a partir de la interrelación entre servicios equivalentes, en dicho caso se habla de coordinación de servicios basados en coreografías. El entorno que habitualmente se utiliza como ejemplo de integración basada en coreografías es el de la gestión de mercados [12][35] aunque, como se verá más adelante, no es el único.

En los últimos años han aparecido estrategias de diseño de coreografías basadas en tecnologías y estándares de bajo nivel tales como WS-Coordination [21], WS-CDL [31] o WS-CF [9], por nombrar algunos de los múltiples lenguajes existentes. No obstante, el contexto de aplicación de la integración de aplicaciones a gran y media escala es lo suficientemente complejo como para

requerir un enfoque de desarrollo de más alto nivel que no sea tan dependiente de la tecnología de implementación.

Para poder obtener una visión más intuitiva de la integración basada en servicios es necesario seguir una estrategia conceptualmente más genérica. Uno de los enfoques de desarrollo que más importancia y atención está recibiendo en los últimos años es el enfoque dirigido por modelos y en particular la propuesta de MDA (*Model-Driven Architecture*, [19]).

Actualmente, existen múltiples metodologías que permiten diseñar y representar soluciones orientadas a servicios [2][6][15][32][34]. Sin embargo, hasta donde nosotros conocemos, pocas o ninguna de ellas aborda el problema de la integración de aplicaciones a nivel de modelado desde el punto de vista de la coordinación de servicios basada en coreografías. El objetivo principal de este artículo es demostrar que la representación de estrategias de coordinación mediante modelos es viable para ser aplicada en un entorno orientado a servicios.

No obstante, para facilitar el modelado del sistema sería recomendable disponer de una metáfora adecuada para describir las coreografías de servicios a alto nivel. Se plantea como analogía válida la descripción de sistemas tácticos en deportes de equipo. Ésta tiene la ventaja de proporcionar una serie de estrategias de coordinación ya predefinidas, lo que la convierte en un esquema particularmente interesante. Además, como objetivo secundario, se pretende demostrar que el uso de sistemas tácticos para deportes de equipo proporciona una estrategia básica de coordinación en el contexto descrito.

En nuestro caso, utilizaremos la descripción de modelos tácticos existentes para deportes de equipo [33] como metáfora para desarrollar el modelado de arquitecturas orientadas a servicios en las que se requiere una coordinación entre ellos. La analogía deportiva resulta adecuada en nuestro caso por tratarse de un entorno en el que entidades aparentemente independientes (jugadores representados por servicios) se comunican entre sí con el fin de obtener un objetivo o funcionalidad común (o individual, según sea el caso) [29]. Mediante esta metáfora intentamos resolver el problema de la integración de aplicaciones que puedan ser transformadas en servicios o que puedan ofrecer sus funcionalidades de forma similar. La reducción del problema a modelar coreografías de servicios nos permite centrarnos en la comprobación de la viabilidad de tal aproximación.

Utilizaremos, como estrategia de coordinación a modelar, una determinada situación de juego: el *bloqueo directo* [22] durante el transcurso de un partido de baloncesto. Esta estrategia servirá para demostrar la viabilidad de la estrategia de modelado para definir coreografías de servicios y el método de desarrollo utilizado.

La estructura del artículo es la que sigue: en la sección 2 se expone el contexto metodológico dirigido por modelos en el que se enmarca la estrategia de modelado de coreografías de servicios. En la sección 3 se presenta una estrategia concreta de coordinación y se analiza la modelización de arquitecturas de servicios que representan coreografías. La sección 4 resume algunos de los principales trabajos relacionados y finalmente la sección 5 presenta las principales conclusiones y líneas de investigación abiertas a partir del tema presentado en el artículo.

2 Contexto metodológico

La composición de servicios como solución al problema de la integración de aplicaciones no se restringe a la modelización de la coordinación en sí, sino que tiene que llevar asociado una metodología de desarrollo que permita crear una solución integral basada en servicios. Por ello, en esta sección se presenta, brevemente, el marco metodológico SOD-M (*Service-Oriented Development Method*, [7]). En él se engloba el estudio de las estrategias de coordinación de servicios basadas en coreografías. Además, dichas estrategias se analizarán desde el punto de vista de la arquitectura, el cual permite una mejor representación de las relaciones entre los diferentes elementos que participan en la solución software [17].

El enfoque dirigido por modelos (*Model-Driven Development*, MDD) se ha postulado como una de las principales alternativas de elección para el desarrollo de sistemas. Este enfoque se caracteriza por utilizar el concepto de *modelo* como artefacto principal del proceso de desarrollo. Una de las propuestas concretas de MDD es MDA [23], que añade a la característica anterior la idea de dividir el proceso de desarrollo en conjuntos de modelos en diferentes niveles de abstracción (CIM, PIM y PSM). Otra de las principales aportaciones de este enfoque es la definición de transformaciones automáticas entre los modelos de diferentes niveles y entre modelos del mismo nivel.

Como se ha comentado anteriormente, la idea de utilizar un enfoque dirigido por modelos para el desarrollo de soluciones orientadas a servicios no es nueva. Existen múltiples metodologías que abordan esta problemática [2][6][15][32][34]. Una de ellas es SOD-M [7] que aboga por el desarrollo de sistemas siguiendo un enfoque ACMDA (*Architecture-Centric Model-Driven Architecture*), tal y como se expone en [17].

SOD-M sigue un enfoque dirigido por modelos multidimensional, donde cada dimensión representa un aspecto del proceso de desarrollo. Por un lado, SOD-M recoge el enfoque MDA a través de la separación en niveles de abstracción describiendo los modelos a definir en cada nivel. Por otro lado, la arquitectura juega un papel director dentro de la metodología ya que, con la creación de un modelo de la arquitectura, es posible identificar qué elementos se incluyen en cada modelo de cada nivel de abstracción. Finalmente, SOD-M es una metodología orientada a servicios, ya que el concepto principal en el que se basa el proceso de desarrollo es el servicio. Por ello, se han definido desde un metamodelo de servicios de usuario hasta modelos de descripción de la arquitectura a diferentes niveles.

Consideramos que el primer contacto real con la estrategia de coordinación, durante el proceso de desarrollo del sistema, se establece en el modelado de la arquitectura a nivel PIM (*Platform Independent Model*). En este nivel se definen aquellos modelos que son independientes de la plataforma y que por lo tanto se abstraen totalmente de cualquier aspecto tecnológico. En el siguiente apartado se exponen las principales características de este modelo.

2.1 Metamodelo de Arquitecturas Orientadas a Servicios

Esta sección expone, de forma breve, el metamodelo de arquitecturas orientadas a servicios de nivel PIM (ver [16] para una descripción más detallada). Este metamodelo tiene como elemento central la definición del concepto de servicio, entendido como *un mecanismo que permite el acceso a un conjunto de una o más capacidades donde dicho acceso es ofrecido a través de un interfaz bien conocido y se lleva a cabo de acuerdo a las restricciones y políticas especificadas en la descripción del servicio* [19].

Todo servicio además pertenece a una determinada organización, modelada como proveedor de servicios (*ServiceProvider*), que representa a aquellas organizaciones que controlan cada servicio que pertenecerá a la solución software. Cada una de estas entidades puede pertenecer a uno de los dos tipos de proveedores definidos: *proveedores externos* de servicios o *proveedores internos*.

Por otra parte, todo servicio se identifica en la solución software mediante un identificador único, denominado *SERVID*, que permite identificar, de forma unívoca cada instancia de servicio existente en el sistema. Además, cualquier servicio ofrece su funcionalidad a través de *operaciones*, consideradas como funcionalidades atómicas independientes. Cada operación puede ser de dos tipos: síncrona o asíncrona, dependiendo de si el servicio que invoca la operación debe esperar o no por la respuesta a la operación (si es que existe).

Los servicios se relacionan entre sí a través de conectores arquitectónicos representados por los *Contratos de Servicio*. Cada uno de estos contratos permite la comunicación punto a punto entre servicios caracterizado por un patrón de intercambio de mensajes. Este patrón puede ser de varios tipos, destacando las opciones de: *One-Way* o comunicación en una sola dirección, *Query/Response* en cuyo caso el tipo de comunicación que se da está formado por la petición de invocación de la operación en un sentido y el envío de la respuesta por otro (de forma síncrona o

asíncrona, según sea la operación); finalmente, el protocolo de comunicación puede estar basado en un intercambio complejo de mensajes, en cuyo caso hablaremos de *Dialogue*.

Los servicios que participan en una solución software basada en este paradigma pueden clasificarse según diferentes criterios. Dependiendo del tipo de operaciones que implementan, podemos diferenciar entre *servicios de interacción*, en el caso de que implementen al menos una operación de tipo síncrona (este tipo de servicio no existe a nivel de implementación, pero se representa a nivel de modelado con el fin de identificar a aquellos servicios capaces de iniciar conversaciones con otros servicios); o *servicios tradicionales* en los que las operaciones son únicamente asíncronas. Por otro lado, de acuerdo al rol realizado por el servicio dentro de la arquitectura del sistema, podemos distinguir entre *servicios básicos*, que ofrecerán funcionalidades propias del sistema software al que pertenecen; o *servicios de soporte*, que representan a aquellos servicios cuya funcionalidad no está relacionada con los requisitos funcionales del sistema sino que realizan tareas de soporte, tales como descubrimiento de servicios o tareas de orquestación en servicios compuestos, necesarias para que el resto de servicios cumplan con su cometido. Finalmente, los servicios pueden clasificarse de acuerdo a su atomicidad, pudiendo, de esta forma, identificar *servicios simples* y *servicios compuestos*. Este aspecto es el que más nos interesa enfatizar puesto que es donde hemos de fijarnos a la hora de diseñar estrategias de coordinación basadas en servicios.

La Figura 1 muestra parte de este metamodelo. En concreto, se muestran los elementos necesarios para poder representar la composición de servicios.

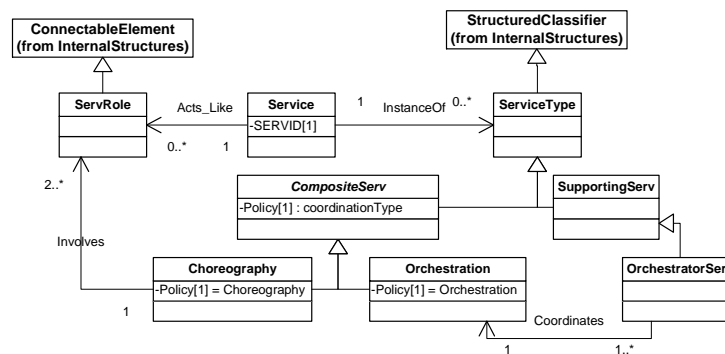


Fig. 1. Parte del metamodelo con los elementos que intervienen en la composición de servicios

Los *servicios compuestos*, o la definición de nuevos servicios a partir de la coordinación de servicios simples más propiamente, pueden ser de dos tipos: orquestaciones y coreografías. Las *orquestaciones*, desde el punto de vista de un modelo arquitectónico de servicios, necesariamente incluirán un servicio de tipo soporte que actúe según el rol de orquestador y que implemente un flujo de trabajo basado en la invocación de operaciones del resto de servicios del sistema. Las *coreografías*, por el contrario, no necesitan de un servicio de soporte especial y se estudiarán más en detalle en la sección siguiente.

3 Modelado Arquitectónico de Coreografías de Servicios

En esta sección exponemos nuestra visión de modelado de coreografías orientadas a servicios usando la metáfora del *bloqueo directo* [22] como ejemplo de estrategia básica de coordinación.

3.1 Descripción de la metáfora de coordinación

La estrategia coordinación que se utilizará como medio para explicar el modelado de coreografías será una situación táctica denominada *bloqueo directo*. Para ello, primero debemos de definir una serie de factores que nos conducirán a una mejor comprensión de esta acción ofensiva.

Partimos de una situación de juego dos-contra-dos (2c2), donde el uso de *bloqueo directo* es una de las distintas opciones de ataque que conFiguran nuestro sistema de juego ofensivo. Wissel [33] determina como en el bloqueo directo se verán implicados cuatro jugadores (dos atacantes y sus defensores), donde los dos primeros interaccionaran con el objetivo de librar a cada uno de ellos de su defensor mediante una obstrucción legal de su movimiento, así el bloqueador *irá a buscar* al defensor que marca al jugador con balón para interrumpir su desplazamiento, dejándole momentáneamente libre de su defensa.

Los roles que se derivan de esta situación táctica son cuatro: *Atacante* o playmaker (atacante con balón, AB), *compañero* o *bloqueador* (atacante sin balón, ASB), *bloqueado* o *defensor* (defensor del jugador con balón, DB) y *ayudante* o defensor del bloqueador (defensor del jugador sin balón, DSB). Cada uno de estos roles estará compuesto de una serie de acciones propias y específicas a cada uno de ellos [33]. La tabla 1 muestra algunas de ellas.

Tabla 1. Acciones asociadas a cada uno de los jugadores participantes en el bloqueo directo.

Acciones tácticas ofensivas		Acciones tácticas defensivas	
<i>Atacante con balón (AB)</i>	<i>Bloqueador (ASB)</i>	<i>Defensor (DB)</i>	<i>Ayudante (DSB)</i>
Parar y tirar	Bloquear	Negar bloqueo	Flash defensivo corto
Penetrar	Ganar posición en continuación	Pasar de 2º	Flash defensivo largo
Pasar a la continuación	Apoyar	Cambio	Cambio
Aclarado	Desmarcarse	Jugar 2c1	Jugar 2c1

Al igual que en otros ámbitos de coordinación basados en coreografías, la aplicación del bloqueo directo en el transcurso de un ataque no será el resultado de la intervención directa y predefinida de uno de los participantes en el sistema, que actúe, en un momento puntual, como coordinador principal (en este ejemplo: un entrenador dando las indicaciones de las tareas a realizar por parte de los jugadores en el terreno de juego), sino que éste viene dado gracias a interrelación que existe entre los diferentes elementos que lo componen y que buscan el mismo objetivo (atacante y compañero buscan el tanto, mientras que defensor y ayudante intentan evitarlo). Además, cada uno de los participantes jamás actuará de igual manera en las mismas jugadas, sino que sus acciones estarán determinadas por las acciones, decisiones o movimientos del resto de los participantes de la coreografía. Por ejemplo, ante la misma jugada, el jugador con balón optará por: pasar a su compañero si se ha desmarcado con claridad después del bloquear, lanzar a canasta si su defensor ha quedado perfectamente bloqueado, realizar un re-bloqueo en caso de no haber conseguido tanto mi compañero como yo no la ventaja necesaria, etc.

A este respecto Ociepa [22] alude al uso de las distintas posibilidades ofensivas en el bloqueo directo en función de las distintas variables que se presenten (tipo de defensa del equipo contrario, características de los atacantes, marcador ajustado, etc.) para plantear gran cantidad de problemas a la defensa y encontrar la solución adecuada.

3.2 Representación de los elementos arquitectónicos de una coreografía

Como se ha comentado en apartados anteriores, nuestra propuesta de desarrollo de soluciones de integración basada en servicios y centrada en la representación de estrategias de coordinación mediante modelos, se realiza dentro de la metodología SOD-M. Esta metodología utiliza UML como notación para los modelos que define, y, por lo tanto, las coreografías de servicios se representan mediante alguno de los conceptos incluidos en la versión 2.0 de UML [23].

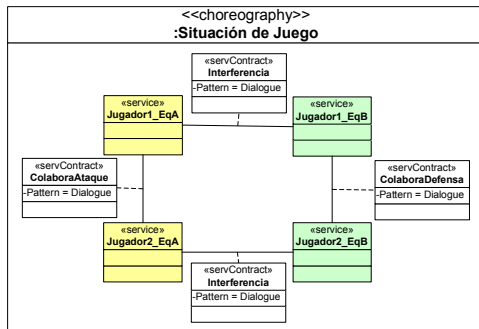


Fig. 2. Modelización del servicio compuesto.

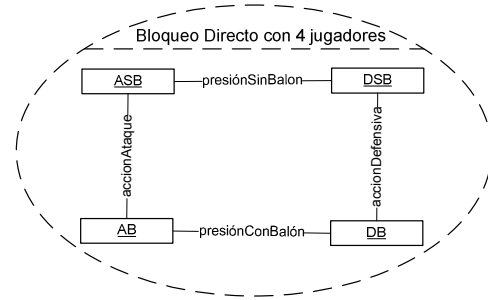


Fig. 3. Modelado de roles como una colaboración para una situación táctica de bloqueo directo.

En nuestro metamodelo, todos los tipos de servicios heredan de *StructuredClassifier* (que a su vez hereda de *Classifier*). Puesto que los servicios compuestos son un tipo de servicio, es posible representar cualquier tipo de coordinación dentro de nuestro modelo. En nuestra metáfora deportiva entendemos como *esquema de coordinación* aquél que representa una situación de juego en la que participan diferentes jugadores. Para nosotros, cada uno de los jugadores será una instancia del tipo de servicio jugador. La situación de juego será representada, por lo tanto, como un servicio compuesto que involucra a distintos *servicios simples* (jugadores) (ver Figura 2).

Cuando un servicio participa en una coreografía debe hacerlo actuando según un rol determinado. En UML 2.0 los roles se representan como elementos conectables (*ConnectableElement*) y así será como se representan dentro del metamodelo de la arquitectura de servicios de nivel PIM (ver Figura 1). Cuando un servicio actúa siguiendo un rol determinado, tendrá asociado un subconjunto de las operaciones que ofrece. Es decir, todo servicio puede realizar un conjunto determinado de operaciones, pero sólo podrá ejecutar, dentro de la coreografía en la que toma parte, algunas de ellas de acuerdo al rol que tenga asignado en ese momento. Cada *ConnectableElement* admite la definición de operaciones como parte de su descripción. En nuestro caso, las operaciones de este tipo de elemento se corresponderán con las operaciones que puede realizar un servicio cuando actúa según un rol determinado. En este punto se puede observar cómo se separa, por un lado, la definición de los servicios (instancias de un tipo de servicio representando a cada jugador) junto a los contratos establecidos entre ellos, de los roles que juegan en cada momento de la situación táctica. De esta forma, podemos redefinir las coreografías como el intercambio de mensajes que se produce como consecuencia de la interacción de los roles jugados por cada instancia de servicio. Estos mensajes provendrán de la invocación de las operaciones definidas en cada rol y de las acciones que éstos realizan en consecuencia.

En UML 2 las relaciones entre los roles de cada elemento conectable se definen a través de una colaboración (*Collaboration*). Siguiendo con la metáfora deportiva, para nosotros, cada una de estos diagramas de colaboración representará una situación táctica concreta en la que cada jugador tiene asignado un rol determinado (ver Figura 3). El conjunto de todos los diagramas de colaboración será lo que se considere como un sistema táctico completo. Cada situación táctica tendrá asociado un conjunto de variantes de evolución del juego que serán representadas con sus correspondientes diagramas de secuencia en los que se representan las invocaciones de las operaciones definidas para cada rol.

Estas dos descripciones independientes se relacionan del modo previsto en UML 2.0, mediante una correspondencia definida en un *CollaborationUse*, tal como se explica en la sección 3.4.

El número de combinaciones tácticas posibles para una misma situación de juego es bastante elevado por lo que representarlas utilizando lenguajes de bajo nivel no es trivial ni sencillo. Con la metáfora deportiva es posible comprobar cómo el enfoque dirigido por modelos es de gran ayuda a la hora de desarrollar soluciones software en las que sea necesaria la implementación de estrategias de coordinación basadas en coreografías. El siguiente apartado se centra en analizar la comunicación entre los diferentes roles involucrados en la coreografía.

3.3 Formas de comunicación dentro de una coreografía

Una vez que se han representado los elementos que intervendrán en la coreografía, es necesario definir cómo se comunicarán y relacionarán los diferentes servicios según sus roles. Existe toda una serie de posibles variantes para el esquema comunicación; en nuestro caso, éstas se pueden resumir de modo general en cuatro posibles patrones arquitectónicos, que identifican cuatro formas distintas de implementar la interacción entre los servicios. Así, puede decirse que la comunicación utiliza un *monitor*, paso de mensajes de tipo multidifusión (*broadcast*), un esquema de pizarra como estructura de memoria compartida, o bien mediante comunicación directa.

En el caso de utilizar un esquema basado en un *monitor*, cada vez que algún servicio ejecuta alguna operación, será el monitor el que detecte el cambio de rol que se produce o el que perciba el mensaje de invocación dirigido al servicio que cambia de rol. En este entorno, el monitor es un servicio especial que conoce el rol actual de cada uno de los participantes en la coreografía y cuál es la última operación que se ha ejecutado (y quién la ha ejecutado, por supuesto). Periódicamente, los participantes de la coreografía preguntan al monitor por la situación de la misma; y, en función de esta información, pueden actuar en consecuencia. Como se ha comentado, esta opción necesita de la existencia de un *servicio de soporte* que se sitúa como elemento central de la coreografía y que conoce en todo momento cuál es el estado global del sistema de coordinación. El concepto de coreografía en este caso está desvirtuado por la existencia de este elemento, que establecerá contratos con todos los servicios involucrados. Debido a esta circunstancia, la coreografía se transformaría en una orquestación.

En el caso de utilizar comunicaciones de tipo multidifusión (*broadcast*), cada vez que un servicio ejecuta una operación debe notificarlo al resto de participantes de la coreografía, a través de la invocación de una operación especial definida para todos los servicios. En este entorno se mandarían tantos mensajes como roles contenga la coreografía, de tal forma que todos estén conectados con todos. Esta solución no encaja bien con el concepto de comunicación punto a punto de los servicios pero sí con el hecho de que exista una estructura común que todos los servicios utilizan para comunicarse, es decir, nos enfrentaríamos a una integración basada en el concepto de ESB (*Enterprise Service Bus*) [14].

La tercera opción puede considerarse como una mezcla de las dos anteriores. La utilización de una *pizarra* o estructura compartida por todos los servicios en la que se reflejan, tanto las comunicaciones que se suceden entre dos servicios, como el rol de cada uno de los servicios en cada momento. En un entorno como el de la orientación a servicios, en el que cualquier elemento computacional o recurso se entiende como un servicio, la pizarra también se entendería como un servicio más. En tal caso, volveríamos a enfrentarnos al diseño de estrategias de coordinación basadas en orquestaciones.

Finalmente, existe la posibilidad de reflejar, en el modelo arquitectónico, las comunicaciones directas que se producen entre los distintos roles, en el que cada uno de ellos notifica cada una de las operaciones realizadas a aquellos servicios interesados (suscriptores) en conocer los cambios de roles que sucedan, de una manera análoga a la utilizada en el patrón *Observador*. Esta situación se modelará mediante diagramas de secuencia en los que se representan los roles involucrados y los mensajes intercambiados (invocación de operaciones). Esta opción es, por tanto, la única que describe una verdadera coreografía, y es la que se considera en este trabajo.

3.4 Modificación de los roles de los servicios dentro de una coreografía

Como se ha comentado en apartados anteriores, la situación de juego se modela mediante un servicio compuesto de tipo coreografía. La funcionalidad de este servicio dependerá de la táctica representada como interacción entre roles. La asignación de roles a servicios se modela mediante un *CollaborationUse* de UML 2.0 (no mostrado aquí por motivos de espacio). Cada uno de los servicios irá cambiando de rol a medida que la táctica evolucione, es decir, a medida que los

servicios interactúen entre sí. Esta interacción se produce, en entornos como el del baloncesto, por el cambio de dueño del balón en cada instante del partido.

Cada una de las interacciones se producirá por la ejecución del proceso definido en cada rol que conllevará, primero, la invocación de las operaciones de otro rol (por ejemplo, el atacante sin balón solicitará/invocará el *pase de balón* del servicio con el rol de atacante con balón) y, segundo, que como consecuencia de esta acción, los servicios cambien de rol.

El cambio de rol no implica la creación de nuevos contratos de servicio puesto que ya se contemplan todas las posibles combinaciones entre servicios cuando se modela el servicio compuesto. Lo único que se modificará será el conjunto de acciones que podrá realizar un servicio debido a su cambio de rol y por lo tanto el diagrama que representa el *CollaborationUse*.

4 Trabajos Relacionados

En esta sección se analizan algunas de las propuestas encontradas en la bibliografía acerca del modelado de coreografías. La gran mayoría de ellas se centran en el desarrollo de soluciones basadas en tecnologías, lenguajes y estándares concretos de coordinación de servicios. Así, podemos encontrar propuestas orientadas al desarrollo de coreografías con información semántica como la de Paolucci et al. , centrada en el lenguaje DAML-S; otras encaminadas a diseñar coreografías de procesos de negocio basadas en la notación BPMN mediante el lenguaje BPEL como ocurre en [8] con BPEL4Cor y otras basadas en los lenguajes específicos para coreografías WS-Coordination y WS-CF [9] o WS-CAF [20].

Con respecto a aquellas propuestas que utilizan un enfoque de más alto nivel, encontramos propuestas que utilizan notaciones particulares como la de Barros et al. [4], que abordan el modelado de coreografías desde diferentes vistas de la arquitectura e incluso iniciativas de representación de coreografías mediante autómatas como la de Mitra et al. [18].

De aquellas propuestas que utilizan un enfoque dirigido por modelos, podemos destacar las propuestas de Gonczy et al. [10] que proponen un metamodelo en el que se incluye el concepto de composición de servicios pero no el de coreografía u orquestación. Heckel et al. [11] y Baresi et al. [3] parten de la propuesta anterior para diseñar una alternativa basada en la propuesta de MDA pero sin que, de nuevo, se tenga en cuenta la representación de coreografías. Esta situación se vuelve a repetir en las propuestas de Autili et al. [2] y Colombo et al. [6].

Muchas metodologías de desarrollo únicamente describen la colaboración o composición entre servicios como orquestaciones. Entre ellas podemos mencionar la iniciativa de IBM [13] que propone un perfil UML para la representación de arquitecturas de servicios. Este enfoque define el concepto de colaboración entre servicios como forma de representar la implementación de coordinación entre servicios, pero mediante BPEL4WS [1].

Otras propuestas de integración de aplicaciones orientadas a servicios utilizan, como núcleo de la coordinación, diferentes plataformas de envío centradas en el paso de mensajes entre servicios. Este tipo de estrategia de integración hace que, por un lado, se dependa de la tecnología de implementación de la plataforma; y, por otro lado, que la plataforma en sí se considere como un elemento director de la coordinación. La existencia de este elemento restringe los grados de libertad que se le puede otorgar a los servicios que participan en la coordinación. En estos casos hablamos principalmente de orquestaciones más que de coreografías. Entre las propuestas que utilizan este enfoque podemos destacar la de Wada et al. [32] que modelan las coreografías siguiendo una arquitectura de filtros y tuberías.

De entre las pocas propuestas que contemplan el concepto de rol dentro del modelado de la coordinación entre servicios, podemos mencionar las iniciativas de Krüger et al. [15] donde se utiliza el concepto de rol desde el punto de vista del paradigma de componentes, o la propuesta de Zhang et al. [34] que también se centra en la definición de roles como parte de los componentes de servicio. En ambos casos no se hace ninguna referencia al concepto de coreografía.

Como se puede comprobar, hasta donde nuestro conocimiento alcanza, no existe prácticamente ningún trabajo que englobe, en una misma propuesta, el desarrollo dirigido por modelos de un sistema orientado a servicios en el que se pueda representar la coordinación entre servicios. Tampoco existen propuestas que soporten, de forma simultánea el modelado de coreografías y orquestaciones como formas diferentes de coordinación.

5 Conclusiones y Líneas de Investigación Abiertas

En este trabajo se ha presentado una propuesta de modelado, a nivel arquitectónico, de coreografías de servicios como estrategia de integración de aplicaciones.

Se ha partido de una estrategia de coordinación real que, utilizada como metáfora válida para la simulación de coordinación de servicios, demuestra que resulta inviable y poco práctico, mediante las técnicas de implementación a bajo nivel actuales, representar en toda su envergadura un sistema táctico complejo. Por ello, se ha propuesto y explicado cómo un enfoque dirigido por modelos puede resolver el problema de la representación de soluciones de integración basadas en servicios. Además, este enfoque se engloba dentro de una metodología de desarrollo más genérica que sigue los principios de la propuesta de MDA, con lo que el diseño de la solución orientada a servicios se desarrolla a través de modelos separados en distintos niveles de abstracción, con la ventaja de poder definir reglas de transformación automáticas entre cada uno de estos modelos. Este último aspecto es objeto de una investigación que actualmente está en progreso y que, por motivos de espacio, no ha sido presentada en este trabajo.

La utilización de una metáfora deportiva nos sirve además para profundizar en la forma en que debe ser definida una arquitectura orientada a servicios. El objetivo es que al tratar los servicios como entidades fundamentales de construcción de soluciones software compuestas, además de las relaciones establecidas entre ellos, resulta posible identificar los elementos que han de definirse en el resto de los modelos, dentro de una metodología orientada a servicios como SOD-M.

Cabe remarcar que en este artículo no se ha explicado en detalle (ni ejemplificado) el modelado del intercambio de mensajes que sucede como consecuencia de la coreografía y su aplicación al bloqueo directo como estrategia de juego. La descripción detallada de los diagramas de secuencia y colaboración, así como el proceso de evolución de los roles de cada servicio, se ha dejado, por motivos fundamentalmente de espacio, para otros trabajos.

6 Agradecimientos

Este trabajo se ha llevado a cabo dentro de los proyectos GOLD (TIN2005-00010) financiado por el Ministerio de Ciencia y Tecnología y el proyecto IASOMM (URJC-CM-2007-CET-1555) cofinanciado por la Universidad Rey Juan Carlos y la Comunidad Autónoma de Madrid, y el proyecto *Agreement Technologies* (CONSOLIDER CSD2007-0022, INGENIO 2010).

Bibliografía

- [1] Andrews T., Curbera F., Dholakia H. et al. *Business Process Execution Language for Web Services, Version 1.1 Specification*. BEA Systems, IBM, Microsoft Corp., SAP AG, Siebel Systems, (2003).
- [2] Autili M., Cortellessa V., Di Marco M. and Inverardi P. A Conceptual Model for Adaptable Context-aware Services. In *Proc. of Web Services Modeling and Testing*, Palermo, Sicily, Italy, June 2006
- [3] Baresi L., Heckel R., Thone S., and Varro D. Modeling and validation of service-oriented architectures: Application vs. style. In *Proc. ESEC/FSE 2003*, Helsinki, Finland, September 2003
- [4] Barros, A., Decker, G., Dumas, M. Multi-staged and Multi-viewpoint Service Choreography Modelling. *Technical Report 4668*, Queensland University of Technology, Brisbane, Australia, 2006.

- [5] BPMI Notation Working Group. *Business Process Modeling Notation (BPMN) Version 1.0*. Disponible en <http://www.bpmi.org/>
- [6] Colombo M., Di Nitto E., Di Penta M. et al. Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. In *ICSOC'05*, Amsterdam, the Netherlands, December 2005.
- [7] De Castro V., E. Marcos, M. López Sanz. A Model Driven Method for Service Composition Modeling: A Case Study. *Intl Journal of Web Engineering and Technology*. Vol. 2, No. 2, Pp.: 335-353, 2006.
- [8] Decker G., Kopp O, Leymann F., Pfitzner K., Weske M: Modeling Service Choreographies Using BPMN and BPEL4Chor. *CAiSE 2008*: 79-93
- [9] Fujitsu Arjuna. *WS-CF: WS-Coordination Framework*, <http://developers.sun.com/techtopics/webservices/wscf/wscf.pdf>, Oracle IONA, Sun. 2003
- [10] Gönczy L., and Varró D. Modeling of Reliable Messaging in Service Oriented Architectures, In Proc. Intern. *Workshop on Web Service Modeling and Testing (WS-MATE 2006)*
- [11] Heckel R., Lohmann M., Thöne S.: Towards a UML Profile for Service-Oriented Architectures, *Workshop on MDA: Foundations and Applications (MDAFA'03)*. Enschede, June 2003.
- [12] Jerome Josephraj. Web services choreography in practice. *IBM DeveloperWorks Site*, mayo 2005. Obtenido de <http://www.ibm.com/developerworks/xml/library/ws-choreography/index.html>
- [13] Johnston S. UML profile for software services. *IBM DeveloperWorks Site*, April 2005. Obtenido de: http://www-128.ibm.com/developerworks/rational/library/05/419_soa/
- [14] Krafzig, D.; Banke K., Slama D. *Enterprise SOA Service Oriented Architecture Best Practices*. Ed. Upper Saddle River: Prentice Hall PTR, 2004.
- [15] Krüger I. H., Mathew R.: Systematic Development and Exploration of Service-Oriented Software Architectures. In *Proc. of WICSA'04*, pp. 177-187. Oslo, Norway. IEEE/IFIP, 2004.
- [16] López-Sanz, M., Acuña, C., Cuesta, C. Marcos, E. Defining Service-Oriented Software Architecture Models for a MDA-based Development Process at the PIM level. *Proc. of WICSA'08*, IEEE/IFIP, 2008.
- [17] Marcos E., Acuña C. J., Cuesta C. E. Integrating Software Architecture into a MDA Framework. In *Proc. of EWSA'06*, pp: 127-143. Nantes, France.
- [18] Mitra S., Kumar R. and Basu S. Automated Choreographer Synthesis for Web Services Composition Using I/O Automata. *IEEE International Conference on Web Services (ICWS) 2007*.
- [19] OASIS. *Reference Model for Service Oriented Architecture. Committee draft 1.0*. Obtenido de: <http://www.oasis-open.org/committees/download.php/16587/wd-soa-rm-cd1ED.pdf>
- [20] OASIS. *Web Services Composite Application Framework Technical Committee*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-caf
- [21] OASIS. *Web Services Coordination (WS-Coordination) Version 1.1. April 2007*. Obtenido de <http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.1-spec-os/wstx-wscoor-1.1-spec-os.html>
- [22] Ociepka, Bob. Defending the pick and roll. *FIBA assist magazine*. Nov. – Dec. 2004, 31 - 34.
- [23] OMG. Model Driven Architecture. Eds.: Miller, J., Mukerji, J. Document No. ormsc/2001-07-01. Obtenido de: <http://www.omg.com/mda>
- [24] OMG. *UML 2.0 Superstructure Specification*, <http://www.omg.org/docs/ptc/03-08-02.pdf>. August 2003
- [25] Paolucci M., Srinivasan N., Sycara K., and Nishimura T. Toward a Semantic Choreography of Web Services: From WSDL to DAML-S. In *Proc. of ICWS'03*. Las Vegas, Nevada, USA, 2003, pp 22-26.
- [26] Papazoglou M. P. Service-Oriented Computing: Concepts, Characteristics and Directions. In *Proc. of WISE'03*. pp. 3-12. Roma, Italy, December 10-12, 2003.
- [27] Papazoglou M. P. What's in a Service? In *Proc. of ECSA'07*: pp.11-28
- [28] Peltz C. Web Services Orchestration and Choreography. *Computer*, vol.36 [10], pp. 46-52, Oct., 2003
- [29] Remmert, H. Analysis of group-tactical offensive behavior in elite basketball on the basis of a process orientated model. *European Journal of Sport Science*, vol 3, issue 3, June 2003, pp. 1-12(11)
- [30] Romay, P., Cuesta, C., López-Sanz, M., Orientación a Aspectos en UML2 sin Extensiones. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(1), 23-49, abril 2008.
- [31] W3C. *Web Services Choreography Description Language Version 1.0. W3C Working Draft 17 December 2004*. Obtenido de <http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/>
- [32] Wada H., Suzuki J. and Oba K., Modeling Non-Functional Aspects in Service Oriented Architecture. In *Proc. of the 2006 IEEE International Conference on Service Computing*, Chicago, IL, September 2006
- [33] Wissel, H. *Baloncesto, aprender y progresar*. Ed. Paidotribo. Barcelona, 1996.
- [34] Zhang T., Ying S., Cao S., Jia X. A Modeling Framework for Service-Oriented Architecture. *Proceedings of the Sixth International Conference on Quality Software (QSIC 2006)*, pp. 219-226.
- [35] Zimmermann, O., Doubrovski, V., Grundler, J., and Hogg, K. Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned. In *Companion To the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*. OOPSLA '05. ACM, New York, NY, 301-312.