(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 315 970**

(51) Int. Cl.:
H04L 29/08 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **06015020 .8**
(96) Fecha de presentación : **19.07.2006**
(97) Número de publicación de la solicitud: **1881675**
(97) Fecha de publicación de la solicitud: **23.01.2008**

(54) Título: **Sistema y procedimiento para administrar una pluralidad de servicios web.**

(45) Fecha de publicación de la mención BOPI:
01.04.2009

(45) Fecha de la publicación del folleto de la patente:
01.04.2009

(73) Titular/es: **SOFTWARE AG.**
Uhlandstrasse 12
64297 Darmstadt, DE

(72) Inventor/es: **Schöning, Harald;**
Viegener, Johannes y
Swenson, Keith

(74) Agente: **Miazzetto, Fabrizio**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

Venta de fascículos: Oficina Española de Patentes y Marcas. Pº de la Castellana, 75 – 28071 Madrid

DESCRIPCIÓN

Sistema y procedimiento para administrar una pluralidad de servicios web.

1. Campo técnico

La presente invención se refiere a un sistema y a un procedimiento para administrar una pluralidad de servicios web de una arquitectura de software orientada al servicio.

2. La técnica anterior

Los servicios web son interfaces estandarizadas para intercambiar datos de un ordenador a otro a través de una red informática, por ejemplo, Internet. Una solicitud puede requerir datos de un servidor a través de la red informática utilizando un servicio web provisto por el servidor. Una petición de este tipo se incluye típicamente en un sobre SOAP ("*SOAP envelope*") que se transporta utilizando HTTP y puede comprender XML junto con otros estándares relacionados con la web. Aplicaciones más avanzadas desde el punto de vista técnico combinan varios servicios web para obtener o procesar los datos deseados. Una arquitectura de software, que se desarrolla alrededor de una pluralidad de servicios web, se llama arquitectura orientada al servicio (SOA).

En un SOA, los recursos SOA se ponen a la disposición de los participantes en la red como servicios independientes a los que los participantes pueden acceder de una manera estandarizada. Mientras que muchas definiciones de SOA utilizan peticiones SOAP transportadas vía HTTP/HTTPS a través de TCP/IP, un SOA puede utilizar cualquier tecnología basada en servicio web. Los servicios web interactúan en base a definiciones formales que son independientes de la plataforma subyacente y del lenguaje de programación. La definición de interfaz engloba las implementaciones específicas. Un SOA es independiente de una tecnología de desarrollo específica (como el Java y el NET). Los componentes de software se vuelven muy reutilizables ya que la interfaz cumple con los estándares y es independiente de la implementación subyacente de la lógica del servicio web. Por ejemplo, una solicitud JAVA podría utilizar un servicio C# (C Sharp) y viceversa.

El desarrollador de una nueva solicitud en un SOA necesita acceso a los diversos servicios web que la nueva solicitud va a utilizar. Esto no sólo se refiere a la disponibilidad de las especificaciones de la interfaz de los servicios web implicados, sino también a la provisión de servicios web realmente activos a los que se puede llamar para probar la nueva solicitud. Sin embargo, dichas llamadas de prueba no deben interferir con el mundo real. Por ejemplo, llamar a un servicio web que provoque una transferencia de dinero a modo de prueba no debe provocar una transferencia de dinero real.

Se aplica un escenario similar tras el desarrollo, cuando la nueva solicitud debe probarse en un entorno mayor antes de ser aprobada para su uso en producción. De nuevo, deben crearse nuevos casos de prueba de servicios web llamados por la nueva solicitud, a causa de los diferentes requisitos relacionados con la estabilidad o la generación de resultado durante la fase de desarrollo y la fase de prueba en un entorno mayor. Cuando la solicitud queda finalmente aprobada para producción, las llamadas a los diversos servicios web de prueba deben ser reemplazadas por llamadas a los servicios reales. Las diversas etapas de reemplazo que intercambian los servicios web para el desarrollo, prueba a gran escala y, finalmente, producción presentan un riesgo substancial por introducir errores en la solicitud final.

En la Patente WO 2005/029807 A1 se presenta un procedimiento y un sistema para procesar información de contexto en solicitudes basadas en la web para proporcionar Servicios Web mediante peticiones respectivas. Se describe para incluir información de contexto en la especificación de punto final ("*endpoint*") de una petición de servicio con el fin de simplificar la comunicación entre el cliente y el proveedor del servicio sin los cambios necesarios en las interfaces de servicio web existentes.

En el documento "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language" de Chinnici *et al*. se define un lenguaje XML para describir servicios Web basados en un modelo abstracto de lo que el servicio ofrece. Además, el documento proporciona criterios de conformidad para documentos en este lenguaje.

La Patente Europea 1 641 215 A2 presenta un sistema y un procedimiento para enlazar identidades en un SOA. Un enlace asociado con un primer dominio de seguridad sirve para interceptar mensajes de un usuario de este dominio para un servicio de un segundo dominio de seguridad. El enlace autentifica las identidades del usuario contra una fuente local de autentificación. El enlace conlleva una credencial de seguridad con el mensaje cuya autenticidad se verifica mediante una pasarela asociada con el segundo dominio cuando se recibe el mensaje autentificado.

Es pues el problema técnico subyacente de la presente invención proporcionar un sistema y un procedimiento que superen las desventajas arriba mencionadas de la técnica anterior y que, en particular, faciliten el cambio desde las diferentes etapas de desarrollo y prueba hasta la difusión misma de una nueva solicitud que involucre a uno o varios servicios web sin el riesgo de introducir errores.

3. Resumen de la invención

En un aspecto de la invención, este problema se soluciona mediante un sistema según la reivindicación 1.

Por consiguiente, servicios web para marcos de ejecución diferentes (por ejemplo, desarrollo, prueba, producción) pueden manejarse dentro de un mismo registro sin perder el control sobre los servicios de los marcos y el riesgo implicado de introducir errores. En la difusión, los puntos finales correctos pueden escogerse automáticamente según la condición de ejecución respectiva. Como resultado, se pueden generar automáticamente diversos entornos de prueba y de producción de una nueva solicitud.

En un modo de realización, el sistema está adaptado además para seleccionar uno de los primer y segundo puntos finales durante la ejecución dependiendo de una evaluación de la información del entorno relacionada con la condición de ejecución provista por una solicitud que llama al servicio web. Establecer el valor de una sola variable en la solicitud permite pues determinar qué marco se utiliza, ya que la o las descripciones de los servicios web en el registro incluyen automáticamente puntos finales adaptados a la condición de ejecución respectiva (por ejemplo, prueba, difusión, etc.).

En otro modo de realización, el sistema está adaptado además para seleccionar uno de los primer y segundo puntos finales durante la ejecución dependiendo de la condición de ejecución almacenada en el registro. El registro puede, por ejemplo, almacenar información que refleje el estado actual del proceso de desarrollo y, por consiguiente, selecciona el marco adecuado de los servicios web para la nueva solicitud.

En un modo de realización, el primer y el segundo puntos finales se definen en un primer y un segundo documentos WSDL para el servicio web almacenado en el registro, que están relacionados entre sí, por ejemplo, haciendo referencia el uno al otro, donde cada uno de los documentos WSDL puede estar marcado por su condición de ejecución especifica.

Sin embargo, en el presente modo de realización preferente, el primer y el segundo puntos finales están definidos en un único documento WSDL para el servicio web, preferiblemente como un componente característico del documento WSDL. Como resultado, el registro almacena descripciones WSDL de servicios web “gemelos”, en las que uno de los gemelos se utiliza para una primera condición de ejecución y el otro para otra condición de ejecución. El valor del componente característico se determina, en un modo de realización, durante la ejecución utilizando un módulo SOAP de una petición para el servicio web.

En el presente modo de realización preferente, el primer punto final está adaptado para ser utilizado para probar una solicitud que llama al, al menos un, servicio web, mientras que el segundo punto final está preferiblemente adaptado para ser utilizado en la difusión de una solicitud que llama al, al menos un, servicio web. Sin embargo, el concepto de la presente invención puede utilizarse también para escoger entre otros marcos, es decir, para otras condiciones de ejecución.

El sistema comprende además medios para monitorizar los puntos finales de la pluralidad de servicios web durante la ejecución y medios para emitir un aviso si se llama a un primer servicio web con un punto final para una primera condición de ejecución y si se llama a un segundo servicio web con un punto final para una segunda condición de ejecución. La función de monitorización evita inconsistencias que pueden causar errores en la ejecución de la nueva solicitud.

Según otro aspecto, la presente invención se refiere a un procedimiento de acuerdo con la reivindicación 11.

Modificaciones ventajosas adicionales de los modos de realización del sistema y del procedimiento de la invención se definen en las reivindicaciones dependientes adicionales.

4. Breve descripción de los dibujos

En la siguiente descripción detallada, se describen con más detalle modos de realización actualmente preferidos de la invención con referencia a las siguientes figuras:

Fig. 1: Una vista general de un ejemplo de un SOA simplificado que comprende una solicitud que llama a un Servicio Web y un registro con una descripción del Servicio Web; y

Fig. 2: Una vista simplificada de un SOA de dos marcos con una pluralidad de servicios web gemelos; y

Fig. 3: Un organigrama que ilustra las etapas llevadas a cabo durante la ejecución en un sistema y procedimiento según un modo de realización de la invención.

5. Descripción detallada de los modos de realización preferentes

La Fig. 1 presenta una visión global de un SOA extremadamente simplificado: Una solicitud 30 emite una petición 31 para un servicio web 20. Con este fin, la solicitud 30 necesita primeramente una descripción de las definiciones de interfaz del servicio web 20. Esta descripción 40, que puede ser suministrada, por ejemplo, en el Lenguaje de Definición del Servicio Web (WSDL del inglés "*Web Service Definition Language*"), se puede obtener tanto directamente del servicio web 20 como a través de un registro 10 (cf. Fig. 1).

El registro 10 comprende preferiblemente descripciones de todos los servicios web del SOA, por ejemplo archivos WSDL 40 con información sobre la función, los datos, el tipo de dato y los protocolos de intercambio del servicio web respectivo. Por consiguiente, si la solicitud 30 intenta enviar una petición al servicio web 20, puede obtener el archivo WSDL 40 desde el registro 10 para descubrir como acceder al servicio web 20.

Sin embargo, cuando la nueva solicitud 30 se está desarrollando, es necesario contactar repetidamente con el servicio web implicado 20, ya con fines de prueba, por ejemplo para verificar si se cumplen todos los requisitos de la interfaz en la llamada al servidor web y si la solicitud 30 recibe y procesa correctamente la respuesta del servicio web. Si el servicio web 20 está realmente en uso, cada una de estas llamadas de prueba causará algunas operaciones en el mundo real (por ejemplo, la reserva de un billete de avión o la transferencia de dinero de una cuenta a otra). En este contexto, hay que notar que el servicio web 20 puede estar en algún lugar en internet y que su funcionamiento puede no estar necesariamente bajo el control del desarrollador de la nueva solicitud 30.

Por consiguiente, la presente invención proporciona, de acuerdo con un modo de realización preferido, servicios "gemelos" como se muestra de forma esquemática en la Fig. 2: Como se puede ver, el registro 10 almacena preferentemente descripciones únicas WSDL1-WSDL3 de servicios web correspondientes con, sin embargo, múltiples puntos finales cada uno de los cuales lleva una referencia a marco de ejecución 50 o 50' al que pertenece. El término "marco" se refiere a una parte específica del SOA 100 en su conjunto que puede distinguirse mediante algunos tipos de atributos designados. En el modo de realización simplificado que se muestra en la Fig. 2 hay un marco de prueba separado 50 y un marco de producción separado 50'. Ambos marcos 50, 50' proporcionan, al menos en parte, los mismos servicios web WS1-WS3 y WS1'-WS3', pero sólo los servicios web de producción WS1'-WS3' funcionan en la vida real y causan, por ejemplo, transacciones comerciales. Las versiones de prueba de servicios productivos (por ejemplo, los proporcionados por alguna 3ª parte) se pueden generar y añadir automáticamente al registro 10.

También se pueden utilizar otros tipos de distinciones de marcos, por ejemplo un marco de desarrollo contra un marco de prueba mayor, o el marco de una empresa contra el marco de otra empresa. Además, puede haber no sólo dos sino tres o más marcos.

Para manejar sin errores la familia de servicios gemelos arriba descrita, se definen en el registro único 10 los puntos finales de los dos marcos 50, 50' en los que existen los servicios web WS1-WS3 y WS1'-WS3',. Así, las descripciones de la interfaz de servicio web WSDL1-WSDL3 se comparten entre los marcos 50, 50', mientras que los puntos finales no.

Sin embargo, hay que notar que, en vez de utilizar descripciones WSDL únicas WSDL1-WSDL3 para los servicios web WS1-WS3 y WS1'-WS3', puede haber también como alternativa dos conjuntos de descripciones relacionadas WSDL1-WSDL3 y WSDL1'-WSDL3' almacenadas en el registro (no mostrado en la Fig. 2), donde las dos descripciones WSDL que pertenecen a un servicio gemelo (por ejemplo WSDL1 y WSDL1') están relacionadas entre sí. Dependiendo de la condición de ejecución, se utiliza una descripción o la otra (o posiblemente una tercera).

En el presente modo de realización preferente, los puntos finales de los dos marcos 50, 50' se definen en un solo archivo WSDL para un servicio gemelo que utiliza el mecanismo de extensibilidad de "características" provisto en el WSDL 2.0. Una "características" representa un llamado valor de ejecución que afecta al comportamiento de un aspecto de una interacción de servicio web. Por ejemplo, un modulo SOAP de una petición puede especificar una características para controlar los puntos finales. Esto se ilustra en el siguiente ejemplo de archivo WSDL que describe servicios gemelos:

```
<?xml version="1.0" encoding="UTF-8"?>
<description
 xmlns="http://www.w3.org/2006/01/wsdl"

targetNamespace="http://namespaces.softwareag.com/2004/wsdl/resSvc
"
 xmlns:tns="http://namespaces.softwareag.com/2004/wsdl/resSvc"
 xmlns:ghns="http://namespaces.softwareag.com/2004/schemas/resSvc"
 xmlns:wsoap="http://www.w3.org/2006/01/wsdl/soap"
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
```

```
 xmlns:wsdlx="http://www.w3.org/2006/01/wsdl-extensions"
>
 <documentation>
    This document describes the reservation Web service.
 </documentation>
 <types>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://namespaces.softwareag.com/2004/schemas/res
Svc" xmlns="http://namespaces.softwareag.com/2004/schemas/resSvc">
    <xs:element name="checkAvailability"
type="tCheckAvailability"/>
    <xs:complexType name="tCheckAvailability">
      <xs:sequence>
       <xs:element name="checkInDate" type="xs:date"/>
       <xs:element name="checkOutDate" type="xs:date"/>
       <xs:element name="roomType" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="checkAvailabilityResponse"
type="xs:double"/>
    <xs:element name="invalidDataError" type="xs:string"/>
  </xs:schema>
 </types>
 <interface name="reservationInterface">
  <fault name="invalidDataFault"
element="ghns:invalidDataError"/>
  <operation name="opCheckAvailability"
     pattern="http://www.w3.org/2006/01/wsdl/in-out"
     style="http://www.w3.org/2006/01/wsdl/style/iri"
     wsdlx:safe="true">
    <input messageLabel="In" element="ghns:checkAvailability"/>
    <output messageLabel="Out"
       element="ghns:checkAvailabilityResponse"/>
    <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
  </operation>
 </interface>
 <binding name="reservationSOAPBinding"
  interface="tns:reservationInterface"
  type="http://www.w3.org/2006/01/wsdl/soap"
  wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">
  <fault ref="tns:invalidDataFault" wsoap:code="soap:Sender"/>
  <operation ref="tns:opCheckAvailability"
     wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-
response"/>
 </binding>
 <service name="reservationService"
           interface="tns:reservationInterface">
  <endpoint name="reservationEndpointTest"
       binding="tns:reservationSOAPBinding"
       address="http://test.softwareag.com/2004/reservation">
    <property
ref="http://features.softwareag.com/landscapes/type">
      <documentation>
        This endpoint belongs to the test landscape
      </documentation>
      <value>test</value>
```

```
      </property>
    </endpoint>
    <endpoint name="reservationEndpointProduction"
          binding="tns:reservationSOAPBinding"

   address="http://production.softwareag.com/2004/reservation">
      <property
  ref="http://features.softwareag.com/landscapes/type">
        <documentation>
         This endpoint belongs to the production landscape
        </documentation>
        <value>production</value>
      </property>
    </endpoint>
   </service>
  </description>
```

Como se puede ver, se definen dos puntos finales, uno para el marco de prueba, el otro para el marco de producción. El valor de ejecución del componente característico determina pues uno de dos posibles puntos finales y, por ello, si el servicio web del marco de prueba o del marco de producción serán tratados o no.

De manera alternativa, los puntos finales de los dos marcos se pueden especificar según el esquema UDDI-XML (UDDI 3.0) tal como se muestra a continuación:

```
<uddi:businessService
 businessKey="uddi:35f2eca0-eef2-11da-a924-c9108119a4e5"
 serviceKey="uddi:4fc7c540-ef08-11da-bf78-dc56dc1260cf"
 xmlns:uddi="urn:uddi-org:api_v3"
 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
 xmlns:scc14n="urn:uddi-org:schemaCentricC14N:2002-07-10">
 xmlns:xml="http://www.w3.org/XML/1998/namespace"
<uddi:name xml:lang="en-US">
   reservationService
 </uddi:name>
 <uddi:description
     xml:lang="en">
      Service for reservation available in test and production
landscape
 </uddi:description>
 <uddi:bindingTemplates>
  <uddi:bindingTemplate
   bindingKey="uddi:50357b30-ef08-11da-bf78-80c376f5f0f0"
   serviceKey="uddi:4fc7c540-ef08-11da-bf78-dc56dc1260cf">
   <uddi:accessPoint useType="endPoint">
     http://test.softwareag.com/2004/reservation
   </uddi:accessPoint>
   <uddi:tModelInstanceDetails>
```

```
      <uddi:tModelInstanceInfo
        tModelKey="uddi:4fb24170-ef08-11da-bf78-8f16f66201d4">
       <uddi:instanceDetails xmlns:uddi="urn:uddi-org:api_v3">
        <uddi:description xml:lang="en-US">
            the wsdl:binding that this wsdl:port implements
        </uddi:description>
        <uddi:instanceParms>
                  reservationSOAPBinding
        </uddi:instanceParms>
       </uddi:instanceDetails>
      </uddi:tModelInstanceInfo>
      <uddi:tModelInstanceInfo
       tModelKey="uddi:4f8516f0-ef08-11da-bf78-c304af4ea516"/>
     </uddi:tModelInstanceDetails>
     <uddi:categoryBag>
     <uddi:keyedReference
        tModelKey="uddi:features.softwareag.com:landscapes"
        keyValue="test"/>
     </uddi:categoryBag>
    </uddi:bindingTemplate>
    <uddi:bindingTemplate
     bindingKey="uddi:50357b30-ef08-11da-bf78-80c376f5f0f1"
     serviceKey="uddi:4fc7c540-ef08-11da-bf78-dc56dc1260cf">
     <uddi:accessPoint useType="endPoint">
       http://production.softwareag.com/2004/reservation
     </uddi:accessPoint>
     <uddi:tModelInstanceDetails>
      <uddi:tModelInstanceInfo
       tModelKey="uddi:4fb24170-ef08-11da-bf78-8f16f66201d4">
       <uddi:instanceDetails>
        <uddi:description xml:lang="en-US">
            the wsdl:binding that this wsdl:port implements
        </uddi:description>
        <uddi:instanceParms>
            reservationSOAPBinding
        </uddi:instanceParms>
       </uddi:instanceDetails>
      </uddi:tModelInstanceInfo>
      <uddi:tModelInstanceInfo
       tModelKey="uddi:4f8516f0-ef08-11da-bf78-c304af4ea516"/>
     </uddi:tModelInstanceDetails>
     <uddi:categoryBag>
      <uddi:keyedReference
        tModelKey="uddi:features.softwareag.com:landscapes"
        keyValue="production"/>
     </uddi:categoryBag>
    </uddi:bindingTemplate>
   </uddi:bindingTemplates>
   <uddi:categoryBag>
    <uddi:keyedReference
       keyName="service namespace"
       keyValue="http://namespaces.softwareag.com/2004/wsdl/resSvc"
       tModelKey="uddi:uddi.org:xml:namespace"/>
    <uddi:keyedReference
       keyName="WSDL type"
```

```
        keyValue="service"
        tModelKey="uddi:uddi.org:wsdl:types"/>
    <uddi:keyedReference
        keyName="service local name"
        keyValue="reservationService"
        tModelKey="uddi:uddi.org:xml:localname"/>
    <uddi:keyedReference
        keyValue="uddi:2dd13990-8991-11d9-8457-ca59ad32ca75"
        tModelKey="uddi:2dafcee0-8991-11d9-8457-f2038ed2fcd0"/>
  </uddi:categoryBag>
</uddi:businessService>
```

Una petición UDDI para encontrar el punto final de un marco dado (por ejemplo, de prueba) puede parecerse pues a lo siguiente:

```
<find_binding xmlns="urn:uddi-org:api_v3"
     servicioKey="uddi:4fc7c540-ef08-11da-bf78-
dc56dc1260cf">
 <findQualifiers>
  <findQualifier>exactNameMatch</findQualifier>
 </findQualifiers>
 <categoryBag>
  <keyedReference
    tModelKey="uddi:features.softwareag.com:marcos"
    keyValue="test"/>
 </categoryBag>
</find_binding>
```

La información de punto final que aparece así en el archivo WSDL o el registro (UDDI) se puede utilizar en varios escenarios, uno de los cuales se describe a continuación con referencia al organigrama de la Fig. 3 que ilustra tres componentes implicados:

Un componente de ejecución de servicio web 70 está disponible en ejecución (por ejemplo para llevar a cabo una prueba) para una solicitud 30. La solicitud 30 llama al componente de ejecución de servicio web 70 en la etapa 200, cuando se tiene que ejecutar un servicio web 20, y recibe una única identificación del servicio web 20 como entrada (por ejemplo, su clave UDDI, o el URL de su archivo WSDL). En la etapa 201, el componente de ejecución de servicio web 70 prepara la ejecución de la llamada. En la etapa 202, accede a la información almacenada en el registro 10 para determinar en la etapa 204 el punto final para el servicio web al que hay que llamar, por ejemplo en base al archivo WSDL obtenido del registro 10. Hay que notar que el punto final que resulta de la etapa 204 puede depender de un parámetro de entrada proporcionado por la solicitud 30 y contenido en su llamada como se explica más arriba, por ejemplo un cierto valor característico con el fin de definir un cierto entorno para la ejecución de la llamada de servicio web. Si no hay ningún servicio web disponible en el punto final indicado, se indica un error. De lo contrario, el componente de ejecución de servicio web 70 llama al servicio web identificado en la etapa 205, que se ejecuta en la etapa 206 para que los resultados puedan devolverse a la solicitud 30 en la etapa 207.

La gestión descrita de los puntos finales en el registro permite además, en el tiempo de difusión de la solicitud probada 30, identificar todos los puntos finales de servicio utilizados previamente para probar esa solicitud y reemplazarlos permanentemente por los puntos finales ya registrados en el registro 10 para el marco de producción. Además, las llamadas de servicio web pueden ser monitorizadas en la ejecución y verificadas si ocurre un cruce de marcos, es decir, si se llama a un servicio web que pertenece a un primer marco, tal como el marco de prueba 50, y se llama a otro servicio web 50' que pertenece a otro marco como el marco de producción 50'. Si se detecta tal inconsistencia durante la ejecución, se emite una alerta.

Generalmente, la identificación del marco de la persona que llama puede utilizar múltiples mecanismos, por ejemplo, una separación de los marcos de prueba y producción en base a dominios DNS. Se puede enviar la información de marco junto con la llamada de servicio web. El siguiente ejemplo de un archivo WSDL almacenado en el registro 10 requiere tal información y asume que hay un módulo SOAP específico para la gestión de la información de marco:

```
<?xml version="1.0" encoding="UTF-8"?>
<description
 xmlns="http://www.w3.org/2006/01/wsdl"
targetNamespace="http://namespaces.softwareag.com/2004/wsdl/resSvc
"
 xmlns:tns="http://namespaces.softwareag.com/2004/wsdl/resSvc"
 xmlns:ghns="http://namespaces.softwareag.com/2004/schemas/resSvc"
 xmlns:wsoap="http://www.w3.org/2006/01/wsdl/soap"
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsdlx="http://www.w3.org/2006/01/wsdl-extensions"
>
 <documentation>
    This document describes the reservation Web service.
 </documentation>
 <types>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://namespaces.softwareag.com/2004/schemas/res
Svc" xmlns="http://namespaces.softwareag.com/2004/schemas/resSvc">
    <xs:element name="checkAvailability"
type="tCheckAvailability"/>
    <xs:complexType name="tCheckAvailability">
      <xs:sequence>
       <xs:element name="checkInDate" type="xs:date"/>
       <xs:element name="checkOutDate" type="xs:date"/>
       <xs:element name="roomType" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="checkAvailabilityResponse"
type="xs:double"/>
    <xs:element name="invalidDataError" type="xs:string"/>
  </xs:schema>
 </types>
 <interface name="reservationInterface">
  <fault name="invalidDataFault"
element="ghns:invalidDataError"/>
  <operation name="opCheckAvailability"
     pattern="http://www.w3.org/2006/01/wsdl/in-out"
     style="http://www.w3.org/2006/01/wsdl/style/iri"
     wsdlx:safe="true">
   <input messageLabel="In" element="ghns:checkAvailability"/>
   <output messageLabel="Out"
       element="ghns:checkAvailabilityResponse"/>
   <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
  </operation>
 </interface>
 <binding name="reservationSOAPBinding"
  interface="tns:reservationInterface"
  type="http://www.w3.org/2006/01/wsdl/soap"
  wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP">
  <fault ref="tns:invalidDataFault" wsoap:code="soap:Sender"/>
  <operation ref="tns:opCheckAvailability"
     wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-
response"/>
 </binding>
 <service name="reservationService"
            interface="tns:reservationInterface">
  <endpoint name="reservationEndpointTest"
       binding="tns:reservationSOAPBinding"
       address="http://test.softwareag.com/2004/reservation">
   <wsoap:module
     uri="http://features.example.com/2005/modules/landscapes"
     required="true"/>
   <property
ref="http://features.softwareag.com/landscapes/type">
```

```
        <documentation>
          This endpoint belongs to the test landscape
        </documentation>
        <value>test</value>
      </property>
    </endpoint>
    <endpoint name="reservationEndpointProduction"
          binding="tns:reservationSOAPBinding"
   address="http://production.softwareag.com/2004/reservation">
      <wsoap:module
        uri="http://features.example.com/2005/modules/landscapes"
        required="true"/>
      <property
  ref="http://features.softwareag.com/landscapes/type">
        <documentation>
          This endpoint belongs to the production landscape
        </documentation>
        <value>production</value>
      </property>
    </endpoint>
  </service>
</description>
```

De modo alternativo, el registro 10 puede almacenar información sobre la fase de desarrollo de una cierta solicitud y causar una selección adecuada de los puntos finales (por ejemplo, en base a la difusión registrada de la solicitud). Esto se puede hacer o bien para todo el marco o individualmente para cada servicio web.

REIVINDICACIONES

1. Un sistema para gestionar una pluralidad de servicios web comprendiendo un registro (10) incluyendo una descripción (WSDL1, WSDL2, WSDL3) para al menos uno de la pluralidad de servicios web, en el que la descripción (WSDL1, WSDL2, WSDL3) del al menos un servicio web comprende un primer punto final (WS1, WS2, WS3) para una primera condición de ejecución y un segundo punto final (WS1', WS2', WS3') para una segunda condición de ejecución, **caracterizado** porque

el sistema comprende además medios para monitorizar los puntos finales (WSJ, WS2, WS3, WS1; WS2; WS3') de la pluralidad de servicios web durante la ejecución y medios para emitir un aviso si se llama a un primer servicio web con un punto final para una primera condición de ejecución y si se llama a un segundo servicio web con un punto final para una segunda condición de ejecución.

2. El sistema de la reivindicación 1 adaptado además para escoger uno de los primer (WS1, WS2, WS3) y segundo puntos finales (WS1', WS2', WS3') durante la ejecución dependiendo de una evaluación de información del entorno sobre la condición de ejecución proporcionada por una solicitud (30) que llama al servicio web.

3. El sistema de la reivindicación 1 adaptado además para escoger uno de los primer (WS1, WS2, WS3) y segundo puntos finales (WS1', WS2', WS3') durante la ejecución dependiendo de la condición de ejecución almacenada en el registro (10).

4. El sistema de cualquiera de las reivindicaciones precedentes, en el que el primer (WS1, WS2, WS3) y el segundo punto final (WS1', WS2', WS3') se definen en un primer y un segundo documentos WSDL para el servicio web, que están relacionados entre sí.

5. El sistema de cualquiera de las reivindicaciones precedentes 1-3, en el que el primer (WS1, WS2, WS3) y el segundo punto final (WS1', WS2', WS3') se definen en un único documento WSDL (WSDL1, WSDL2, WSDL3) para el servicio web, como un componente característico del documento WSDL.

6. El sistema de la reivindicación 5, en el que el valor del componente característico se determina durante la ejecución utilizando un módulo SOAP de una solicitud para el servicio web.

7. El sistema de cualquiera de las reivindicaciones precedentes 1-6, en el que el registro (10) es un registro UDDI.

8. El sistema de cualquiera de las reivindicaciones precedentes, en el que el primer punto final (WS1, WS2, WS3) está adaptado para ser utilizado para probar una solicitud que llama al, al menos un, servicio web.

9. El sistema de cualquiera de las reivindicaciones precedentes, en el que el segundo punto final (WS1', WS2', WS3') está adaptado para ser utilizado para la difusión de una solicitud (30) que llama al, al menos un, servicio web.

10. El sistema de cualquiera de las reivindicaciones precedentes, en el que el registro (10) comprende además un tercer punto final para una tercera condición de ejecución.

11. Un procedimiento para gestionar una pluralidad de servicios web comprendiendo la etapa de almacenar una descripción (WSDL1, WSDL2, WSDL3) para al menos uno de la pluralidad de servicios web en un registro (10), en el que la descripción del, al menos un, servicio web comprende un primer punto final (WS1, WS2, WS3) para una primera condición de ejecución y un segundo punto final (WS1', WS2', WS3') para una segunda condición de ejecución, **caracterizado** porque

el procedimiento comprende además la etapa de monitorizar los puntos finales (WS1, WS2, WS3, WS1', WS2', WS3) de la pluralidad de servicios web durante la ejecución y emitir un aviso; si se llama a un primer servicio web con un punto final para una primera condición de ejecución y si se llama a un segundo servicio web con un punto final para una segunda condición de ejecución.

12. El procedimiento de la reivindicación 11 comprendiendo además la etapa de seleccionar uno de los primer (WS1, WS2, WS3) y segundo punto final (WS1', WS2', WS3') durante la ejecución dependiendo de la evaluación de la información del entorno sobre la condición de ejecución proporcionada por una solicitud que llama al servicio web.

13. El procedimiento de la reivindicación 11 comprendiendo además la etapa de almacenar la condición de ejecución en el registro; y

escoger uno de los primer (WS1, WS2, WS3) y segundo punto final (WS1', WS2', WS3') durante la ejecución.

14. El procedimiento de cualquiera de las reivindicaciones precedentes 11-13, en el que la etapa de almacenar una descripción para al menos uno de la pluralidad de servicios web en un registro comprende almacenar un primer documento WSDL para el servicio web indicando el primer punto final y almacenando un segundo documento WSDL

para el servicio web indicando el segundo punto final, en el que los primer y segundo documentos WSDL están almacenados en el registro como documentos WSDL relacionados.

15. El procedimiento de cualquiera de las reivindicaciones precedentes 11-13, en el que la etapa de almacenar una descripción (WSDL1, WSDL2, WSDL3) para al menos uno de la pluralidad de servicios web en un registro comprende almacenar un único documento WSDL para el servicio web con componentes característicos indicando los primer y segundo puntos finales.

16. El procedimiento de la reivindicación 15, en el que los valores de los componentes característicos se determinan durante la ejecución utilizando un módulo SOAP de una petición para el servicio web.

17. El procedimiento de cualquiera de las reivindicaciones precedentes 11-16, en el que el registro (10) es un registro UDDI.

18. El procedimiento de cualquiera de las reivindicaciones precedentes 11-17, comprendiendo además la etapa de utilizar el primer punto final (WS1, WS2, WS3) para probar una solicitud que llama al, al menos un, servicio web.

19. El procedimiento de cualquiera de las reivindicaciones precedentes 11-17, comprendiendo además la etapa de utilizar el segundo punto final (WS1', WS2', WS3') para la difusión una solicitud que llama al, al menos un, servicio web.

20. El procedimiento de cualquiera de las reivindicaciones precedentes 11-19, en el que el registro (10) comprende además un tercer punto final para una tercera condición de ejecución.

Fig. 1

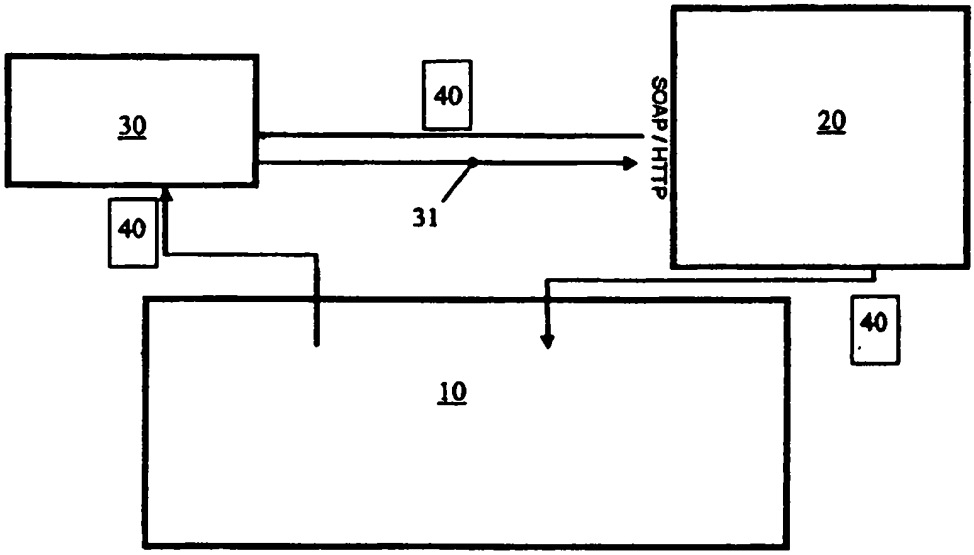


Fig. 2

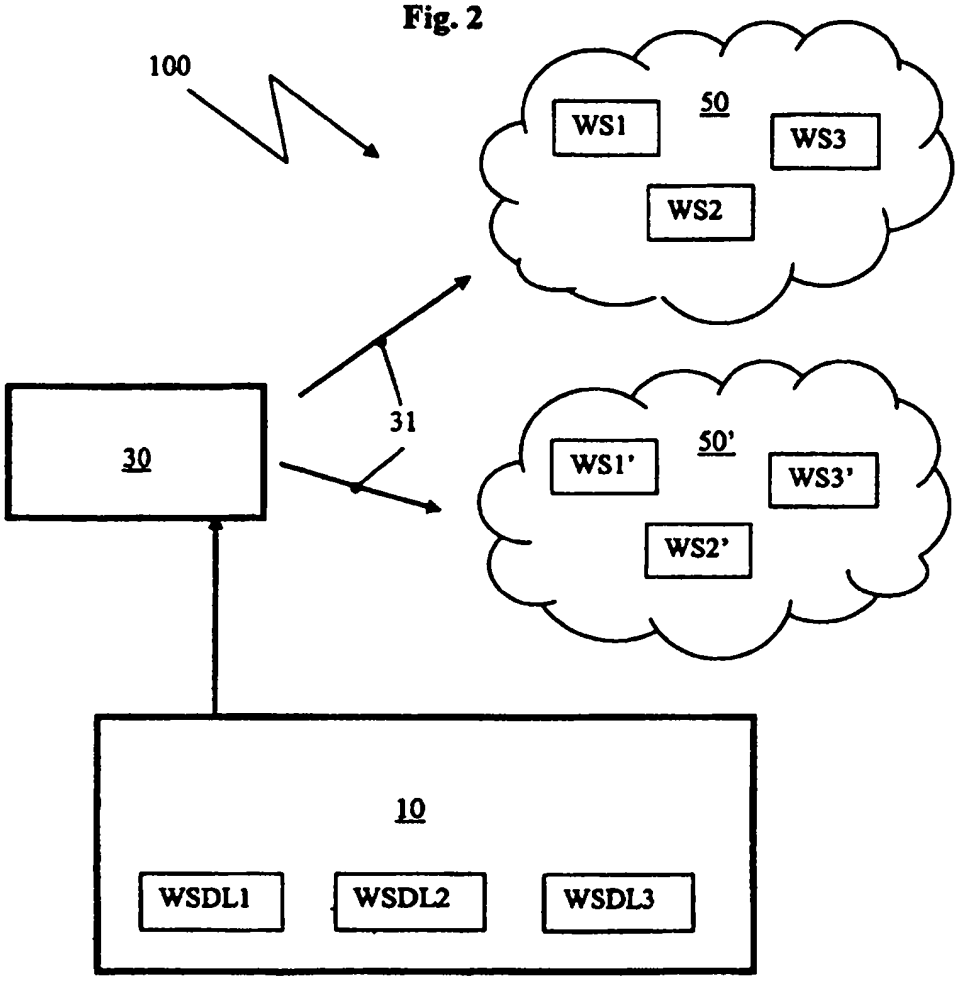


Fig. 3

