



Software

## **Conceptos de Orquestador O2**

**EMPRESAS TUXPAN**

[www.tuxpan.com](http://www.tuxpan.com)

**AÑO 2007**

**TUXPAN**

## INDICE DE CONTENIDO

<i>1</i>	<i>Software de Servicios y Orquestación de Procesos</i>	<u>2</u>
1.1.1	Introducción	<u>2</u>
1.1.2	Software de Orquestación como Integrador	<u>3</u>
1.1.3	Automatización de Procesos	<u>6</u>
1.1.4	Software de Orquestación como Habilitador SOA	<u>8</u>
1.1.5	Arquitectura del Software de Orquestación	<u>10</u>
1.1.6	Ejemplo de Uso del Software de Servicios y Orquestación	<u>11</u>

## 1 SOFTWARE DE SERVICIOS Y ORQUESTACIÓN DE PROCESOS

### 1.1.1 INTRODUCCIÓN

La arquitectura que se propone está basada en los conceptos de SOA y BPM (y su integración formal). A continuación se explican las funcionalidades del software (middleware) que satisface los requerimientos de ser completamente J2EE, con una arquitectura sin estado (stateless), abierta y basada en estándares.

Como software de orquestación y BPM “**Orquestador O2**”, desarrollado por TUXPAN, tiene la característica de haber sido conceptualizado y diseñado como una plataforma habilitadora de SOA y de BPM en conjunto.

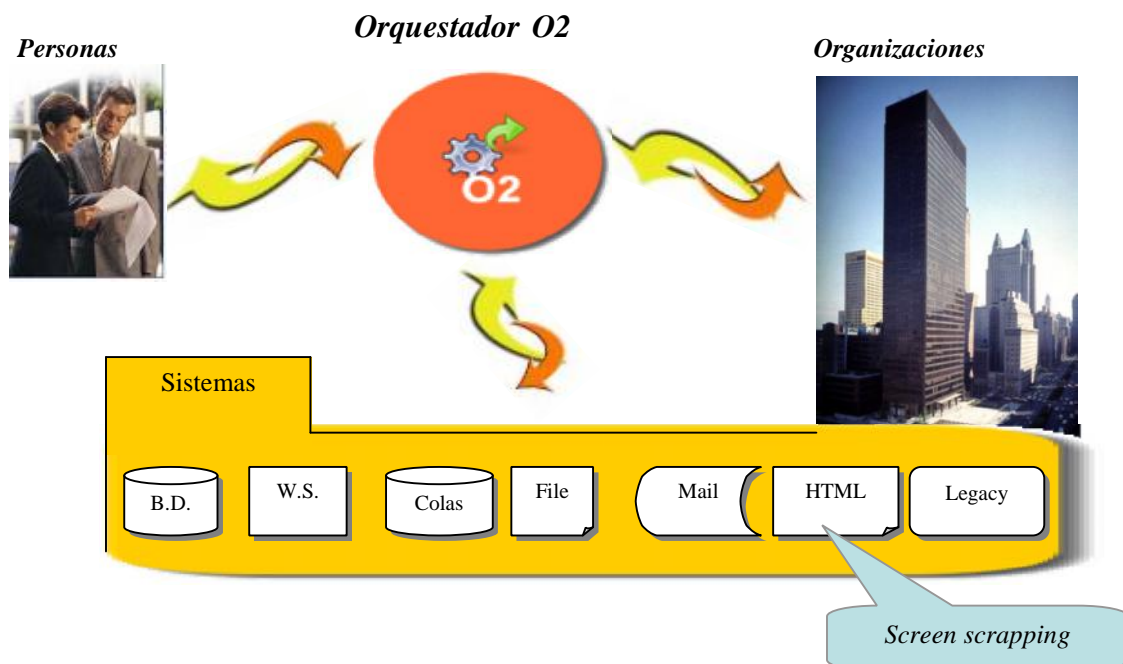
En el mercado existen otras ofertas que presentan funcionalidades equivalentes, sin embargo, estas soluciones corresponden a una suite que integra variados productos (para modelar, para administrar bus de servicios, para auditar, etc.). En el caso del **Orquestador O2**, todas estas funcionalidades están integradas coherentemente y de una forma simple en una aplicación J2EE estándar (el Orquestador se distribuye en el servidor J2EE como una aplicación “ear” más, que usa sus recursos como cualquiera de ellas). Por otra parte, la simpleza se traduce en bajo consumo de recursos, lo que permite que éstos queden disponibles en el servidor de aplicaciones para ser usados en los procesos que se automatizan y en el resto de las aplicaciones J2EE.

Una gran ventaja de contar con un producto como plataforma base de servicios de BPM desarrollada por TUXPAN, es que ésta cuenta con todo el conocimiento para adaptarla a necesidades particulares; por ejemplo, para agregar requerimientos transversales de seguridad para la operatividad entre diferentes organizaciones.

Las funciones que ofrece el Software de Orquestación pueden describirse desde tres enfoques, los que se presentan a continuación.

### 1.1.2 SOFTWARE DE ORQUESTACIÓN COMO INTEGRADOR

Uno de los patrones de arquitectura más complejos de implementar cuando se diseñan e implementan varios sistemas al mismo tiempo, es la integración desacoplada entre ellos. Esto es especialmente cierto si, además, se considera que la puesta en marcha de ellos es con diferencias temporales, y que los sistemas deben (al inicio de su puesta en marcha) operar en forma integrada con sistemas legados, para luego ser “apuntados” hacia los nuevos sistemas que se vayan incorporando.



Una de las principales funciones del Orquestador es la integración, la que se ve reflejada en los siguientes puntos:

#### Integración entre Sistemas

Mediante diferentes protocolos, tal como se muestra en la figura anterior, el Orquestador facilita y promueve la integración desacoplada entre sistemas de información, ya sea para su operación en línea (modo transaccional) o para procesos tipo batch. Los mecanismos para implementar esta integración son variados, desde el consumo y publicación de servicios Web, hasta la interacción con sistemas de mensajería (colas) y varios tipos de archivo para traspaso (fuera de línea) de información.

Los mecanismos de integración pueden ser sincrónicos o asincrónicos, los que dependerá del protocolo (o canal) que se utilice para la comunicación, y de las necesidades y conveniencias de cada proceso que se automatiza.

La integración es desacoplada. Independiente del canal de comunicación que se utilice, los accesos a la utilización de ese canal pueden ser fácilmente modificados (desde los procesos que se orquestan en la misma herramienta) para apuntar hacia nuevos destinos o ser resueltos como otros procesos compuestos dentro del mismo Orquestador. De esta forma, se asegura una integración desacoplada entre las aplicaciones que componen la suite de soluciones del Servicio, lo que permitirá fácilmente reemplazar algunas de ellas realizarle mantenciones (producto de cambios en las leyes, por ejemplo) sin afectar el correcto funcionamiento del resto.

### **Integración con Personas**

La plataforma de orquestación O2 ofrece además algunas características de workflow, las que incluyen notificaciones (mediante correo electrónico), confirmaciones manuales (para iniciar o continuar procesos sólo luego de haber sido confirmados por usuarios con ciertos roles o privilegios), ingreso de datos o User Data Entries, los que permiten completar procesos en curso (detenidos, en estado de espera de estas acciones de completación) con datos extra para continuar su ejecución, la posibilidad de iniciar procesos desde un portal, completando datos de inicio si es necesario (opciones útiles para operadores que, por ejemplo, deben realizar tareas manuales como “generar archivo de traspaso” u otras) y, finalmente, la calendarización de tareas como procesos automáticos a ser ejecutados en periodos de tiempo configurables.

Si bien la orientación del Orquestador O2 es la integración de procesos automáticos, éste ofrece suficientes elementos de workflow como para ser usados en aquellas excepciones de negocio en que se requiera intervención manual. Para el caso de los workflows de negocio que incluyen las aplicaciones, éstos serán modelados y construidos dentro de cada sistema y sus actividades serán orquestadas dentro de O2, al ser expuestas como servicios en el bus privado.

Un workflow (basado principalmente en decisiones humanas) dentro de un proceso de negocio, será entonces resuelto usando el software de Orquestación como base para el consumo, publicación y orquestación de servicios básicos, mientras que el estado completo será administrado desde las aplicaciones (como servicios de aplicaciones J2EE) y expuesto a los procesos para integrarse al resto de los sistemas. El modelamiento e implementación de los servicios atómicos dentro de las aplicaciones J2EE será basado en diagramas de actividades (para los casos más complejos si se requiere) o en Máquinas de Estado y su implementación como objetos con pares de atributos “estado-versión” que los implementen.

### **Integración entre Organizaciones**

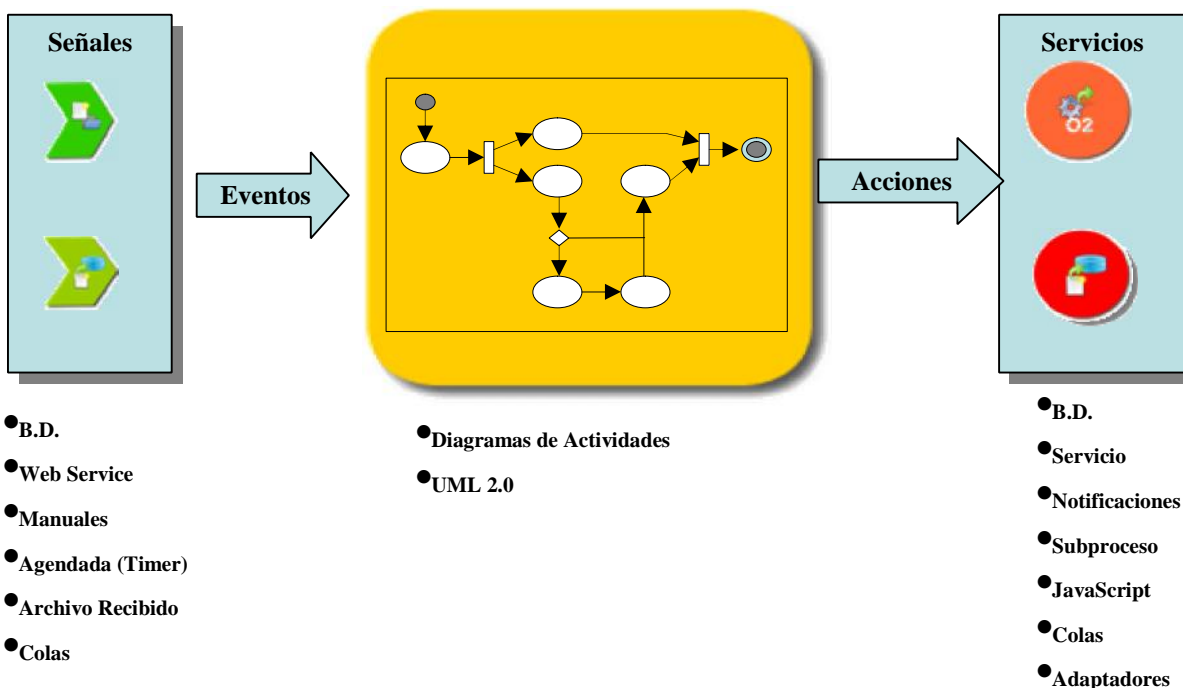
La separación de los servicios en un Bus privado y un Bus público, permiten diseñar los servicios de las aplicaciones de acuerdo a la seguridad en sus accesos que ellos requieran. En el Bus público se pueden registrar aquellos servicios que se desea que sean consumidos desde otras organizaciones, aplicando los mecanismos de seguridad que se desee. En este caso, estos mecanismos incluyen la interacción con los servicios de seguridad expuestos por el grupo (clúster) de servidores del negocio de Seguridad y Auditoría (con sus propias instancias del Orquestador exponiendo estos servicios).

Gracias al alto grado de desacoplamiento que se produce al invocar servicios de los buses desde los procesos que se orquestan o directamente desde las aplicaciones, los servicios externos (que se consuman desde otras organizaciones) pueden ser fácilmente remplazados sin afectar a sus clientes (consumidores).

Mediante la administración por separado de buses empresariales para diferentes consumidores (se explica a continuación) el Orquestador permite efectivamente dedicar y controlar con un alto nivel de detalle la invocación (consumo) de los servicios por otras organizaciones.

### 1.1.3 AUTOMATIZACIÓN DE PROCESOS

Una posible definición para un proceso de negocio es la transformación de entradas en salidas, producto de la ocurrencia de ciertos hechos externos, que provocan esta reacción, y que aportan con las entradas. Esta simplificación de un proceso puede verse en el siguiente esquema:



Mediante diferentes protocolos y mecanismos, el software de Orquestación y servicios detecta (o es informado de) la ocurrencia de hechos externos (señales). Algunos de estos hechos pueden incluir datos (por ejemplo, si se inserta un registro de una tabla en una base de datos, la señal incluye los campos del registro insertado). La señal puede ser mapeada (asociada) a uno o más eventos, los que pueden corresponder al inicio de una nueva instancia de un proceso modelado, o a la continuación de un proceso en espera de la ocurrencia (y los datos) de la señal.

Una vez que un proceso es iniciado, el motor de ejecución de BPM usa el diagrama de actividades definido para ese proceso para ejecutar la lógica de coordinación de las actividades de ese proceso. Cada actividad UML definida dentro del diagrama del proceso incluye una serie de acciones, las que corresponden a la "reacción del proceso sobre el medio" producto de la señal que lo activó.

Las acciones de las actividades ejecutan servicios (del bus privado) o configuran tareas específicas del orquestador, como acciones sobre Bases de Datos (insertar registros), envío de notificaciones, colocar mensajes en colas, etc.

Usando un protocolo de señales y eventos dentro de los modelos UML, es posible para un proceso informar la ocurrencia de hechos que pueden ser capturados por el mismo orquestador para reaccionar con nuevos procesos en forma desacoplada.

### **Integración entre Procesos de Negocio y el clúster Seguridad y Auditoría**

Se identifican operaciones de negocio (concepto recursivo, en el que una operación puede componerse de otras). Las operaciones de negocio tienen la característica de ser: seguras (requieren de un privilegio de usuario), auditables (se registran en la base de datos de auditoría) y medibles .

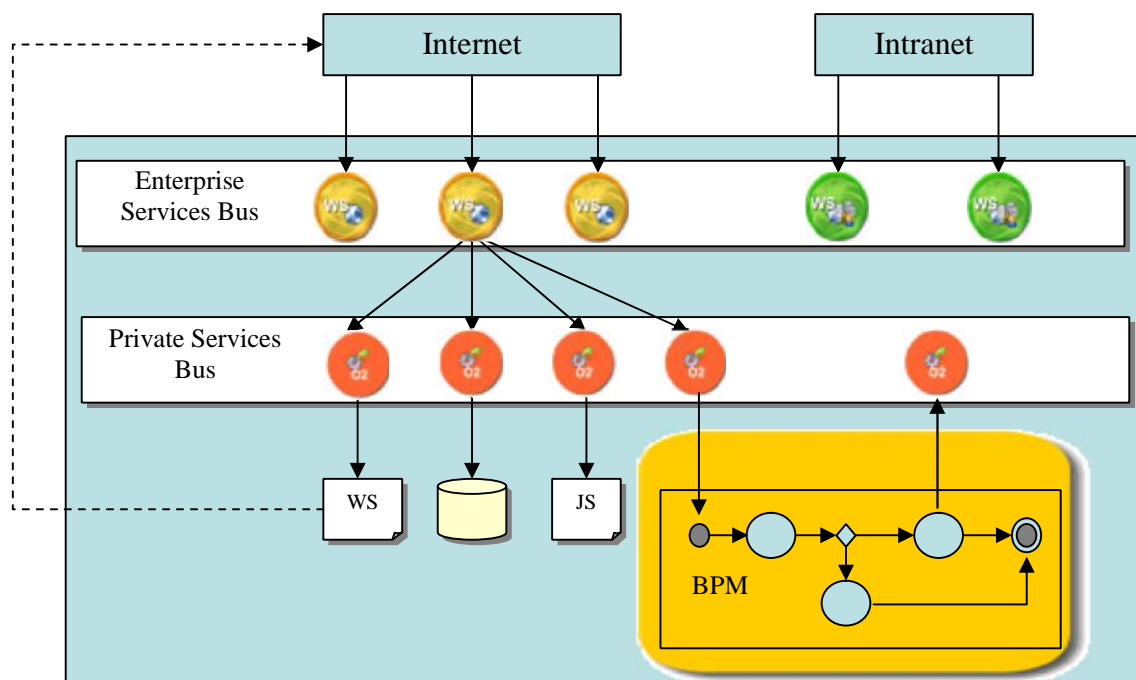
Muchas de las operaciones de negocio se implementan como procesos orquestados en O2 (expuestos en un Bus de servicio). El clúster de Seguridad y Auditoría está también basado en el mismo Software de Orquestación y publica sus servicios en el Bus público. Los procesos que implementan operaciones de negocio invocan dentro de sus actividades a los servicios de Seguridad (los que antes se han publicado en el Bus privado) y, dependiendo del resultado de estas invocaciones, el proceso que implementa la operación toma decisiones como abortar su ejecución (por falta de privilegios, por ejemplo).

Para aquellas operaciones que se implementen como servicios atómicos (desde métodos de EJB sin estado en aplicaciones J2EE) el funcionamiento es similar. La diferencia es que las invocaciones a los servicios del Bus público del clúster de Seguridad y Auditoría se hacen directamente desde el código Java mediante el protocolo SOAP.



### 1.1.4 SOFTWARE DE ORQUESTACIÓN COMO HABILITADOR SOA

El Software de Orquestación es un habilitador de arquitecturas orientadas a servicios, ya que ofrece las funcionalidades de definición y administración de buses privados y públicos, en donde se lleva a cabo la integración y se colocan a disposición de los clientes (de cada servicio) los procesos orquestados (BPM) como servicios disponibles para ser invocados.



El Bus de Servicios privado contiene aquellas definiciones de servicios que podrán ser coordinados (orquestados) desde dentro de los procesos que se modelen y ejecuten sólo en la misma instancia del Software de Orquestación. Si se desea que un servicio del Bus privado (de cualquier tipo, incluyendo procesos O2) pueda ser invocado desde otro servidor (O2) o desde otras aplicaciones o capas de presentación, estos servicios deben ser expuestos en el Bus privado, desde donde serán invocados mediante algún protocolo remoto (Se soporta en este momento SOAP para web services, RMI para invocaciones desde aplicaciones J2EE hacia un EJB sin estado y un puente de conexión para aplicaciones GWT).

Como se ejemplifica en la figura anterior, el Software de Orquestación integra los procesos y servicios (BPM y SOA) ofreciendo un alto nivel de enmascaramiento

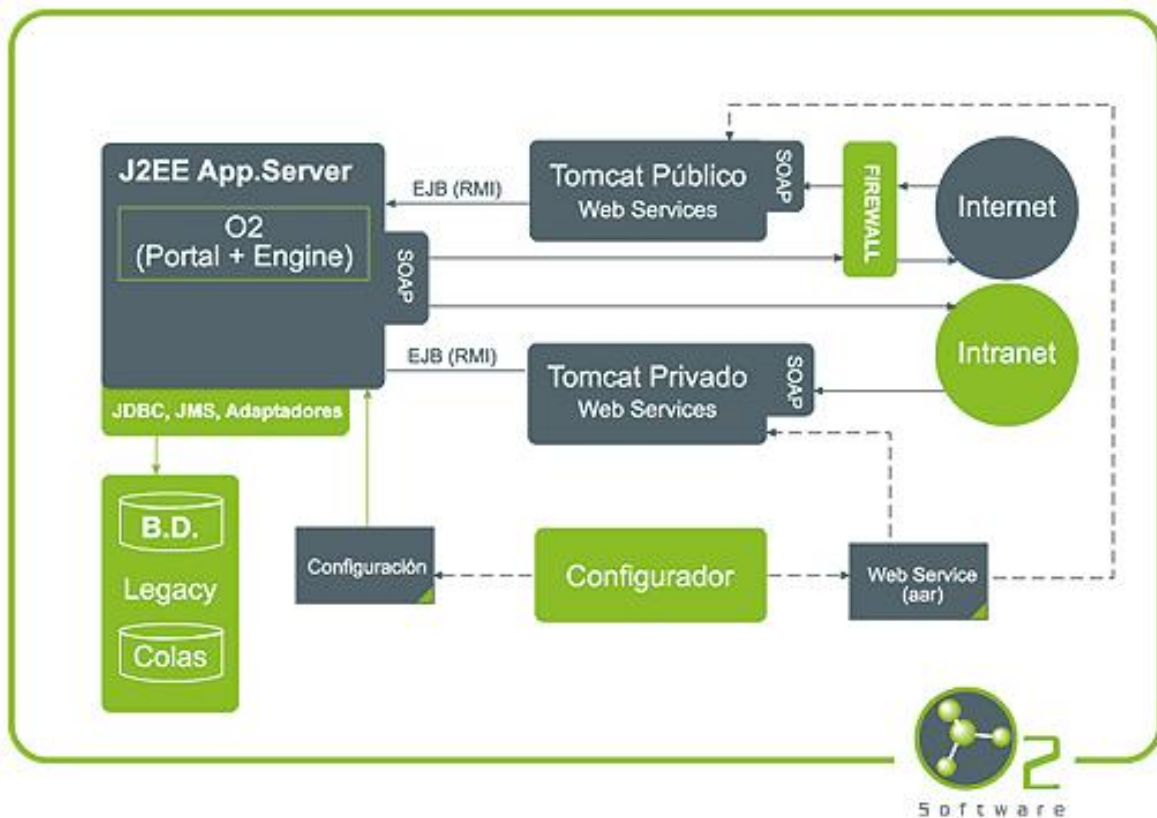
a los clientes (de los buses de servicios). Un proceso modelado y ejecutándose dentro del Orquestador equivale a un servicio, el que transforma entradas en salidas y realiza acciones.

Un servicio privado del tipo “proceso O2” corresponden a la visión de caja negra de un proceso que se ejecute en el mismo servidor O2.

Al exponerse un proceso dentro del Bus privado del Orquestador, este proceso (visto ahora como servicio) puede ser orquestado desde otros procesos (en el mismo u otro servidor, importándolo hacia su propio bus privado) definiendo procesos en forma recursiva (un proceso se define como la coordinación de procesos) Esto permite elevar el nivel de abstracción de los servicios, desde los atómicos, resueltos por aplicaciones J2EE, hasta los procesos de alto nivel, cercanos al negocio que se automatiza.

Además de los servicios resueltos por procesos y de las invocaciones a servicios atómicos resueltos por aplicaciones J2EE, el Orquestador permite crear nuevos servicios de algunos de los tipos básicos soportados, como Servicios escritos en JavaScript, servicios que inserten registros en bases de datos, consumo de web services externos, etc.

### 1.1.5 ARQUITECTURA DEL SOFTWARE DE ORQUESTACIÓN



El Software de orquestación está construido como una aplicación estándar J2EE y basado en productos, lenguajes y protocolos abiertos.

En la figura anterior se muestran los diferentes componentes que configuran el Software de Orquestación y la forma en que ellos interactúan. En el servidor J2EE se aloja la aplicación principal, que es la encargada de la ejecución de los procesos y servicios. Los servidores Tomcat externos son los encargados de implementar el bus público de servicios, y se recomienda separar aquellos que serán públicos (invocados desde otras organizaciones) de los privados (dentro de la misma organización) ya que se aplicarán políticas de seguridad diferentes a cada uno de ellos (encriptación por hardware, firewalls, etc.).

La publicación y el consumo de Web Services se basan en el producto Apache AXIS2, el que es ampliamente usado el mercado y es incluso la base de varios productos comerciales. Los servidores Tomcat que implementan el Bus Empresarial ejecutan la aplicación (war de J2EE) axis2, la que implementa el protocolo SOAP y que agrega un punto más de escalabilidad a la solución completa, separando la ejecución de las transformaciones necesarias para implementar los protocolos de comunicación (SOAP, GWT) de la resolución de los servicios de negocio. Tanto los servicios como los procesos, es decir la

lógica de los negocios, es ejecutada siempre en los servidores de aplicación J2EE y no en los servidores Tomcat.

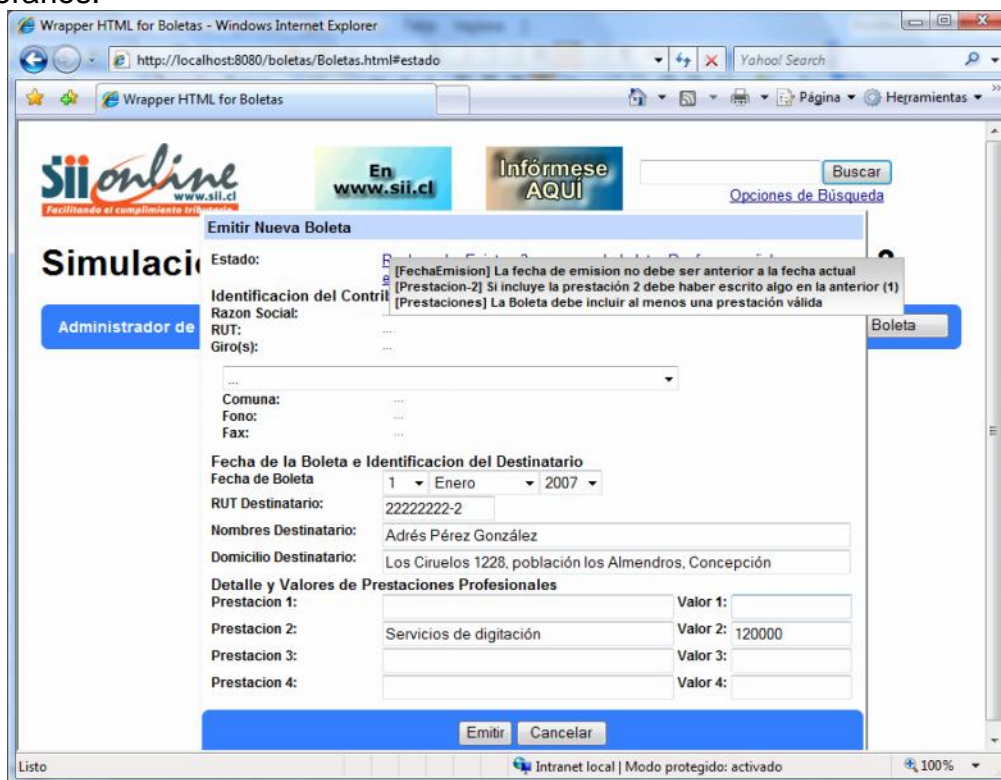
### 1.1.6 EJEMPLO DE USO DEL SOFTWARE DE SERVICIOS Y ORQUESTACIÓN

A continuación se presenta el desarrollo de un ejemplo de uso del Software de Orquestación y Servicios O2, el que clarifica el tipo de problemas que mejor resuelve, y muestra los altos grados de adaptabilidad y reusabilidad de procesos y servicios que las soluciones basadas en él alcanzan.

Este ejemplo se desarrolló utilizando información pública del Servicio de Impuestos Internos de Chile (SII) y permite mostrar los beneficios que se obtienen al usar Orquestador O2.

#### Primera Iteración

El ejemplo muestra el ingreso de un formulario para la emisión de una Boleta de Honorarios.



The screenshot shows a web browser window titled "Wrapper HTML for Boletas - Windows Internet Explorer". The address bar shows "http://localhost:8080/boletas/Boletas.html#Estado". The page content includes the SII online logo and a search bar. The main form is titled "Emitir Nueva Boleta" and contains the following fields:

- Estado: [Dropdown menu]
- Identificación del Contribuyente:
  - Razon Social: [Text input]
  - RUT: [Text input]
  - Giro(s): [Text input]
- Comuna: [Dropdown menu]
- Fono: [Text input]
- Fax: [Text input]
- Fecha de la Boleta e Identificación del Destinatario:
  - Fecha de Boleta: [Dropdown menu] 1 Enero 2007
  - RUT Destinatario: 22222222-2
  - Nombres Destinatario: Adrés Pérez González
  - Domicilio Destinatario: Los Ciruelos 1228, población los Almendros, Concepción
- Detalle y Valores de Prestaciones Profesionales:

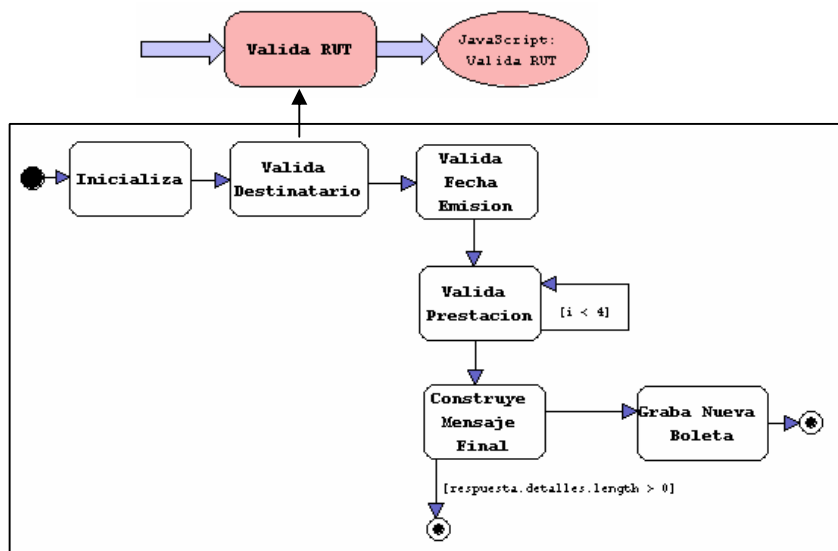
Prestacion	Valor
Prestacion 1:	Valor 1:
Prestacion 2: Servicios de digitación	Valor 2: 120000
Prestacion 3:	Valor 3:
Prestacion 4:	Valor 4:

Buttons: "Emitir", "Cancelar", "Boleta".

La aplicación GWT debe invocar un servicio del Orquestador que se encargue de validar los datos de la boleta que se emite y grabarla en caso de ser aceptada. En caso de detectarse problemas, se debe informar de vuelta a la interfaz de usuario indicando el campo a que corresponde cada mensaje de error, para que la aplicación determine la mejor forma de presentar estos errores (por ejemplo cambiando el color de las etiquetas de los campos con problemas).

Para resolver los requerimientos de negocio del ejemplo anterior, se define en el Orquestador un nuevo formato de documentos (un nuevo tipo de datos compuesto) que representa a la “Boleta de Honorarios” y se crea un nuevo proceso llamado “Emite Nueva Boleta Honorarios”, el que recibe como entrada el objeto compuesto (y definido por el formato de documentos antes creado) Boleta, lo procesa y retorna otro objeto compuesto con el resultado de la operación (resultado global, mensaje global y una colección de mensajes por cada uno de los campos con problemas en la boleta)

El proceso modelado y ejecutable en O2 se presenta en la siguiente figura:



Cada proceso de O2 se ejecuta sobre un “contexto” que representa su visibilidad (las variables que conoce cada instancia de ese proceso y que son modificadas por las diferentes acciones que en él se ejecutan). En este caso se inicializan contadores, se realizan validaciones (destinatario y fecha) y luego se valida cada una de las prestaciones (iterando sobre ellas). Si no hay errores que informar, se invoca un servicio encargado de grabar la boleta.

La actividad “Validaciones Destinatario” incluye varias acciones JavaScript para validar los campos como el nombre y la dirección del destinatario (que no sean

vacíos). La validación del RUT del destinatario se reconoce como un servicio reutilizable, por lo que se decide crear un nuevo servicio en el Bus Privado, de tipo JavaScript, y que valide cualquier RUT que se le pase como parámetro, implementando el algoritmo del módulo para el dígito verificador. De esta forma, cualquier otra actividad en cualquier proceso que los requiera, podrá invocarlo.

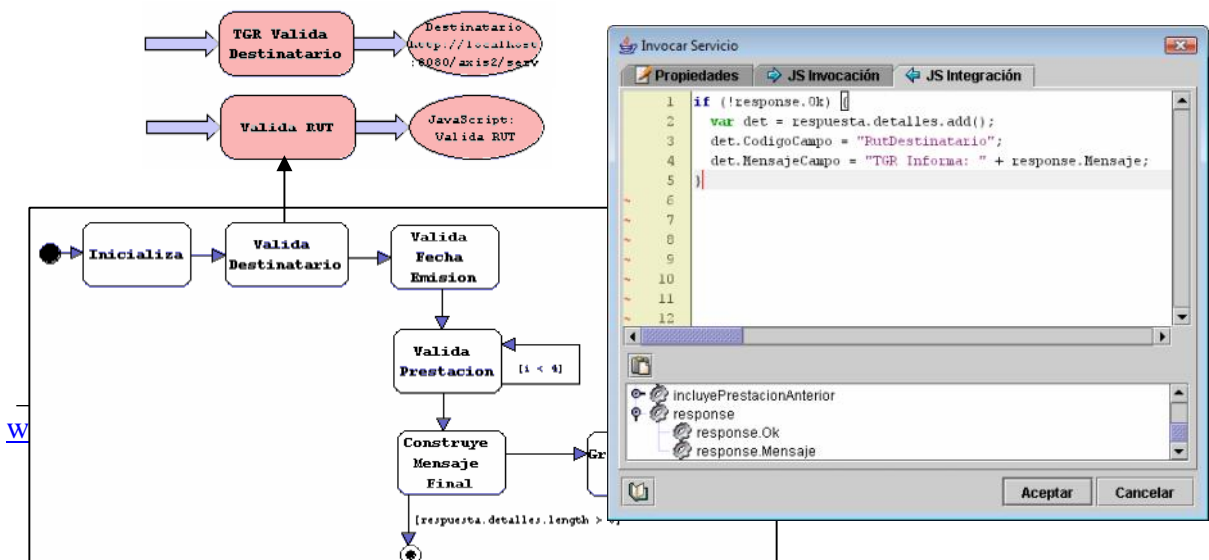
Cualquier cambio en la lógica de validación sobre los campos de la boleta se puede realizar sobre las acciones dentro de las actividades del proceso. Los cambios se graban (como un archivo de configuración) y se suben al sitio del portal del Orquestador. Las modificaciones quedan inmediatamente disponibles para las nuevas invocaciones. Si los cambios incluyen además modificaciones sobre las interfaces de comunicación con los servicios (APIs), es posible crear nuevas configuraciones y mantener las anteriores funcionando para los procesos que con ellas se hayan iniciado.

### Segunda Iteración

Se debe satisfacer el siguiente requerimiento (de ejemplo): “Una organización externa provee un Servicio Web que valida si un contribuyente (desde su RUT) puede ser destinatario de una boleta de honorarios. Se debe agregar una invocación este servicio remoto como parte de las validaciones normales durante la emisión de todas las boletas de honorarios”.

En el Bus Privado del orquestador se crea un nuevo servicio del tipo “Consumir Web Service”. Usando la dirección URL con la información de la metadata del web service (WSDL) el Orquestador crea automáticamente los tipos (complejos) que representan a las entradas y salidas del servicio que se consumirá, y completa los parámetros propios del protocolo de invocación (document type, SOAP operation).

Se agrega a la actividad de “Validación del Destinatario” una nueva acción del tipo “Invocación de Servicio desde el Bus Privado” y se le entregan sus argumentos y se procesa la respuesta del servicio desde la organización externa.

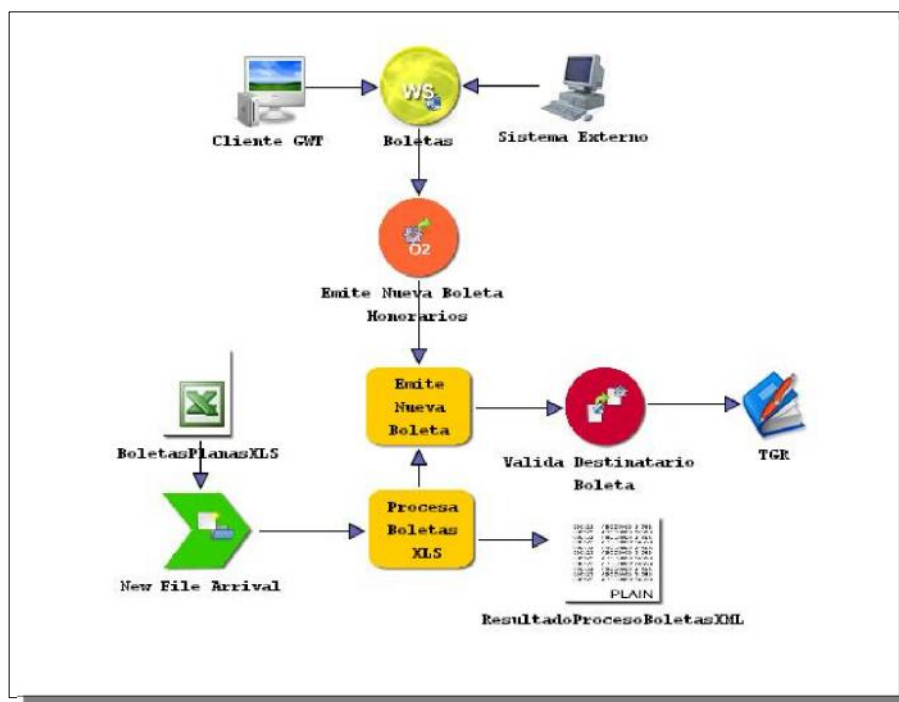


Como se muestra en la figura anterior, se agrega al proceso una nueva acción que invoque al servicio creado en el Bus Privado y, dependiendo de su respuesta, se agrega un nuevo mensaje de error a la salida del proceso. La nueva configuración es subida al servidor y las nuevas boletas incluyen instantáneamente (en “caliente”) la nueva validación.

### Tercera Iteración

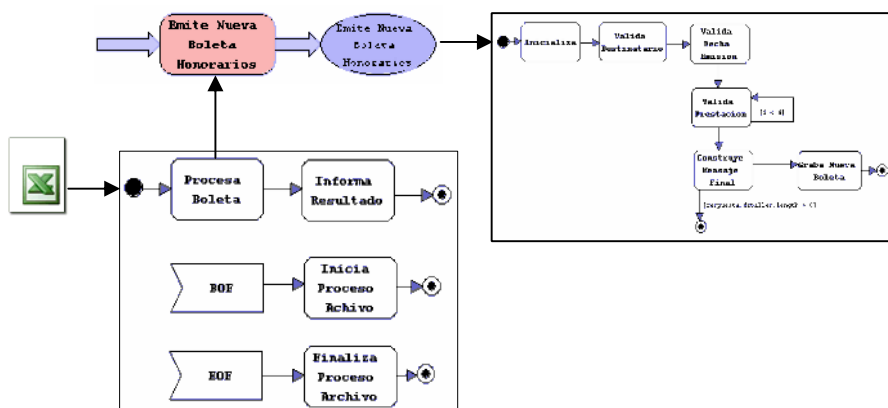
Se desea ahora procesar una serie de Boletas en modo batch. El origen de los datos de cada boleta es un archivo Excel, y se desea que el resultado se almacene en un archivo XML de salida, en donde se identifique cada boleta y los posibles mensajes de error para los campos de cada una de ellas.

El Orquestador ofrece la posibilidad de representar archivos como entradas y salidas en varios formatos (planos, CVS, Excel, XML). Cada formato de archivo (físico) está asociado a un formato de documento (lógico) y permite leer / escribir registros con invocaciones simples a servicios dentro de procesos y de acuerdo a cómo se parametricen estos formatos. Esto significa que pueden luego cambiarse los formatos de los archivos (anchos de columnas, nombres de tags, etc.)



La figura anterior muestra un diagrama de contexto (modelado en O2) con la solución. En ella se aprecia la recepción del archivo Excel, el que dispara una

nueva señal (New File Arrival) y que esta señal se mapea al inicio de un nuevo proceso (Procesa Boletas XLS) cuyo modelo se muestra a continuación. Por cada fila de datos (con una boleta cada una) del archivo de origen, se invoca al proceso anterior (se reutiliza como un servicio más)  
Emite Nueva Boleta y su resultado es capturado y agregado al archivo XML de salida con los resultados.



El Orquestador dispara una señal al inicio y al final del procesamiento del archivo de entrada. Estas son capturadas en el proceso inicial (para inicializar el archivo de salida y luego para moverlo hasta su ubicación final). Por cada fila se invoca al servicio “Emite Nueva Boleta” el que corresponde a la exposición del mismo proceso anterior en el Bus Privado.

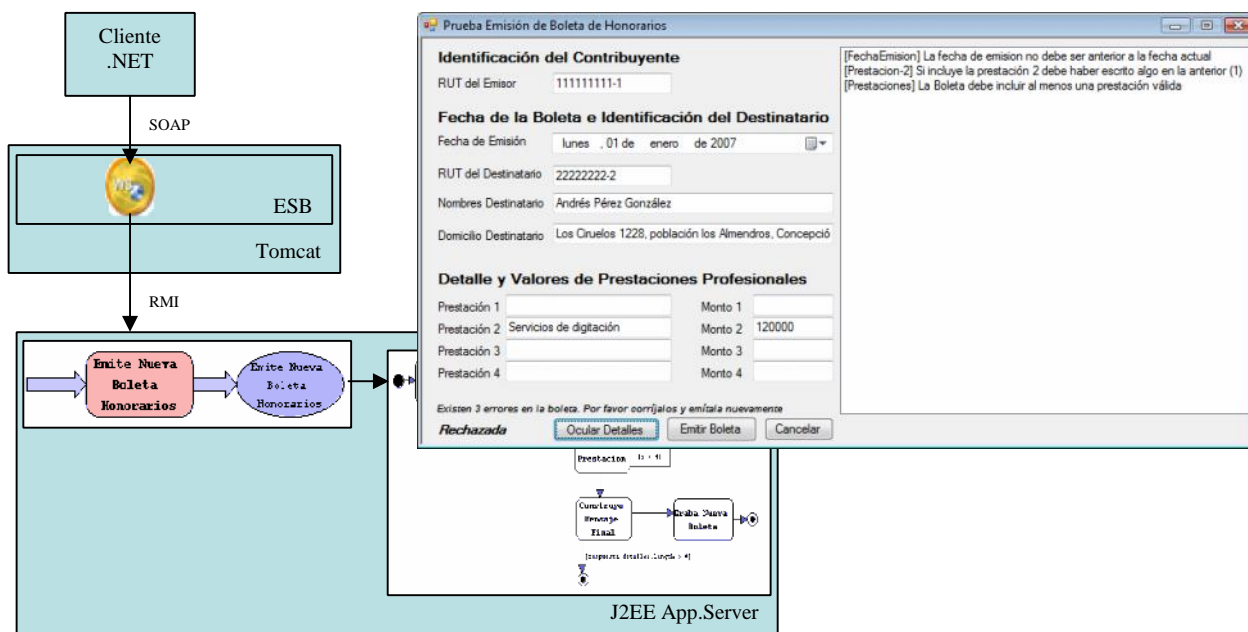
En este ejemplo se observa claramente las capacidades de reutilización de un proceso completo al colocarlo como un servicio más dentro del Bus Privado, y disponible para nuevas orquestaciones.



## Cuarta Iteración

Se desea que las Boletas de Honorario puedan ser creadas automáticamente desde otros sistemas, y no necesariamente digitadas por la web por los contribuyentes.

El proceso inicial “Emite Nueva Boleta Honorarios” ya se encuentra disponible en el Bus Empresarial para poder ser invocado por la aplicación web (GWT en este caso). Al estar en este Bus, sólo es necesario generar su representación como Web Service para axis2 (desde la herramienta cliente de modelamiento del Orquestador) y distribuirla en el servidor Tomcat que se desee (de acuerdo a su disponibilidad como Internet o Intranet).



Como se observa en el ejemplo, una vez que un proceso a sido modelado y está en ejecución dentro del Orquestador, publicarlo como servicio web (dentro o fuera de la organización) es tan simple como exponerlo en el Bus Privado (realizando si se desea transformaciones de datos para sus entradas y salidas) y, desde ahí, en el bus Empresarial.