Tecnología

ARQUITECTURA CONCEPTOS TECNOLÓGICOS

Requerimientos de Integración
Uno de los principales requerimientos del Orquestador es que debe ser un facilitador de la Integración automática.

Esta integración se refleja y debe ser soportada en los siguientes escenarios:
Integración entre Sistemas.
El Orquestador debe apoyar la implementación del patrón de arquitectura de ?Integración por Orquestación?. El objetivo de este patrón de arquitectura es establecer un marco (protocolos, diseño y metodología) sobre el que se deben construir los nuevos sistemas de una organización para evitar al máximo las dependencias estructurales entre ellos. Cada vez que se crea una dependencia estructural entre dos sistemas (aquella en que desde un sistema se debe conocer algún detalle de las estructuras de otro, como por ejemplo un acceso desde un sistema a una base de datos administrada por otro) se están aumentando los futuros esfuerzos (y con ello los costos) en la mantención de esos sistemas. Los cambios en alguno de ellos van a afectar al correcto funcionamiento del otro.

La Integración por Orquestación propone que los sistemas ofrezcan sus servicios para ser integrados por una herramienta que los coordine, de acuerdo a alguna lógica de negocio definida por el proceso que se automatiza. De esta forma, los sistemas no invocan directamente a los servicios (ni acceso a datos) de otros, sino que sólo colocan a disposición de un orquestador sus servicios de negocio y le solicitan sólo a él la ejecución de nuevos servicios, y no directamente a los sistemas finales que los resuelven. Además de coordinar la ejecución de estos servicios, el orquestador debe ofrecer las capacidades de transformación de datos. Gracias a estas características, un sistema completo puede ser reemplazado por otro que satisfaga la misma funcionalidad, y sin afectar al resto de las aplicaciones que utilizan sus servicios (terminando en una considerable reducción en los costos de mantención de sistemas).
Integración entre Organizaciones
Por otra parte, si se desea integrar comunicaciones automáticas entre los sistemas de la organización con otras compañías, el Orquestador deberá ofrecer las capacidades de generar (salida) y de procesar (entrada) archivos que permitan intercambiar información como parte de procesos integrados. Las salidas de los procesos de una organización (aplicando alguna lógica) son las entradas a los procesos de otra, formando procesos complejos que atraviesan

organizaciones.

La transferencia de información usando archivos es útil en procesos de carga masiva (tipo batch) o para los casos de integración en que interactúan sistemas legados. Para el caso en que se desea integración en línea (transaccional) la comunicación entre los sistemas de las organizaciones deberá ser en línea, usando protocolos como los Servicios Web (basados en SOAP).
Integración con Personas
El objetivo principal de un motor de ejecución para BPM es la solución a la automatización de procesos de negocio. Se permitirá que algunos de estos procesos incluyan la intervención de personas en la toma de decisiones o el ingreso manual de datos, agregando capacidades de workflow al Orquestador. La herramienta deberá además ofrecer la posibilidad de enviar correos electrónicos a ciertas personas en la forma de notificaciones, como envío de alertas que permitan coordinar la respuesta a ciertas excepciones. Gracias a estas características, el Orquestador facilitará la integración de Personas, Sistemas y Organizaciones, dentro de procesos de negocio complejos.
Protocolos de Integración
En la Figura 1 se observa un esquema simplificado de la forma en que el Orquestador resolverá las funcionalidades de integración entre sistemas de la misma organización y con los sistemas de otras organizaciones.

El Orquestador debe proveer varios mecanismos para facilitar la integración de los sistemas internos, utilizando técnicas poco invasivas. Si los sistemas internos se diseñan y construyen de acuerdo al patrón de integración por orquestación, éstos pueden disparar señales y consumir servicios directamente desde esta herramienta usando estándares como web services. Para los sistemas que ya existen en la organización, se propone usar mecanismos de comunicación que no requieran de la mantención de esos sistemas (que pueden ser productos externos de los que no se tiene acceso a los códigos fuentes). Estos mecanismos serán principalmente comunicación entre bases de datos o traspaso de archivos.

Los mecanismos y protocolos propuestos para coordinar a los sistemas internos son:

Web Services por Intranet para los sistemas que se diseñen bajo arquitectura SOA

Acceso (lecturas / escrituras) a bases de datos de esos sistemas desde el Orquestador. Se podrán definir nuevos servicios (en el Orquestador) que accedan a bases de datos corporativas, y que luego sean coordinados dentro de procesos.

Mensajería por Colas. Los sistemas que puedan ser mantenidos podrán enviar y procesar mensajes usando algún servidor de colas. Desde el Orquestador, se podrán definir servicios que coloquen y que lean mensajes hacia y desde estas colas, usando el protocolo (API) JMS de J2EE.

Archivos de traspaso. Muchos sistemas legados generan y procesan archivos planos. Éstos podrán seguir siendo utilizados, y se procesarán como señales de entrada a la ejecución de procesos orquestados.

De igual forma, el Orquestador ofrece servicios nativos para la generación de archivos de estos tipos, los que luego podrán ser consumidos por los sistemas legados.
El orquestador provee una interfaz web (portal) para que los usuarios puedan interactuar con los procesos, colocando datos de entrada, tomando decisiones y actuando en casos de excepciones. Las señales que inician procesos podrán contener datos de entrada, los que serán digitados por el usuario que dispara la señal. El Portal crea dinámicamente, y de acuerdo a la estructura de los datos de entrada para ese proceso, una página web para el llenado de esos datos.

Algunos procesos podrán quedar detenidos a la espera de algún ingreso de datos (completar información faltante) o confirmación de algún usuario en un rol determinado. Al igual que para las señales manuales, el Portal deberá desplegar dinámicamente una página para recibir estos datos o la confirmación del usuario responsable, para luego continuar con el ejecución del proceso que se encontraba detenido.

Además de las interacciones anteriores con usuarios, el motor de ejecución de los procesos deberá enviar notificaciones vía correo electrónico en los puntos del proceso que así lo especifiquen.

Estas tres características (iniciar proceso con datos de entrada, completar o confirmar para continuar proceso y recibir notificaciones) implementarán las funcionalidades de workflow que ofrecerá el Orquestador.
Requerimientos de BPM
El Orquestador debe permitir que se modelen y documenten procesos de negocio, y que luego estos modelos puedan ser ejecutados por un motor que los interprete de acuerdo a la lógica de control de los diagramas de actividades, y a la ejecución específica de cada tipo de acción dentro de las actividades (invocaciones de servicios, consultas SQL, etc.)
Motor de Ejecución de Procesos
Los procesos se inician como respuesta a una señal del mundo exterior. Si la señal incluye datos, éstos podrán ser inyectados (previa trasformación) como datos de entrada a cada nueva instancia del proceso. Por ejemplo, si la señal corresponde a una señal disparada manualmente por un usuario y se definen datos de entrada para ellas (a través de un Formato de Documento que defina su estructura) el evento mapeado en la señal puede tomar los datos que viajan con la señal (llenados por el usuario que la inicia en el Portal Web), transformarlos y asignarlos a las variables del contexto del nuevo proceso que se inicia.

Durante la ejecución de la lógica del proceso, las variables de su contexto son accesadas para lectura y escritura desde sus acciones. Por ejemplo, si desde una acción dentro de un proceso se invoca a algún servicio del Bus, los argumentos de entrada del servicio se pueden obtener desde variables del contexto del proceso (incluyendo transformaciones) y la salida o resultado de la ejecución de ese servicio podrá ser mezclada (mediante transformaciones) e integrada a las variables del contexto para que continúe su ejecución.

Según el modelo planteado, un proceso se puede considerar también como un transformador de datos de entrada en datos de salida (contexto modificado desde su entrada hasta su salida, de acuerdo a reglas de negocio). El proceso completo puede ser visto como un servicio que recibe argumentos y retorna datos de salida. El Orquestador debe permitir publicar procesos del mismo Orquestador como servicios en el Bus, para ser consumidos (orquestados) dentro de otros procesos o publicados como Web Services para ser invocados desde Internet o Intranet. Esto se indicará con más detalle en la especificación de los requerimientos de SOA.

La figura 2 muestra un esquema del Orquestador como automatizador de procesos.

Señales
Como ya se ha mencionado, los procesos se iniciarán (o continuarán si estaban en estado de espera) mediante señales provenientes desde el exterior y mapeadas hacia ellos mediante eventos.

El Orquestador deberá proveer un conjunto de señales básicas que será posible capturar y asociar (mediante eventos) a la ejecución de procesos. Estos tipos básicos de señales son:

Arribo de un Archivo a un Directorio
Se podrán configurar señales que se activen cada vez que un archivo de ciertas características (de su nombre y asociado a un Formato de Archivo) se reciba en un directorio visible por el servidor.

DBRead
Se podrá activar una señal cada vez que se inserte un registro en una tabla, capturando los valores de sus campos, de acuerdo a la estructura de un Formato de Documento. Estos datos podrán ser usados para iniciar una nueva instancia o continuar un proceso en ejecución, inyectando los valores insertados en la base de datos dentro del contexto del proceso que se inicia o continúa.

El Orquestador leerá el registro, mapeará sus eventos a procesos y luego eliminará el registro leído, por lo que se propone que este tipo de señales actúen sobre tablas ?de paso? creadas específicamente para integrar sistemas existentes y actuar de acuerdo a ciertos hechos que en ellos se registren. Dependiendo del servidor de bases de datos que se trate, se podrán definir triggers que provoquen estos ?inserts? automáticamente, sin necesitarse mayores mantenciones a los sistemas que se desee integrar. Por ejemplo, será posible iniciar un proceso como respuesta a la creación de una nueva factura por parte de un sistema de venta, y disponer de los datos de esa factura recién creada como variables del contexto del proceso que se inicia.

Señal Manual
Algunos procesos podrán ser iniciados o continuados de acuerdo a las indicaciones explícitas de un usuario (operador) del sistema. El Orquestador deberá definir una estructura básica de usuarios y roles de seguridad para ellos. Cada rol tendrá asociadas ciertas responsabilidades (como iniciar un proceso manual o continuar uno detenido en una acción de ingreso de datos de usuario). Al identificarse un usuario del Portal Web, el Orquestador le presentará sus opciones disponibles de acuerdo a los roles que ese usuario tenga asociados.

Al seleccionarse un disparo de señal manual, el Portal verificará si existen datos de entrada definidos para esa señal (según un Formato de Documentos) y generará en forma dinámica una página web para capturar esos datos de entrada. Se deberán soportar las características de registros compuestos (registros en donde algunos de sus atributos son otros registros) y de cardinalidades multivaluadas (algunos atributos ?compuestos o no- pueden ser colecciones de valores).

Los datos capturados en la interfaz web del portal serán transformados para integrarse al contexto del nuevo proceso o del proceso en ejecución. Se podrán definir las señales manuales como ?interactivas?, en cuyo caso se deberá indicar además una variable del contexto del proceso como un ?string con código HTML?, el que se mostrará como respuesta sincrónica a la acción del usuario operador (resultado de la operación que inicia o continua). Si la señal no se define como interactiva, el proceso que inicia o continúa se ejecuta como una hebra separada (en background) y no se informa al operador de su resultado.

Señal Periódica
Se podrán definir señales que se disparen en forma calendarizada de acuerdo a alguna parametrización, y que no aportarán datos de entrada. Estas señales deberán poder definirse para los diferentes días de la semana, rangos de horarios y meses.
Mensaje de Texto recibido en una Cola
El Orquestador permitirá asociar mensajes de una cola, interpretar su texto (xml) como los datos dentro de un Formato de Documento, y mapearlo al inicio o continuación de un proceso.

Señales disparadas por Procesos
Dentro de los diagramas de actividades UML 2.0 existen artefactos que representan el disparo explícito de señales. Se podrá modelar un proceso que dispare señales y se configurarán los datos que en ellas viajarán. Estas señales

podrán luego ser capturadas y mapeadas al inicio o configuración de nuevos procesos, sin afectar o acoplar su ejecución a los procesos que originalmente las dispararon.
Documentador de Procesos

En la Figura 3 se muestra la relación existente entre el módulo en el que se realizará la configuración y la carga de ésta como la?Versión Activa? dentro del servidor que ejecuta los procesos automáticos.

Requerimientos de SOA

Dentro de una arquitectura orientada a servicios, los diferentes sistemas de información de una organización publican y consumen servicios (típicamente en la forma de Web Services, a pesar de que el concepto es independiente de la tecnología). El Orquestador deberá actuar como intermediario entre las aplicaciones, de tal forma de evitar acoplamientos innecesarios entre ellas y que podrían llevar a mayores esfuerzos y costos en futuras mantenciones a esos sistemas.
Administrador de Buses de Servicios
Se propone que el Orquestador implemente y administre dos buses de servicios, uno para colocar aquellos servicios que serán usados (orquestados) desde dentro de los procesos que se modelan en la misma herramienta (Bus Privado o PSB), y el otro en donde se publicarán los servicios que se desean colocar como disponibles para otras aplicaciones, ya sea por Internet o desde la Intranet (Bus Empresarial o ESB).

En la figura 4 se muestra la relación existente entre el Bus Privado y el Bus Empresarial. Más adelante se detallan los tipos de servicios que se ofrecerán y su relación con los procesos (BPM como proveedor de nuevos servicios).

Buses de Servicios

Un servicio representa a una unidad de lógica (incluyendo acciones) de negocio reutilizable. Dependiendo de la ubicación de un servicio (ESB o PSB) éste será invocable desde fuera del Orquestador o no.

Los servicios privados pueden ser de alguno de los tipos básicos que más adelante se detallan. Cualquier servicio privado puede pasar a formar parte de un servicio público como una ?operación? dentro de un Web Service, para ser consumida desde los sistemas internos o desde la Internet por algún socio de negocios.

Los tipos básicos de servicios disponibles para ser publicados dentro del bus privado (y luego en el Empresarial, si así se desea) serán los siguientes:

Servicio JavaScript
Usando código JavaScript (compilado, con tiempos de ejecución comparables a código compilado Java) será posible programar lógica de negocio como servicios reutilizables y para ser invocados y coordinados dentro de procesos del Orquestador.

Mediante Formatos de Documento se describirán los argumentos de entrada y la estructura de la respuesta que un servicio JavaScript proveerá. Se proveerán además algunas clases utilitarias (?helpers?) con tareas comunes y que podrán ser utilizadas desde estos servicios JavaScript. Por ejemplo se proveerán helpers con funciones para manejo e archivos, para consultas sql y para manejo de tipos de datos como strings, fechas y números.

Servicio DBInsert
Este servicio sin datos de retorno, se podrá configurar para insertar un registro en una tabla en alguna base de dato visible desde el Servidor de Aplicaciones en donde está instalado el Portal (junto al motor de ejecución del Orquestador).

Los servicios de este tipo serán parametrizados usando un Formato de Documentos que identifique los registros de la tabla y posibles transformaciones durante su invocación. Si la inserción del registro falla (clave duplicada, error de constraint de base de datos, etc.) el proceso invocador dispara una excepción.

JMS Service
Servicio de envío de mensaje XML a una Cola (API JMS de J2EE). A partir de una estructura de datos definida por un Formato de Documentos, será posible configurar un servicio que envíe un mensaje de texto a una cola, en el que el contenido del mensaje se forma desde los datos de entrada al servicio (llenados por el contexto del proceso que lo invoque).

Invocador de Web Services
Se define un tipo de servicios especial al consumidor de Servicios Web, los que podrán ser externos a la organización o incluso publicados usando el mismo Orquestador. Usando un Formato de Documentos para identificar los argumentos de entrada y otro para la respuesta, se define una URL desde donde se debe invocar al servicio.

El Configurador del Orquestador deberá proveer la facilidad de crear los Formatos de Documento a partir de la Metadata que describe al Servicio Web apuntado, y que se provee según el estándar WSDL. De esta forma, quien publica el servicio dentro del Bus Privado, no debe conocer los detalles respecto a la invocación del Servicio Web, más allá de la dirección URL de su descriptor WSDL.

Screen scrapping / Web scrapping
Se proveerá de un servicio que recibirá los datos de entrada para configurar dinámicamente una dirección URL, cargará una página web (en memoria) y, utilizando expresiones regulares, extraerá la información desde el código HTML para retornarla como datos estructurados y definidos como un Formato de Documento que especifica la salida del servicio.

Proceso del Orquestador publicado como Servicio
Un proceso modelado y ejecutable por el Orquestador (BPM) tiene asociado un Contexto de Ejecución, que corresponde a un Formato de Documentos que identifica las variables visibles dentro de cada instancia de ese proceso.

Un tipo especial de servicio permite publicar en el PSB (y desde ahí al ESB, si se desea) un proceso modelado como un servicio, definiendo un Formato de Documento para sus entradas (y transformado en el contexto del proceso) y otro de salida (obtenido desde el contexto ya modificado por la lógica y acciones del proceso).

Este proceso publicado como servicio en el bus es invocable (orquestable) desde cualquier otro proceso, definiendo una estructura de jerarquía de procesos y subprocesos, desde el punto de vista del BPM o de servicios compuestos, desde el punto de vista de SOA.
Parametrización de la Carga del Servidor
El Servidor del Orquestador podrá ser utilizado para procesamiento de transacciones masivas, en donde la configuración de la utilización de los recursos (Conexiones a Bases de Datos, tiempos de uso de CPU, archivos, etc.) puede determinar la escalabildad de estas soluciones.

Para la invocación a los servicios dentro del Orquestador se podrá optar por su ejecución sincrónica (con datos de respuesta) o asincrónica (sin esperar respuesta). En este último caso, se podrá además configurar por cada servicio que se invoca desde un proceso, el tratamiento del paralelismo. En particular, se podrá configurar cuántas instancias del proceso que se invoca podrán estar en ejecución en paralelo, mientras el resto de las invocaciones quedan encoladas (y sus procesos invocadores se mantienen en memoria, esperando su turno de ejecución).

Para el resto de las configuraciones, como el tamaño de los ?pools? de conexiones a bases de datos o tiempos de espera para transacciones, el Orquestador se basará en las configuraciones del servidor de aplicaciones J2EE que se utilice como base para su ejecución.

var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
document.write("\\");

var pageTracker = _gat._getTracker("UA-3223641-1");
pageTracker._initData();
pageTracker._trackPageview();