

**Instituto de Computación (InCo)
Facultad de Ingeniería
Universidad de la República**

Proyecto de Grado:

B2B para Q-flow

Gustavo Degeronimi

Usuario Responsable:

Pablo García, Urudata

Supervisor:

Doc. Alejandro Gutiérrez

JULIO 2002

Abstract

El proyecto “B2B para Q-flow” estudia el problema de integrar un sistema de Workflow con aplicaciones empresariales. El mismo propone desarrollar un módulo para Q-flow, el sistema de Workflow de Urudata, que permita integrar procesos de workflow con los procesos ejecutados por el resto de las aplicaciones de la organización. La solución, basada en el **intercambio de documentos XML**, utiliza como intermediario a **BizTalk**, el nuevo servidor de mensajería XML de Microsoft.

1. INTRODUCCIÓN.....	5
2. CONCEPTOS PREVIOS	6
PROCESO DE NEGOCIO.....	6
WORKFLOW	6
SISTEMAS DE WORKFLOW.....	8
INTERCAMBIO ELECTRÓNICO DE DATOS.....	9
INTEGRACIÓN DE APLICACIONES EMPRESARIALES Y B2B	9
RESUMEN	10
3. HERRAMIENTAS	11
Q-FLOW	11
BIZTALK.....	14
4. OBJETIVOS ESPECÍFICOS	17
5. ANÁLISIS	18
CASOS DE USO	18
PAQUETES DE ANÁLISIS.....	19
PAQUETE AGENTE EXTERNO	20
PAQUETE QFLOW B2B_OUT.....	22
PAQUETE MIDDLEWARE.....	26
PAQUETE QFLOWB2B_IN.....	29
RESUMEN	30
6. DISEÑO.....	31
DESCRIPCIÓN GENERAL.....	31
PAQUETE QFLOW B2B.....	32
PAQUETE CONFIGURACIÓN BIZTALK.....	33
DIAGRAMA DE SECUENCIA.....	36
RESUMEN	37
7. IMPLEMENTACIÓN.....	38
PLATAFORMA DE SOFTWARE	38
VISTA DE COMPONENTES.....	38
DOCUMENTOS XML.....	39
PERSISTENCIA	42
MÓDULO QFLOW B2B	44
B2B CONFIG WIZARD.....	48
B2BGATE.....	50
8. TESTING.....	51
AGREE1	51
AGREE2.....	51
9. CONCLUSIONES	52
APÉNDICES	49
APÉNDICE A: XML	50
APÉNDICE B: WFMC. MODELO DE REFERENCIA.....	59
APÉNDICE C: BIZTALK SERVER.....	69
APÉNDICE D: Q-FLOW.....	80
BIBLIOGRAFÍA.....	89

1 Introducción

La tecnología de Workflow se ocupa de la automatización de procesos de negocio. Históricamente, la misma se ha enfocado en procesos que hacen uso intensivo de Recursos Humanos, logrando buenas soluciones al problema de la colaboración de tipo persona – persona.

Por otro lado, desde el desarrollo del comercio electrónico, las organizaciones se enfrentan a la necesidad de integrar aplicaciones originalmente aisladas. En un escenario de B2C¹, por ejemplo, se hace indispensable conectar los front-ends de una tienda virtual con los sistemas de producción, para automatizar el procesamiento de los grandes volúmenes de transacciones generados por el sitio. Esta necesidad ha impulsado el desarrollo de la tecnología de EAI (por Enterprise Application Integration), cuyo objetivo es ofrecer soluciones al problema de integrar aplicaciones empresariales.

Los recursos de una organización se componen tanto de sus Recursos Humanos como de sus Recursos Tecnológicos (las aplicaciones). Para obtener el máximo rendimiento, es necesario lograr una alta integración entre ambos, por lo que los sistemas de Workflow y las soluciones de EAI no pueden volverse “islas de automatización”[Ultimus], incomunicadas entre sí.

El proyecto “B2B para Q-flow” propone una solución al problema de la integración de un sistema de Workflow con aplicaciones empresariales. La propuesta se basa en el intercambio de documentos de negocio utilizando los servicios de mensajería XML de Microsoft BizTalk Server. El objetivo del proyecto es **desarrollar un módulo para Q-flow, el sistema de Workflow de Urudata, que permita integrar procesos del mismo con procesos ejecutados por aplicaciones externas**, ya sean domésticas o foráneas (B2B).

El documento comienza presentando el contexto tecnológico en que se desarrolla el proyecto. El capítulo 3 ofrece una introducción a la tecnología de Workflow y EAI, y el capítulo 4 presenta las herramientas que participarán en la solución: “Q-flow” y “Microsoft BizTalk Server”. El capítulo 5 detalla los objetivos del proyecto, que son luego analizados en el capítulo 6. El capítulo 7 describe el diseño de la solución desarrollada. Luego, el capítulo 8 describe la implementación y el capítulo 9 el testing realizado. Finalmente, se cierra el documento con las conclusiones, en el capítulo 10.

¹ B2C: “Business to Consumer”. Área del Comercio Electrónico en el que la contraparte en la transacción es una persona física (un consumidor).

2 Conceptos Previos

El proyecto “B2B para Q-flow” involucra básicamente dos tecnologías: “Workflow” y “EAI”(por “Enterprise Application Integration”). A continuación, se ofrece una introducción a las mismas.

2.1 Proceso de Negocio

Un proceso de negocio es un [ISO]:

*“Conjunto **estructurado** de tareas, que contribuyen **colectivamente** a lograr los objetivos de una organización”*

Algunos ejemplos de proceso de negocio son:

- Procesamiento de Ordenes de Compra
- Autorización de Horas Extra
- Gestión de Aprovisionamiento (Proceso de Compra)
- Procesamiento de Notas de Crédito
- Autorización de Viáticos
- Gestión de Proyectos

La definición de procesos de una organización forman parte de la cultura de la misma. Estos se describen en Manuales de Procedimiento, en forma de diagramas de flujo o directamente en lenguaje natural.

Los procesos de negocio son la base operativa de una empresa, y el éxito de la misma depende de la eficiencia con que sean gestionados. Una empresa con una mala gestión de procesos incurrirá en costos excesivos, baja productividad de recursos y altos tiempos de respuesta.

A diferencia de los procesos industriales, la entrada, salida y productos intermedios de un proceso de negocio es información. La misma se estructura habitualmente en documentos de negocio, como facturas, boletas y ordenes de trabajo. Así mismo, es común que los propios procesos se desarrollan entorno a este documento: “Procesar” una Orden de Compra, “Procesar” una Factura, una liquidación de viáticos, un pedido de horas extra, etc.

2.2 Workflow²

En un proceso de negocio tradicional, la gestión del proceso esta distribuida entre sus participantes. Cada uno de ellos es responsable de gestionar el proyecto entorno a sus tareas; es responsable de decidir que tarea debe ejecutar, que tarea le sigue a la suya y quien esta encargado de la misma. Estas decisiones son tomadas interpretando libremente las reglas de la organización, por lo que los procesos corren un gran riesgo de desviarse, generando costos y retrasos innecesarios.

² El término Workflow surge de observar como en un proceso automatizado, la culminación de una tarea genera instantáneamente trabajo pendiente para las tareas que la siguen, lo cual puede verse como que el trabajo "fluye" a lo largo de las tareas y los participantes del proceso, conformando un "flujo de trabajo" (un Work-flow) que recorre el proceso hasta su fin. Los términos Proceso de Negocio y Workflow son utilizados como sinónimos en la mayoría de los casos, aunque intentaremos utilizar el segundo como la versión automatizada del primero.

La automatización de procesos intenta resolver estos problemas, sistematizando la forma en que los procesos son gestionados³. En un proceso automatizado, cada decisión acerca del flujo del proceso es tomada de forma determinística, evaluando el conjunto de reglas que define el proceso.

La tecnología de Workflow se ocupa de la automatización de procesos de negocio. Según la WfMC (Workflow Management Coalition), un Workflow es:

“La automatización total o parcial de un proceso de negocio, en el cual documentos, información o tareas son pasadas de un participante a otro, de acuerdo a un conjunto de reglas procedurales”

Aunque un sistema de Workflow puede ser implementado manualmente, mediante formularios de papel que viajan físicamente entre los participantes, desarrollan su máximo potencial sobre medios electrónicos.

2.2.1 Modelo

Describiremos los aspectos que componen un Workflow basándonos en el modelo elaborado por la WfMC. Según la misma, la definición de un Workflow se compone de:

- Las tareas y la **secuencia** en que serán ejecutadas (las reglas de flujo). Se representan como diagrama de flujo.
- La asignación de **recursos** a cada tarea (quien hace qué),
- La **información** que debe estar disponible y ser producida en cada etapa, y
- Las **herramientas** que podrán ser utilizadas para ejecutar cada tarea.

³ Notar la diferencia entre el proceso y la gestión del mismo. El objetivo es automatizar la gestión de procesos, no las tareas que lo constituyen, las cuales podrán o no estarlo.

2.3 Sistemas de Workflow

Según la WfMC, un “Sistema de Workflow” es un:

“Sistema capaz de **definir, administrar y ejecutar** Workflows, a través de **software** que interpreta una **representación electrónica** del mismo e interactúa con los recursos participantes”.

Un sistema de Workflow ofrece una estructura tecnológica sobre la cual implementar, de forma eficiente, las políticas de administración de procesos de una organización. Tradicionalmente, las implementaciones de Workflow han utilizado el papel como transporte de la información. Boletas, facturas y formularios se mueven de bandeja de salida a bandeja de entrada cada vez que es terminada una tarea. El tiempo que implica este desplazamiento es llamado “tiempo en bandeja” o “tiempo de transferencia”, y puede llegar a consumir el 90% del tiempo total del proceso, mientras que solo el 10% restante es aportado por el “tiempo de tarea”. El uso de medios electrónicos permite a los sistemas de Workflow reducir significativamente el tiempo en bandeja, atacando directamente la causa mayor de retraso en los procesos.

Los sistemas de Workflow pertenecen típicamente al área de operaciones (B.O.) de las empresas, pero están fuertemente relacionados con áreas de Administración del Conocimiento (KM). Los mismos generan gran cantidad de información acerca de la operativa, y son un excelente medio de impactar las “lecciones aprendidas” en el negocio. Es por esto que es común ver soluciones de Workflow como resultado de reingenierías de proceso y como base de sistemas de mejora continua. [QFlow]

La implantación de un sistema de Workflow ofrece beneficios para todos sus participantes [QFlow]:

- **Cliente:** Aumenta la calidad del servicio ofrecido. Mejora los tiempos de respuesta y permite ofrecer al cliente información acerca del progreso de su solicitud.
- **Trabajador:** Permite una mejor organización de las tareas pendientes de cada trabajador y le provee toda la información que necesita para su actividad.
- **Supervisor:** Ofrece información para la toma de decisiones en tiempo y forma. Ofrece mecanismos para monitorear el desempeño de los supervisados y para distribuir la carga de trabajo entre los mismos.
- **Analista de procesos:** El sistema le provee de información estadística, costos, picos de trabajo, etc.
- **Empresas:** Aumenta el control sobre las tareas y refuerza la confidencialidad ofreciendo esquemas de control de acceso a la información.

Los sistemas de Workflow son utilizados por gran variedad de industrias: Bancos, Aseguradoras, Estudios Jurídicos y Entes Estatales, entre otros. Cada solución es distinta a las demás en aspectos como capacidad de almacenamiento, infraestructura de comunicaciones, plataforma de software disponible y estrategia de interacción con los recursos. Es por esto que resulta imposible desarrollar un sistema de Workflow genérico. Mas bien la tendencia de los desarrolladores es concentrarse en un conjunto de características y construir productos que se ajustan correctamente a las necesidades de una fracción del mercado[WfMC].

2.4 Intercambio Electrónico de Datos (EDI)

Las normas de EDI (por Electronic Data Interchange) son un conjunto de estándares que controlan el intercambio electrónico de documentos de negocio (Ordenes de compra, facturas, etc.). Las organizaciones pueden apoyarse en estas normas para asegurar que los documentos que generan serán interpretados correctamente sin importar la plataforma y la aplicación con la que estén interactuando.

La tecnología de EDI permite mover documentos de negocio de un sistema a otro sin intervención humana, por lo que resulta más rápida, económica, y segura que el intercambio de documentos en papel.

La estandarización de EDI comienza en los 60, impulsada por la industria del transporte, expandiéndose posteriormente a otras. Luego, la necesidad de uniformizar los distintos estándares surgidos dentro de cada industria fomentó la creación de la Accredited Standards Committee X12, en 1979. Este comité creó el estándar de EDI conocido como **X12**, que fue utilizado principalmente para comercio interno dentro de EE.UU.

Mientras tanto, la Comunidad Europea desarrolla su propio estándar, llamado **GTDI** (Guidelines on Trade Data Interchange). Luego, tomando partes de X12 y GTDI, las Naciones Unidas desarrollan EDIFACT (Electronic Data Interchange For Administration, Commerce, and Transport). En 1987, la ISO adopta EDIFACT, y en el 92 también es adoptada por la ASC X12.

Por más que la tecnología de EDI existe desde hace más de 40 años, no ha fomentado en gran medida al comercio B2B. El número de intercambios comerciales sobre EDI sigue siendo limitado en relación a los realizados por teléfono o fax. Entre las razones de esto se encuentra que [Biztalk]:

- Las aplicaciones servidor de EDI son habitualmente costosas.
- El documento EDI es bastante críptico (en parte causal de lo primero).
- Las redes de transporte (comunicación) de EDI han sido históricamente redes privadas de valor agregado (V.A.Ns), de uso costoso para las empresas.

El desarrollo de las tecnologías de Internet ha hecho cambiar esta perspectiva del intercambio de documentos. El uso de XML permite desarrollar aplicaciones de EDI de forma económica, y disponer de una vasta red pública como es Internet, sobre la cual las empresas ya han hecho grandes inversiones para sus sitios de B2C, permite evitar los costos asociados al uso de las antiguas redes de EDI.

2.5 Integración de Aplicaciones Empresariales (EAI) y B2B

2.5.1 Integración EAI

En los últimos años, el desarrollo de aplicaciones empresariales ha sido pauta por la creciente especialización de los productos. Los mismos se concentran en ofrecer soluciones a problemas de negocio específicos como “Enterprise Resource Planning”(ERP)⁴, “Customer Relationship Management” (CRM)⁵, “Sales Force Automation”(SFA)⁶, Sistemas Financiero / Contables, etc.

Generalmente, estas aplicaciones son “cerradas”, y proveen poca funcionalidad para interactuar con el resto del entorno. En muchos casos, han evolucionado de antiguos sistemas confinados funcional o departamentalmente, donde no existían razones para interactuar con otras aplicaciones.

Este enfoque de aplicaciones empresariales genera “islas de automatización” dentro de las empresas; zonas de la operativa altamente automatizadas con pobres lazos de colaboración con el exterior, que obligan al personal a funcionar de interfase entre sistemas, reingresando información generada por un sistema en otro[ULT_EAI].

⁴ Administración de los recursos de una organización

⁵ Automatización de Help Desk, Soporte técnico, Call Centers, etc.

⁶ Mejorar la productividad de las fuerzas de venta (Vendedores, Visitadores, Distribuidores y Representantes.)

La tecnología de EAI (por Enterprise Application Integration) surge de la necesidad de integrar nuevamente estas “islas de automatización”. La misma se enfoca en el movimiento de información entre sistemas heterogéneos como parte de un proceso de negocio[ULT_B2B].

Varios factores han contribuido a la demandada de EAI [ULT_EAI]:

- El crecimiento del comercio electrónico presiona a las empresas a operar en “tiempos de Internet”, lo cual solo puede conseguirse mediante soluciones integradas.
- La estructura de las organizaciones ha cambiado del modelo funcional , bien soportado por soluciones específicas y cerradas en su frontera, por modelos horizontales, que demandan acceso a los sistemas desde fuera del área funcional al que pertenecen.
- Cuando una empresa adquiere otra, recibe una gran inversión en aplicaciones habitualmente incompatibles con las propias. Las soluciones de EAI permiten salvar este valor al integrar las mismas.
- Las empresas conservan información en sistemas legado, la cual es necesario integrar a nuevos desarrollos.
- El avance en la tecnología de middleware desde el intercambio de archivos ASCII, pasando DCOM y CORBA, hasta XML el día de hoy ha permitido mejorar la calidad de las soluciones de EAI.

2.5.2 Integración B2B

Business To Business (B2B) es el área dentro del comercio electrónico donde ambas contrapartes en la transacción son empresas (proveedores, clientes o partners). Las soluciones de B2B se centran en mejorar las comunicaciones entre las partes, haciendo más efectiva su interacción, habilitando la oferta de nuevos servicios y nuevas formas de negocio.

Las soluciones de B2B son casos especiales de EAI, en donde:

- Las aplicaciones a integrar son controladas por organizaciones distintas,
- La relación entre sistemas es habitualmente "uno a muchos"
- La comunicación es hecha sobre infraestructura pública (Internet)

Ejemplos son sistemas de aprovisionamiento automatizado, soporte a distribuidores, etc.[E-Business]

2.5.3 Workflow en el contexto de integración

Los sistemas de Workflow son potentes herramientas de integración, capaces de manejar la relación entre los distintos recursos de la organización. A diferencia de otras aplicaciones, los sistemas de Workflow no se circunscriben a un área funcional de la empresa. Los procesos de Workflow describen un corte horizontal de la organización y dan una visión global de la operativa, lo cual permite identificar en éstos los requerimientos de integración de cada subsistema.

Por estas razones, los sistemas de Workflow son excelentes guías para planear, diseñar e implementar integración hacia adentro (EAI) y hacia afuera (B2B) de las aplicaciones de nuestra empresa.

2.6 Resumen

La tecnología de Workflow se ocupa de la automatización de procesos de negocio, enfocándose típicamente en procesos de tipo persona – persona. Un sistema de Workflow, como cualquier otra aplicación empresarial, corre el riesgo de convertirse en una “isla de automatización”, resultando en procesos de tipo persona-persona altamente automatizados, pero totalmente aislados del resto de los procesos (basados en aplicaciones) de la organización. El movimiento de información desde y hacia el resto de las aplicaciones es realizado por el personal, lo cual es intolerable para los tiempos de respuesta y volúmenes de carga que son habituales en Internet. (Por ejemplo, en Comercio Electrónico).

La tecnología de EAI se encarga de ofrecer soluciones a este problema, que permitan construir “puentes” de automatización necesarios para integrar nuevamente las “islas”.

3 Herramientas

El proyecto se desarrolla entorno a dos productos: “Q-flow” y “Microsoft BizTalk Server”. Q-flow, el sistema de Workflow de Urudata, es el producto central del proyecto, ya que el objetivo del mismo es extender las funcionalidades del producto para resolver el problema planteado. Microsoft BizTalk Server es el servidor de EAI de Microsoft, y es parte principal de la solución propuesta. El capítulo siguiente hace una introducción a ambos productos.

3.1 Q-flow 2.0

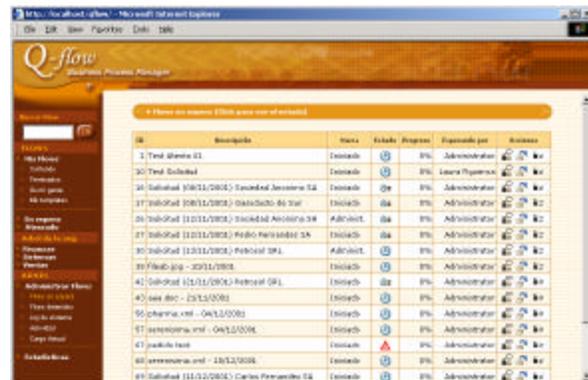
Q-flow es un conjunto de herramientas desarrollado por Urudata, que permite modelar procesos de negocio en forma electrónica. El paquete esta compuesto por:

- El motor de Workflow
- El sitio Web de la aplicación
- Herramientas de diseño de procesos y formularios
- Herramientas de administración, configuración y mantenimiento
- Herramientas de estadística e integración con sistemas de información gerencial (OLAP).

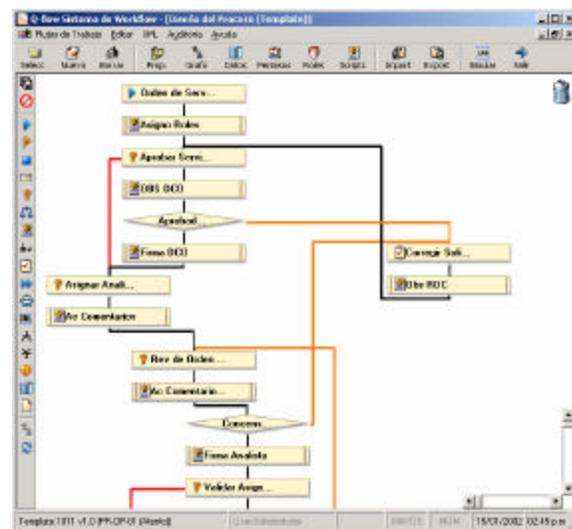
3.1.1 Características

[Qpaper]

- **Arquitectura:**
 - o Interacción con Recursos Humanos mediante Mensajería y Web, permite llegar en “tiempo cero” a cada uno de los puestos de trabajo de la organización. Las tareas son enviadas proactivamente a la casilla del usuario utilizando correo con contenido Web y el mismo podrá contestar accediendo al sitio o desde el propio contenido del correo
 - o Transaccionalidad de las operaciones, lograda almacenando la información del workflow en bases de datos relacionales y no sobre los sistemas de correo.
 - o Almacenamiento de Archivos Adjuntos sobre el sistema de archivos local, Web Storage, Sharepoint Portal Server, o cualquier otro almacenamiento de documentos que soporte WebDAV⁷. Políticas de control de versiones tipo “check-in” / “check-out” disponibles para todas las opciones.



- **Funcionalidades:**
 - o Roles separados para el analista de procesos (Diseñador) y para los técnicos de Informática (Administrador).
 - o Diseño gráfico de procesos: 17 tipos de paso, entre los que se cuenta: Inicio, Fin, Hito, Tarea, Pregunta, Notificación, Evaluación, Invocación a Subproceso, Fork, Join, BOT (para pasos externos) y Script.



Manejo de Datos de Aplicación, Roles propios del proceso y Archivos Adjuntos.

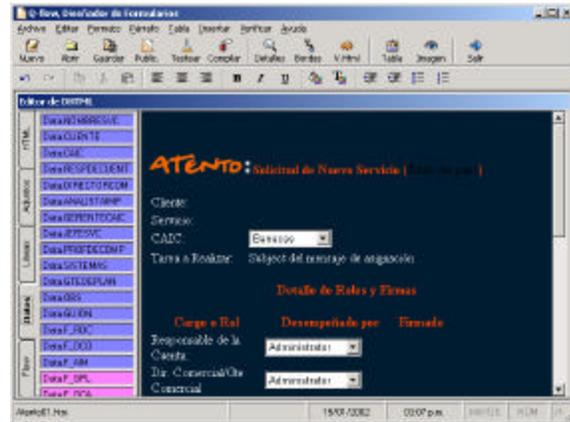
Capacidad para alterar el diseño de los procesos “en caliente” generando nuevas

⁷ WebDAV: “Web Distributed Authoring and Versioning”, es una extensión del protocolo HTTP 1.1 que habilita la publicación y edición de contenido en servidores Web.

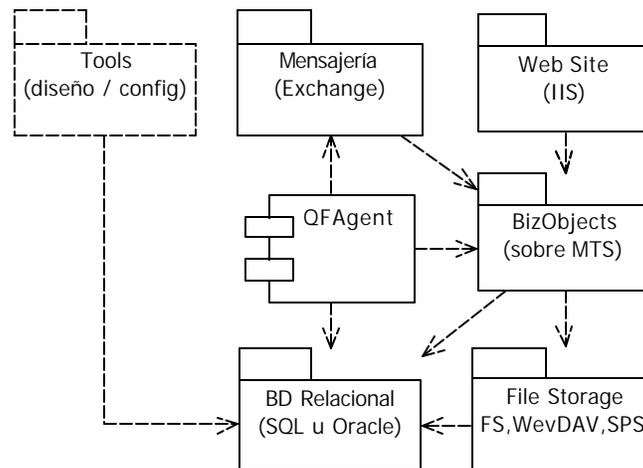
versiones y / o impactando en los procesos que están corriendo.

Manejo de delegaciones automáticas, recordatorios y alertas, basados en tiempos netos de trabajo, en fechas fijas o en tiempos o fechas especificados por los usuarios o calculados en un paso anterior.

- o Diseño gráfico de formularios: Integrado al diseño de procesos: Permite incorporar los datos de aplicación del proceso como campos del formulario haciendo “drag-and-drop”.
- o Capacidad de asignar un dominio a los datos de aplicación, de forma de que se ingresen al sistema datos validados, ya sea en formato o contra los sistemas de información de la empresa. Un Dominio permite definir también el control gráfico utilizado para desplegar el campo en los formularios (textbox, drop-down, checkbox, etc.)
- o Manejo de organigrama con capacidad de sincronizarlo con un servicio de directorio u otra fuente externa. Se soportan roles relativos. (“El superior de..., los subordinados de...”).
- o Auditoria: Soporte de auditoría de cambios organizacionales, cambios en la definición de los procesos, operación del sistema (delegaciones, re-asignaciones, confirmaciones de lectura, etc.), tracking detallado de las variables y datos adjuntos, permitiendo mantener las versiones intermedias de los mismos.
- o Estadísticas: Estadísticas operativas para el administrador de procesos o el manager de usuarios. Generador de cubos OLAP para análisis detallado de los procesos.
- **Integración:**
 - o Integración con el proveedor de seguridad de la empresa: Microsoft Active Directory (W2K), Dominios de Windows NT 4, Novell Directory Service (NDS o Binderly), Proveedores de autenticación Radius (Unix, IAS, etc.). Sincronización de usuarios y manejo de “single signon”
 - o Capacidad de interactuar con sistemas existentes en la empresa:
 - Invocando componentes COM, Procedimientos Almacenados, transacciones CICS (vía COMTi y Host Integration Server).
 - Tomando datos de bases de datos relacionales de la empresa o vía Base de Conocimiento de Genexus.
- **Extensibilidad:**
 - o Capacidad de definir pasos externos y utilizarlos como parte del sistema.
 - o Capacidad de diseñar digital dashboards customizadas para los usuarios.
 - o Capacidad de iniciar Workflows desde aplicaciones externas o páginas web.
 - o Capacidad de responder a pasos desde aplicaciones externas.
 - o Completa customización de la interfase web del sitio y de cada paso de tarea.



3.1.2 Arquitectura



La arquitectura de Q-flow se organiza en tres capas:

- **Presentación:**
 - **Web/Mail:** La interacción con los usuarios es hecha mediante Mensajería y Web. Las tareas son enviadas proactivamente a la casilla del usuario usando correo con contenido Web y podrán contestarse accediendo al sitio o desde el propio contenido del correo.
 - **Herramientas de Diseño y Administración:** Las herramientas de diseño y administración son clientes “pesados” que interactúan directamente con la base de datos. Las mismas permiten intercambiar las definiciones con otras instalaciones de Q-flow importando / exportando archivos XML.
- **Middle Tier** compuesta por:
 - **Agente Q-flow:** El agente Q-flow es la parte proactiva del sistema. El mismo tiene una arquitectura de motor / intérprete, sobre los datos(flows) y metadatos(diseños, definiciones, etc.) almacenados en la base relacional. El mismo recorre periódicamente los procesos en ejecución, identificando los pasos finalizados, y activando los pasos siguientes según el orden de ejecución establecido por el Template.
 - **BizObjects:** Implementan los objetos de la aplicación, los cuales encapsulan la lógica del sistema.
- **Data Tier:**
 - **Base relacional:** A diferencia de otros sistemas de Workflow, Q-flow mantiene tanto la definición de los procesos (los templates) como la información de las instancias (los flows) sobre una base relacional, almacenando de forma no estructurada solamente los archivos adjuntos.
 - **File Storage:** El almacenamiento de los archivos adjuntos puede ser realizada sobre el sistema de archivos local, Web Storage, Sharepoint Portal Server, o cualquier otro almacenamiento de documentos que soporte WebDAV⁸. Para todos los casos el sistema implementa políticas de control de versiones tipo “check-in” / “check-out”.

⁸ WebDAV: “Web Distributed Authoring and Versioning”, es una extensión del protocolo HTTP 1.1 que habilita la publicación y edición de contenido en servidores Web.

3.2 BizTalk

BizTalk es una iniciativa promovida y desarrollada por Microsoft, con el objetivo de resolver el **intercambio de documentos de negocio en un ambiente de e-commerce**[Travis].

Utilizar XML para el diseño de documentos de negocio asegura que el destinatario será capaz de validar los documentos que reciba y recuperar los datos contenidos. Sin embargo, reconocer los datos dentro de un documento no significa poder darles un significado. Sin conocer el formato y la semántica, el destinatario no puede convertir los **datos** del documento en **información** procesable. Por esto, cuando dos organizaciones se proponen interactuar de forma electrónica, deben firmar previamente un Acuerdo de EDI (“EDI Agreement”) que defina, entre otros, el formato de los documentos intercambiados.

En un escenario de B2B, (donde estaremos interactuando con muchas organizaciones), la administración de estos Acuerdos de EDI individuales resulta altamente costosa e ineficiente. Potencialmente, cada acuerdo utilizará esquemas de documento distintos, aun para transmitir la misma información. Por ejemplo, pensemos en dos organizaciones con las cuales intercambiamos “Ordenes de Compra”. Como ambas diseñan sus esquemas de forma descoordinada, la primera describe los ítems de la orden como

```
<item cant=10>2332</item>
```

y la otra

```
<item>
  <cant>10</cant>
  <cod>2332</cod>
</item>
```

Por esta razón, resulta necesario definir un conjunto de estándares y reglas (un “Framework”) que coordinen y asista el diseño de documentos de negocio.

3.2.1 BizTalk Framework

La **BizTalk Framework** provee un conjunto de reglas y de tags (los “BizTags”) para diseñar esquemas de documento para transacciones de tipo Business-to-Business. La misma impulsa el uso de XML, simplificando el diseño de esquemas que puedan ser compartidos por varias organizaciones[Travis].

La BizTalk Framework agiliza la adopción de EDI en una gran cantidad de industrias que ya utilizan XML. Como consecuencia, permite alcanzar una masa crítica de acuerdos establecidos consistentemente, que resulta un blanco para el desarrollo de nuevas herramientas de e-commerce y EAI⁹ [Travis].

La BizTalk Framework se compone de:

- **Especificación de Documento y Mensaje BizTalk:** Un documento BizTalk es un mensaje SOAP 1.1, donde el Body contiene uno o varios documentos de negocio, y el Header contiene entradas específicas de BizTalk (llamadas “BizTags”) que definen la semántica de manipulación de los documentos.
- **BizTalk.org:** Sitio Web que ofrece un repositorio de esquemas donde publicar los esquemas propios de cada industria y organización.
- **Especificación BizTalk Framework Compliant (BFC) Server:** Define las características de un sistema capaz de leer e interpretar documentos BizTalk.

Disponer de esta plataforma, permite que los desarrolladores se enfoquen en resolver los requerimientos de información de cada industria y eviten los detalles comunes a todas.

⁹ Este es, finalmente, el objetivo de las empresas detrás de la iniciativa BizTalk.

3.2.2 Microsoft BizTalk Server

Microsoft BizTalk Server 2000 es uno de los componentes de la plataforma de servidores empresariales .NET de Microsoft. El producto propone una solución al problema de integrar aplicaciones y datos, permitiendo a las empresas integrar y automatizar sus procesos de negocio. [BTCurso]

Microsoft BizTalk Server provee una plataforma confiable para el intercambio de documentos y la integración de procesos de negocio. El mismo soporta una gran cantidad de formatos de documento y protocolos, lo que permite a las organizaciones implementar nuevas tecnologías reutilizando el software existente.

BizTalk Server 2000 incluye un conjunto de herramientas que asisten en la implementación de canales de intercambio de documentos de negocio que cumplan (o no) las especificaciones de la BizTalk Framework. En particular, nos asisten en:

- Crear especificaciones de documento,
- Definir transformaciones de una especificación de documentos en otra (Mapeos¹⁰, o "Mappings"),
- Integrar aplicaciones, vía diseño y ejecución de "Schedules", y
- Monitoreo y log de documentos en tiempo de ejecución.

Los servicios BizTalk se dividen en dos áreas: **Mensajería** y **Orquestación**. El servidor de Mensajería funciona como intermediario en el envío y recepción de documentos. Ofrece una interfaz independiente del protocolo y esquema de datos, y agrega servicios de entrega confiable, seguridad, etc.

El servidor de Orquestación se encarga de ejecutar los "schedules" que secuencian la actividad de las aplicaciones que participan en un proceso de negocio. Los "Schedules" modelan un proceso de negocio como un diagrama de flujo, donde cada paso representa el intercambio de información (vía el servicio de mensajería u otro medio) entre el scheduler y una aplicación. El servicio está especialmente diseñado para ejecutar procesos largos ("long term"), como suelen ser los procesos de negocio.

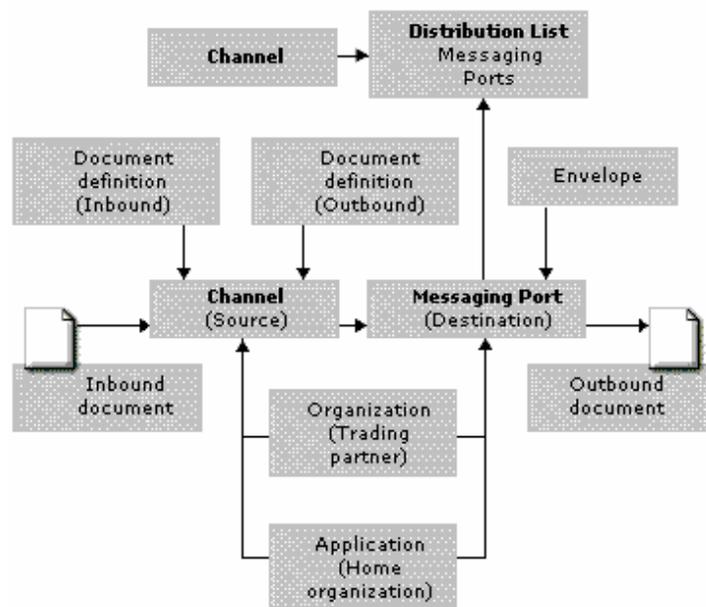
Servicios de Mensajería

El servicio de mensajería de BizTalk se ocupa de **enviar y recibir documentos**. Sus tareas incluyen el mapeo de esquemas, generación y correlación de confirmaciones (receipts) para entrega confiable, control de integridad y seguridad.

El servicio de mensajería se configura mediante un conjunto de "objetos de configuración". Los mismos definen la forma en que son procesados los documentos que pasan por el servidor. El esquema de la derecha describe la relación entre los mismos.

Canal: Representan las reglas usadas para procesar los documentos que llegan al servidor. Cada canal especifica:

- Origen: Puede ser una aplicación o una organización.
- Esquemas de Documento de entrada y salida.
- Mapeo utilizado para convertir el documento entrante en el saliente.



¹⁰ Se entiende como "Mapeo" a una "correspondencia" establecida entre dos esquemas de documento.

Es obligatorio si los documentos son distintos.

- Información de registro y seguimiento.

Puerto: Define la forma en que son transportados los documentos a su destino. El destino podrá ser una organización externa o una de nuestras aplicaciones. Cada puerto especifica:

- Destino: Una organización, una aplicación, o un schedule XLANG.
- Transporte: HTTP, HTTPS, SMTP, Loopback o AIC (Application Integration Components, desarrollados “a medida” del sistema de destino)
- Envelope: El envelope que se utilizará para encapsular los documentos antes de transportarlos.
- Seguridad: Se podrán especificar certificados de encriptación para utilizar con HTTPS y las extensiones s/MIME

Organizaciones: Representan las organizaciones con las cuales existen relaciones de EDI.

Aplicaciones: Representan los sistemas de nuestra organización que soportan los acuerdos de B2B.

Definiciones de Documento: Representa un tipo de documento. La misma contiene:

- Una referencia a una especificación de documento creada en BizTalk Editor. El archivo de especificación podrá estar almacenado en un espacio WebDAV¹¹ o en cualquier otra ubicación accesible por BizTalk y por la organización externa a la vez.
- Parámetros de registro y seguimiento del tipo de documento.

Envelopes: Consiste de un header y un footer que se agregan a cada documento antes de ser enviado. El mismo contiene información útil para el ruteo y procesamiento del documento en el servidor de destino.

Listas de Distribución: Son conjuntos de puertos que pueden utilizarse para enviar el mismo documento a un conjunto de destinatarios

¹¹ WebDAV es una extensión de HTTP que permite editar y administrar archivos en un servidor Web.

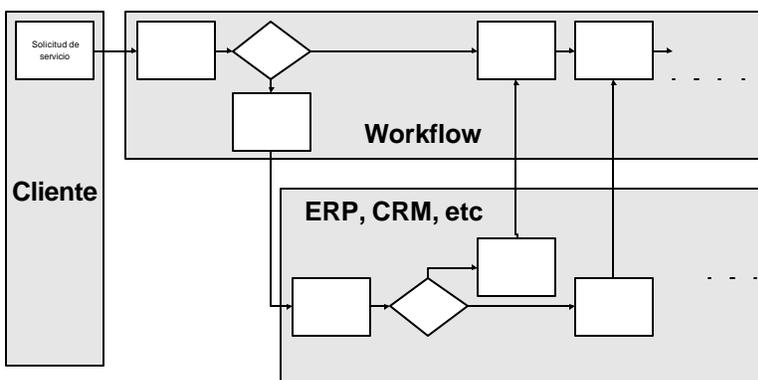
4 Objetivos Específicos

El objetivo del proyecto, es desarrollar un módulo que permita integrar a Q-flow con el resto de las aplicaciones empresariales de la organización. El mismo, permitirá concatenar procesos de Workflow con procesos ejecutados por otros sistemas, automatizando la interacción entre los mismos. El sistema será capaz de actuar como cliente, iniciando procesos externos, tanto como servidor, habilitando el inicio de procesos locales desde el exterior.

La comunicación entre sistemas será realizada a través del intercambio de documentos XML sobre protocolos de Internet. Se habilitará el uso de especificaciones de la BizTalk Framework para el intercambio de documentos de negocio.

El módulo será utilizado en dos posibles escenarios:

- **EAI:** Integración con Aplicaciones Empresariales. Automatizar la interacción entre procesos de Workflow y procesos ejecutados por el soporte informático de la organización.
- **B2B:** Integración con los sistemas de clientes, proveedores y partners.



4.1.1 Integración con Aplicaciones Empresariales

Los procesos de negocio involucran tanto a recursos humanos como tecnológicos de la organización. En los mismos, las aplicaciones de negocio colaboran estrechamente con el personal. La toma de decisiones y el trabajo crítico es realizado por los recursos humanos, y la parte mecánica del proceso es delegada a las aplicaciones de negocio. Por ejemplo: El procesamiento de una Nota de Crédito puede dividirse en dos etapas: Autorizar y Generar la nota. La primera parte es ejecutada típicamente por RR.HH; implica negociación, no existen reglas generales, se aceptan excepciones y la decisión final depende de una gran cantidad de factores. En cambio, la segunda parte es sistemática, por lo que es ejecutada en el ERP¹²: Se asienta la Nota, se ajusta la cuenta del cliente y se imprime el documento en papel.

En la actualidad, Q-flow permite automatizar el trabajo de Recursos Humanos, pero no es capaz de modelar la relación de estos con aplicaciones empresariales. Las soluciones que lo requieren toman dos estrategias:

- **No integrada:** La conexión entre el Workflow y los otros sistemas no se automatiza. El intercambio de información es hecho a través del usuario, el cual debe reingresar información a partir de listados.
- **Integración a medida:** Se desarrollan módulos de integración a medida para cada solución. Los mismos acceden externamente a la lógica interna del producto, y resultan soluciones poco elegantes y difíciles de mantener.

El módulo EAI / B2B permitirá modelar la interacción de los flows con los procesos que corren las aplicaciones empresariales de la organización.

4.1.2 Integración con Clientes, Proveedores y Partners

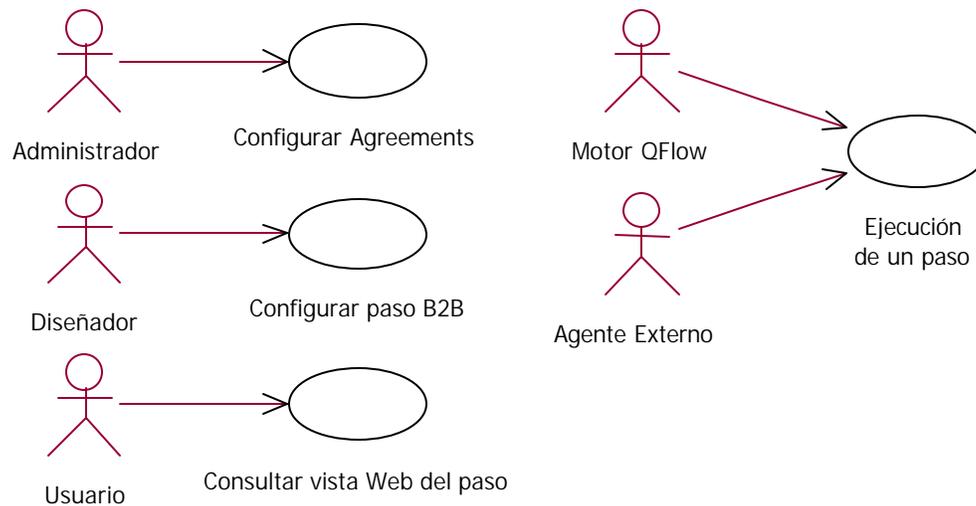
El segundo escenario prevé la integración de procesos con sistemas de otras organizaciones en el marco de acuerdos de B2B. De la misma forma en que se podrá interactuar con aplicaciones locales, el módulo permitirá interactuar con sistemas foráneos, habilitando la invocación de procesos externos y el inicio de flujos desde el exterior.

¹² ERP: "Enterprise Resource Planning"

5 Análisis

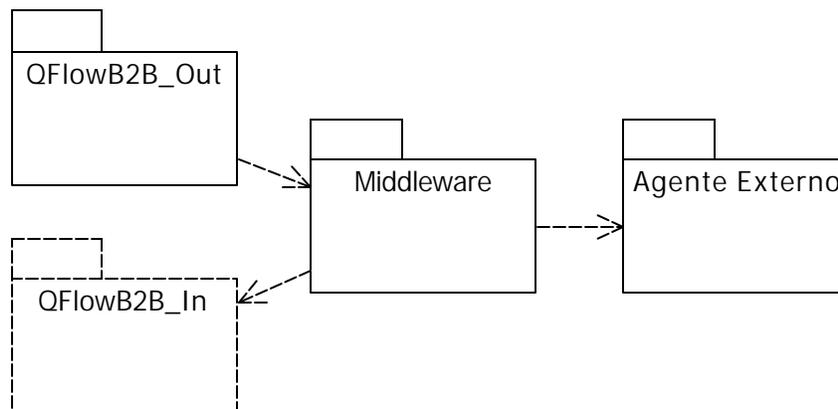
El capítulo siguiente estudia la naturaleza del problema planteado. Se identifican los aspectos que deberán ser resueltos para alcanzar los objetivos. Se presentan los actores del sistema y se definen los paquetes de análisis, los cuales organizarán el trabajo de aquí en adelante. Cada paquete analiza el problema desde un punto de vista distinto: Del sistema de Workflow, de las aplicaciones externas, y de la comunicación entre ambas.

5.1 Casos de Uso



- **Actores:**
 - **Administrador:** El Administrador del sistema Q-flow. Define los aspectos técnicos asociados a la interacción con aplicaciones externas. Tiene una visión “caja blanca” del paso B2B.
 - **Diseñador:** Es el analista de procesos. Es el encargado del diseño de los flujos. Tiene una visión “caja negra” del paso B2B.
 - **Usuario:** Cualquier participante del sistema.
 - **Motor Q-flow:** El motor del sistema de Workflow.
 - **Agente Externo:** Las aplicaciones integradas al sistema.
- **Casos de Uso**
 - **Configurar Agreements:** Incluye altas, bajas y modificaciones de los acuerdos de EDI (los “Agreements”) instalados.
 - **Configurar paso B2B:** Durante el diseño de un flow, se asocia un paso de B2B a un Agreement previamente instalado.
 - **Consultar vista Web del paso:** Un usuario consulta desde el sitio Web la información de tiempo de ejecución del paso.
 - **Ejecución de un paso:** El control de un flujo llega a un paso B2B y el motor despacha la ejecución del mismo. El Agente Externo recibe la solicitud, la procesa y retorna los resultados.

5.2 Paquetes de Análisis



El análisis del problema se divide en cuatro paquetes: “QflowB2B_In”, “QflowB2B_Out”, “Middleware” y “Agente Externo”.

Los paquetes QflowB2B_Out y QflowB2B_In estudian el problema desde el punto de vista de Q-flow como sistema; el primero como “cliente”, y el segundo como “servidor”. Se estudia la integración con aplicaciones externas a nivel conceptual: ¿Qué significa “integrar a Q-flow con sistemas foráneos”, y como se modela esta interacción?

El paquete “Middleware” analiza el intercambio de documentos XML como estrategia de Middleware. Se describen los aspectos técnicos involucrados: Manejo de múltiples esquemas y protocolos de transporte, entrega confiable, seguridad, etc. Se estudian los pros y contras de utilizar BizTalk, comparando la estrategia propuesta contra la solución adoptada por el producto “Ultimus Workflow Software”.

Por último, el paquete Agente Externo describe las características de los sistemas con los cuales estaremos interactuando y las interfases que proveen.

5.3 Paquete Agente Externo

Llamaremos “Agente Externo” a los sistemas con los cuales seremos capaces de interactuar. Dada la diversidad existente, no existe una única estrategia de integración que permita adaptarse a todos, por lo que es necesario enfocarse en un conjunto particular de sistemas

La definición de “Agente Externo” debe obtener **la mejor relación entre variedad de sistemas integrables y complejidad de la solución**. Para esto, estudiaremos las características comunes a los sistemas con los cuales típicamente nos interesaría colaborar y nos enfocaremos en estos. El concepto de “Agente Externo” cubrirá tanto las características de las aplicaciones empresariales, como de Sistemas Foráneos (contexto de B2B), lo cual permitirá unificar los dos escenarios planteados en los objetivos.

5.3.1 Orientación a Procesos

Un Agente Externo es un sistema típicamente **orientado a procesos**. Los servicios que ofrece son del tipo “Procesar una factura”, “Emitir una orden de compra”, etc. Quedan afuera de la clasificación, por ejemplo, consultas de información en tiempo real y transacciones en línea, para los cuales Q-flow ya dispone de soluciones adecuadas.

5.3.2 Procesos “Largos”

A diferencia de los sistemas transaccionales habituales, donde los resultados de una consulta son obtenidos “en línea”, los Agentes Externos no son capaces de procesar las solicitudes en tiempo real. Se identifican dos razones principales de esto:

- **Procesamiento Batch**: Los Agentes Externos realizan gran parte de su trabajo en modo batch. Las solicitudes son acumuladas en lotes y procesadas posteriormente. Por ejemplo:
 - Un sistema bancario típicamente registra en línea los movimientos de una cuenta, pero retrasa a la hora de cierre el cálculo de totales, intereses, generación de balances, etc.
 - Un supermercado virtual registra en línea las compras de los clientes, pero generará los listados de “picking”¹³ una vez por hora.
- **Disponibilidad de Recursos**: Ciertos procesos pueden requerir para su ejecución de recursos que no estén permanentemente disponibles, y por eso tener tiempos de respuesta inciertos. Por ejemplo:
 - El proceso de generación de una Nota de Crédito termina con la impresión en papel del documento. Deseamos que el servicio nos de como respuesta que los documentos fueron impresos (por ejemplo para continuar con tareas humanas que requieran el papel)
 - Un proceso de compra por Internet puede requerir intervención humana en caso de constatar excepciones (como falta de stock), por lo que queda detenido.
 - Podemos depender de sistemas con altos niveles de “down-time”, que nos impidan procesar las solicitudes a tiempo.

Por esto, los Agentes Externos proponen una estrategia de **interacción asíncrona**. A diferencia de las estrategias habituales (SOAP, RPC, DCOM), en las que la solicitud y los resultados son intercambiados sobre la misma “conexión”, los Agentes Externos sincronizan con los clientes únicamente al inicio y al final del proceso, permaneciendo desconectados durante el tiempo intermedio.

5.3.3 Interfases sobre Intercambio de Documentos

Los Agentes Externos interactúan con el resto de las aplicaciones **intercambiando documentos XML**. De esta forma, se asegura poder colaborar con sistemas desarrollados sobre otras plataformas y lenguajes de programación.

¹³ Detrás de un sistema de supermercado virtual, los pedidos de los clientes son confeccionados y empacados por personal, quien recorre (con un carrito de verdad) las góndolas (de verdad) del depósito, seleccionando (“picking”) los productos que fueron elegidos por el cliente de forma “virtual”. Los listados de “Picking” optimizan el recorrido dentro del depósito, ordenando adecuadamente los ítems de cada pedido, según la disposición de los mismos en la planta del depósito.

El transporte de estos documentos puede ser realizado sobre varias estrategias de Middleware:

- Protocolos de Internet como HTTP, SMTP y FTP.
- Message Queues, Sistema de Archivos.

Los esquemas de datos intercambiados son propiedad de cada Agente. El cliente deberá encargarse de generar e interpretar los documentos en el “idioma” del mismo. Eventualmente, los Agentes Externos podrán soportar el intercambio de documentos dentro de la BizTalk Framework. Definiremos:

- Documento/Esquema de Solicitud: Documento/Esquema utilizado para solicitar un servicio. Para el agente XX, lo llamaremos **XX_Request**.
- Documento/Esquema de Respuesta: Documento/Esquema utilizado para notificar el progreso/estado de una tarea. Será **XX_Response**.

5.4 Paquete QflowB2B_Out

5.4.1 Escenarios de Interoperabilidad

Según el modelo de la WfMC, existen 4 escenarios de integración¹⁴ con sistemas de Workflow, en grado creciente de complejidad:

1. **Conexión Discreta:** Un proceso lanza la ejecución de otro proceso en un sistema externo. El primero continua sin esperar la finalización del segundo.
2. **Jerárquico:** Se lanza la ejecución de un proceso en otro sistema, y se espera a que éste termine para continuar, como si fuera un subproceso del primero.
3. **Peer-to-Peer:** Las tareas de ambos procesos se intercalan sin importar el sistema al que correspondan.
4. **Paralelos Sincronizados:** Ambos procesos corren de forma simultánea en su sistema, y definen puntos de sincronización el uno con el otro. El primer proceso que alcance el punto de sincronización espera a que el segundo llegue al suyo para continuar la ejecución.

La solución propuesta implementa los escenarios 1 y 2, que pueden ser implementados desarrollando un nuevo tipo de “paso externo”: El paso “EAI/B2B”.

5.4.2 El paso de EAI/B2B

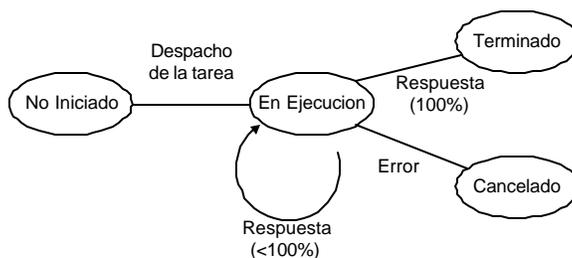
El paso EAI/B2B (o simplemente B2B) representa una “llamada a subproceso remoto” en un flow.

El mismo tiene asociados los valores:

- **Estado:** “No Iniciado”, “En Ejecución”, “Terminado”, “Cancelado”.
- **Porcentaje de Completitud:** Representa el grado de avance del proceso invocado.

Los mismos son actualizados a partir de las respuestas generadas por el Agente Externo durante la ejecución del paso. El paso podrá ser configurado para continuar la ejecución del flow inmediatamente (escenario 1), o esperar la finalización del proceso invocado (escenario 2). En este caso, el flujo continúa recién cuando el Agente notifica un 100% de avance.

Cuando un flow llega a un paso B2B, el mismo se encuentra en estado “No Iniciado”. Entonces, el Workflow lanza la ejecución del proceso externo, y el paso cambia al estado “En Ejecución”. En caso de no esperar respuesta del agente, salta inmediatamente al estado “Terminado”. Sino, el mismo queda “En Ejecución” hasta recibir una respuesta que indique “Terminado / 100%”. En caso de detectarse un error en el proceso, el Agente notifica el mismo y el paso queda en estado de “Error”.



¹⁴ El estudio de la WfMC considera únicamente la integración entre sistemas de Workflow. Generalizaremos la clasificación a la integración con otros sistemas.

5.4.3 Intercambio de Datos

El Intercambio de Datos es uno de los aspectos fundamentales de la Integración de Aplicaciones. En un sistema de Workflow, la información del proceso es almacenada en las variables (o Datos de aplicación), por lo que son el medio justo para el intercambio de datos.

El módulo B2B permite:

- Pasar Datos de aplicación del flow como **parámetro** al proceso invocado
- Impactar los valores de retorno de los procesos invocados en los Datos de Aplicación.

El modulo deberá observar la visibilidad definida para las variables en el paso:

	Editables	Visibles	Ocultas
Pasar como Parámetro	Si	Si	No
Recibir resultados	Si	No	No

5.4.4 Configuración

La configuración de un paso B2B se divide en:

- **Configuración de “Agreements”**: Un “Agreement” representa un acuerdo de EDI entre el sistema de Workflow y un Agente Externo.
- **Configuración del paso**: Cada paso estará asociado a un “Agreement” en tiempo de diseño.

Un “Agreement” contiene todos los parámetros necesarios para comunicarse con un Agente Externo:

- Esquemas de los documentos “Request” y “Response” utilizados por el agente.
- Mapeos entre los esquemas de Q-flow y los externos.
- Dirección de protocolo donde atiende el servicio (la URL del servicio)
- Información de envelopes, timeouts, firmas, etc.

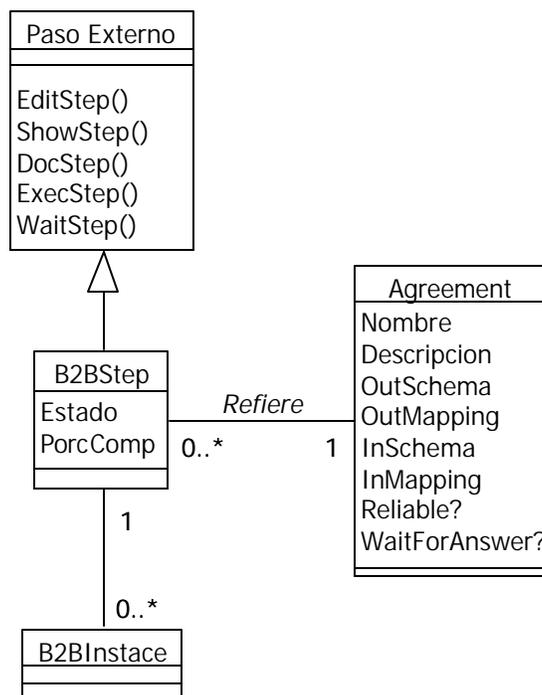
Configurar los “Agreements” como objetos independientes, permite separar las tareas de diseño de las de administración, de forma similar a como se trabaja con los pasos de BOT:

1. Primero, el administrador “instala” un Agreement y configura los aspectos técnicos del intercambio de documentos.
2. Luego el diseñador, agrega pasos de B2B a sus procesos y lo asocia a un Agreement según la lógica de negocio deseada.

5.4.5 Persistencia

Aparte de mantener de forma persistente tanto los Agreements como la configuración del paso, interactuar de forma asíncrona, exige que el sistema mantenga información de los procesos iniciados en los Agentes Externos. En particular, el sistema debe conservar información que le permita vincular las respuestas recibidas del exterior con las solicitudes que las generaron, y vincularlas a la instancia del paso que las originó.

5.4.6 Estructura Estática



5.4.7 Interfaz Paso Externo

La arquitectura de Q-flow prevé la incorporación de nuevos tipos de paso, desarrollando componentes que cumplan la interfaz "Paso Externo".

Un paso externo debe implementar las siguientes funciones:

- **EditStep**: Invocado por el diseñador de formularios al momento de editar el paso.
- **DocStep / ShowStep**: Son invocadas por el sitio Web para visualizar el paso.
- **ExecStep**: Lanza la ejecución de una instancia de paso.
- **WaitStep**: Consulta el estado de ejecución de una instancia de paso.

La rutina **EditStep** es invocada por la herramienta de diseño al momento de editar las propiedades del paso. La misma debe implementar la interfaz de usuario (un dialog box) con la cual el diseñador configura el paso. La rutina es invocada tanto para crear el paso en blanco como para editarlo. Debe devolver "Verdadero" en caso de que se deseen salvar los cambios, o "Falso" de lo contrario.

La rutina **DocStep** es invocada para documentar las propiedades del paso y debe devolver una tabla HTML con las propiedades del mismo. **ShowStep** es similar a la anterior, pero la tabla contendrá información del estado de ejecución del paso.

ExecStep es invocada por el motor para iniciar la ejecución de un paso. Debe responder al motor el estado en que queda el paso:

- **Ready**: El paso fue completado y se debe continuar la ejecución del flow inmediatamente en el paso siguiente.
- **Wait**: El paso queda en espera (ver WaitStep)
- **Error**: Se ha causado un error que amerita detener el workflow.
- **Sleep**: Estado de hibernación, no se esperan novedades pronto, a lo que el motor reduce a la mitad la frecuencia con que consultara el estado del paso (ver WaitStep)

WaitStep es invocada periódicamente por el motor mientras el paso está en espera. Debe indicar la acción a seguir, devolviendo un código de status de acuerdo al siguiente esquema:

- Ready: El paso ha sido completado y el flow puede continuar.
- Wait: El paso no ha sido completado aún.
- Error: Idem a ExecStep.
- Sleep: Idem a ExecStep.
- Notification: El motor debe generar una notificación en relación al paso.
- Alert: El motor debe generar una alerta en relación al paso.
- Delegate, Overtimed: Se cumplieron los plazos de ejecución del paso. El flujo debe continuar por el conector de delegación.

La persistencia del Template de paso (o sea, la configuración) es resuelta por la interfase Agente Externo. Cada función de la interfase recibe como parámetro un campo de tipo texto en donde se podrá serializar la configuración del paso, por ejemplo, en XML. El campo será editado por la función EditStep, y leído por el resto de las funciones.

5.4.8 Clase B2BStep

La clase B2BStep representa al tipo de paso B2B. El mismo implementa la interfase “Agente Externo” con la lógica del paso B2B.

5.4.9 Clase B2BInstance

La clase B2BInstance representa una “instancia de paso”. Cada ejecución de un paso de B2B, genera un objeto B2BInstance, el cual contiene información “de tiempo de ejecución” del paso. En particular, mantiene la información necesaria para vincular el paso con los documentos enviados y recibidos.

5.4.10 Clase B2BAgreement

La clase Agreement representa los acuerdos de EDI configurados en el sistema. Define todos los parámetros que necesita el sistema para gestionar el acuerdo:

- Esquemas de entrada y salida,
- Mapeos entre los esquemas de Q-flow y los externos,
- Información de envelopes, timeouts, firmas, etc.

Disponer de los agreements como objetos independientes de los pasos, permite separar las tareas de diseño de las de administración. El administrador configura los agreements independientemente de los pasos que lo utilicen, y el diseñador configura los pasos simplemente nombrando el agreement al que estarán asociados.

5.4.11 Servicios de Q-flow

Los clientes off-line de Q-flow interactúan con el sistema enviando mensajes con contenido XML a una casilla SMTP asignada al motor. En particular, los mensajes recibidos que cumplen con el esquema QFLOWRESPONSE, son interpretados como respuestas a un paso.

Para aprovechar este servicio, el modulo B2B se encarga de convertir las respuestas del Agente Externo al esquema QFLOWRESPONSE.

5.5 Paquete Middleware

5.5.1 Documentos

Los Agentes Externos interactúan con el resto de las aplicaciones intercambiando documentos XML. El esquema utilizado es propio del Agente, por lo que en potencia, cada uno de ellos utilizará un esquema distinto. El mismo deberá ser especificado como parte de los parámetros de integración.

Para poder adaptarse a esto, la lógica del sistema es genérica, siendo capaz de adaptarse a cualquier esquema de documento pasado como parámetro. La generación y procesamiento de documentos es guiada por la configuración de cada intercambio, la cual indica:

- Que información debe incluirse al construir un documento, y como debe ser formateada.
- Como recuperar la información de interés al recibir un documento.

5.5.2 Transporte

El sistema debe ser capaz de intercambiar documentos sobre cualquiera de los protocolos utilizados por el Agente. Al igual que los esquemas, el mismo es determinado por el Agente Externo, y especificado al momento de configurar el intercambio.

El sistema debe disponer de alguna estrategia para manejar errores de comunicación, que evite la cancelación del paso ante problemas en la infraestructura de comunicaciones, los cuales son probables en un escenario de EAI, y frecuentes en B2B. Como ejemplo, se podrá implementar una política de reintentos periódicos, o almacenar los documentos y permitir al administrador que repita manualmente los envíos fallidos.

Por las características de la información intercambiada, los Agentes Externos podrán requerir la aplicación de políticas de seguridad en el intercambio como ser:

- Encriptación : HTTPS, FTPS, S/MIME.
- Firmado digital de documentos.

Esto es deseable en un escenario de EAI, y vital para los casos de B2B, en los que se utiliza infraestructura pública para el intercambio de documentos.

5.5.3 BizTalk Server

Al momento de proponer una solución, surgen dos alternativas:

- Desarrollar todo el soporte multi-esquema , multi-protocolo necesario sobre las primitivas básicas de XML (DOM, XSLT) y de cada protocolo (CDO¹⁵, XMLHTTP, etc.), y sobre éste la lógica del paso, o
- Utilizar Microsoft BizTalk Server, y programar el intercambio sobre primitivas de alto nivel como “canal”, “puerto”, “documento”, “mapeo”, etc.

El módulo adoptara la segunda de las estrategias. Utilizar BizTalk permite delegar todos los aspectos de comunicación, y enfocar el desarrollo en la lógica del paso B2B.

Como forma de justificar nuestra decisión, comparemos nuestra propuesta con la estrategia de integración adoptada por el sistema de Workflow “Ultimus”.

¹⁵ CDO: “Colaboration Data Objects” es una biblioteca provista por Microsoft que permite acceder a las funciones de la API de Mensajería de Windows (MAPI) mediante primitivas de mayor nivel.

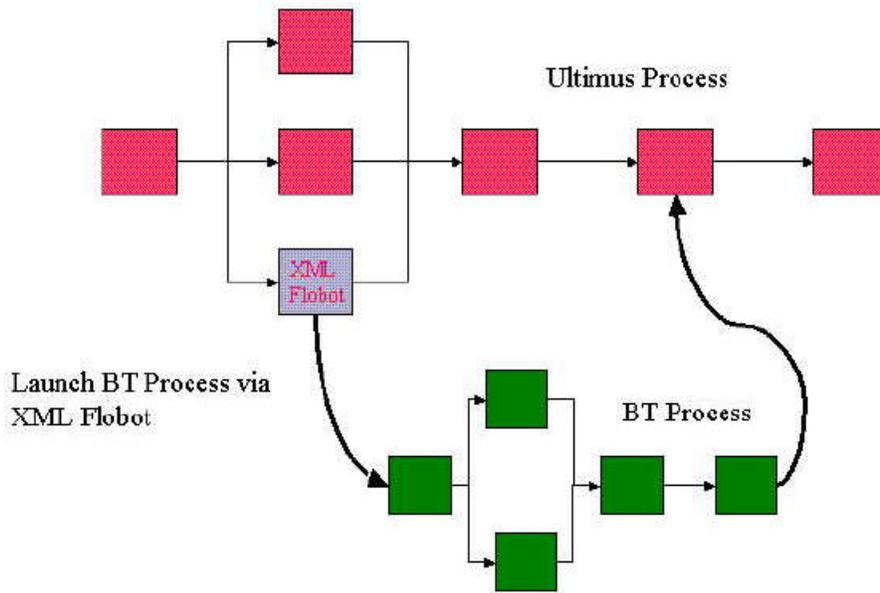
5.5.4 Ultimus Workflow Software

La solución de Ultimus para la integración con aplicaciones empresariales, se basa en dos funcionalidades provistas por el sistema[Ultimus]:

- XML FloBot: El “XML FloBot” es un tipo de paso que permite generar documentos XML a partir de datos de aplicación, y enviarlos a un destinatario vía HTTP.
- XML Step Completion: Permite que un tercero de por finalizada la ejecución de un paso enviando un documento XML (en formato propietario de Ultimus) al motor de Workflow.

Para invocar un agente externo, un proceso de Ultimus debe:

- Generar y despachar un documento XML utilizando el XML FloBot, y
- Recuperar el resultado deteniéndose en un paso, que será completado desde el exterior mediante XML Step Completion.

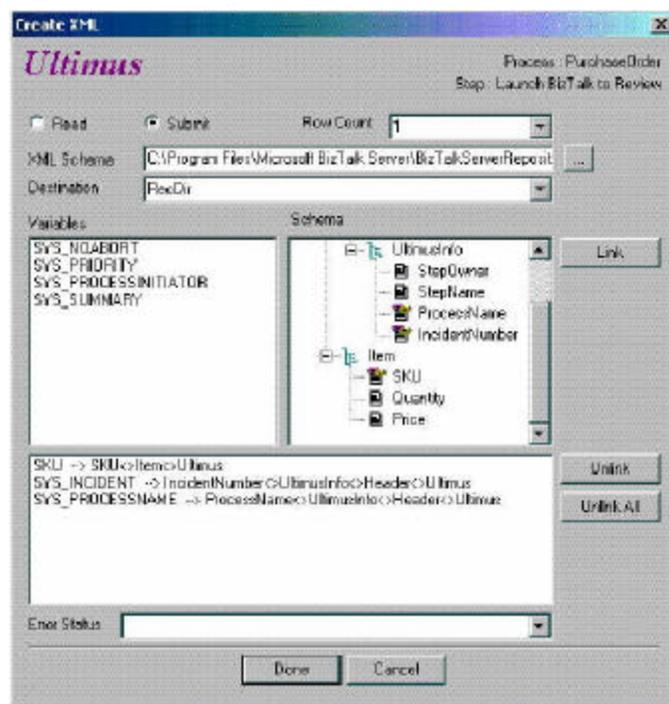


La funcionalidad del FloBot XML queda descrita por su configuración. La misma permite especificar:

- Un esquema de datos para el documento a generar
- Una URL de destino
- Una relación entre las variables del flow y los tags del documento.

Podemos imaginar que el sistema ejecuta los siguientes pasos para ejecutar el FloBOT:

- Cargar un objeto DOM con la estructura descrita por el esquema,
- Cargar los valores de las variables en la estructura, identificando los nodos mediante rutas XPath generadas a partir de la información de los mapeos, y
- Enviar el documento al destinatario utilizando XMLHTTP.



En comparación a las posibilidades que ofrece BizTalk, la solución de Ultimus plantea múltiples limitaciones:

- Solo se dispone del protocolo HTTP, lo cual restringe el tipo de aplicación que podremos invocar. Utilizando BizTalk podremos ofrecer SMTP, FTP, Message Queues, etc. por el esfuerzo de implementar un solo protocolo.
- No existe manejo de errores. En el mejor de los casos, el paso da error cuando la comunicación falla. BizTalk maneja políticas de retransmisión que aseguran que los documentos no se pierdan en casos de caída del Agente Externo.
- No hay manejo de seguridad. No se dispone de encriptación ni de firmado digital de los documentos
- No se soporta el intercambio de documentos dentro de la BizTalk Framework, por lo que se desaprovechan los beneficios de esta como entrega segura de documentos.
- Los mapeos tipo uno-a-uno permitidos son bastante restrictivos comparado con las posibilidades de la tecnología (léase XSLT). BizTalk Mapper ofrece esta potencia y de forma visual.
- Las funciones de XML Step Completion manejan un único formato de documento, por lo que será necesario desarrollar código que convierta el esquema del Agente en el de Ultimus (“glue code” [Mohr]). La solución propuesta es capaz de interpretar cualquier esquema como respuesta de forma nativa.

En resumen, utilizar BizTalk permite elaborar una solución mas económica en líneas de código, mas flexible y mucho mas potente.

5.6 Paquete QflowB2B_In

El paquete QflowB2B_In estudia el escenario simétrico a QflowB2B_Out. El mismo analiza el caso en el que un Agente Externo es quien inicia procesos en Q-flow.

A diferencia de la invocación de procesos externos, el inicio de flows desde el exterior ya existe en Q-flow. El módulo QflowB2B solamente amplía funcionalidades ya provistas por el sistema, aprovechando los beneficios de la estructura de Middleware construida.

5.6.1 Servicio “StartFlowRequest”

El servicio “StartFlowRequest” de Q-flow habilita el inicio de flows desde una aplicación externa. El mismo, es similar conceptualmente al inicio de flows desde el acceso Web:

1. Se selecciona un Template (indicando el ID),
2. Se carga el Título para el Flow,
3. Se cargan valores iniciales para los Application Data,
4. Se cargan los roles del flow,
5. Se adjuntan archivos,
6. Se inicia el flow.

5.6.2 Interfases del Servicio

Q-flow ofrece las siguientes interfases para el servicio “StartFlowRequest”:

1. Componente ActiveX: Instanciando al objeto “QFlowCOM.START” incluido en la biblioteca QFlowCOM.dll
2. Documento XML vía SMTP: Enviando un documento que cumpla el esquema QFLOWSTARTREQUEST a la casilla del motor. Los archivos adjuntos se toman de los attachments del mail.
3. Documento XML vía Message Queue: Ídem, pero a través de Message Queues.
4. Stored Procedure: Invocando el stored procedure “sp_StartFlow” en la base de Q-flow.

5.7 Resumen

“Integrar un sistema de Workflow con aplicaciones externas” consiste en permitir que los procesos manejados por el sistema de Workflow invoquen procesos ejecutados por otras aplicaciones. Esta invocación, es representada en el diseño de workflows por un tipo de paso particular: El paso B2B. El mismo permite invocar un proceso externo, pasarle datos de aplicación como parámetro, y recuperar en los mismos los valores de retorno del proceso invocado.

Cada paso B2B tiene un estado de ejecución asociado (“No Iniciado”, “Iniciado”, “Finalizado” o “Cancelado”) y un porcentaje de completitud.

Para incorporar el nuevo tipo de paso al sistema, se debe desarrollar un módulo que implemente la interfaz “Paso Externo”, definida por la arquitectura de Q-flow. La misma incluye funciones para editar la configuración del paso, iniciar su ejecución, controlar su estado de avance, etc.

Dada la diversidad de aplicaciones empresariales existentes, se restringe el alcance a un subconjunto de las mismas que llamaremos “Agente Externo”. Un Agente Externo es un sistema que recibe solicitudes de servicio en forma de documentos XML, realiza alguna tarea y, de forma asíncrona, notifica progreso/completitud/éxito devolviendo otro documento XML al remitente. El esquema de los datos intercambiados es propiedad del agente; nuestro sistema deberá ser capaz de adaptarse a los mismos. De forma similar, el protocolo de transporte será elegido por el agente, pudiendo ser HTTP, SMTP, FTP, Message Queues, etc.

Esta definición de “Agente Externo” define las características de la comunicación entre éstos y el sistema de Workflow, la cual podría resumirse como “**intercambio asíncrono de documentos XML**”, donde el esquema de los datos y el protocolo de transporte son determinados por el agente.

Al momento de proponer una solución se presentan dos opciones:

- Implementar a partir de las primitivas de XML (XMLDOM) y de los protocolos (XMLHTTP, CDO, etc.) un módulo de comunicación multi-esquema y multi-protocolo, para luego, sobre éste implementar la lógica del nuevo paso, o
- Delegar todos los aspectos del intercambio de documentos a BizTalk, implementando el paso B2B sobre primitivas de alto nivel como “canal”, “puerto”, “organizacion”, etc, enfocándonos directamente en la lógica del paso.

El proyecto desarrollará la segunda opción, lo cual nos permitirá obtener una solución mas potente en menos líneas de código.

De forma simétrica, se deberá permitir que los Agentes Externos inicien procesos en Q-flow. A diferencia de la invocación de procesos externos, el inicio de flows desde el exterior ya existe en Q-flow, por lo que el módulo solamente amplía funcionalidades ya provistas por el sistema, aprovechando los beneficios de la estructura de Middleware construida.

6 Diseño

Habiendo completado el análisis de los objetivos, el siguiente capítulo plantea la solución que será finalmente desarrollada.

6.1 Descripción General

Ejecutar un paso B2B consiste en invocar el proceso externo asociado y recuperar los resultados que el mismo pueda generar. Como vimos, esto es hecho intercambiando documentos XML en el “idioma” del Agente Externo(AE):

- El módulo solicita la ejecución del proceso externo enviando el documento “AE_Request” al Agente, y
- El Agente responde a la solicitud enviando (eventualmente mas de uno) documentos “AE_Response” al módulo.

El problema se resume entonces, a generar y procesar los documentos “AE_Request” y “AE_Response”, **cuyo esquema es a priori desconocido**, y será provisto como parte de la configuración .

El diseño propuesto, ataca el problema dividiéndolo en:

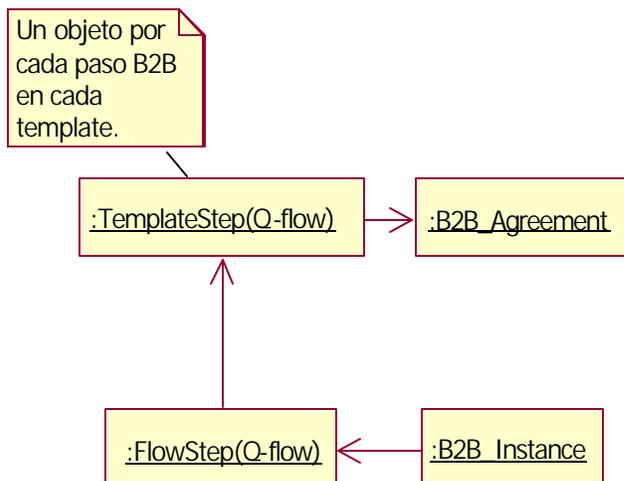
1. Generar y procesar documentos en los esquemas estándar B2B_Request y B2B_Response, y
2. Mapear los documentos B2B_Request y B2B_Response a los esquemas AE_Request y AE_Response.

Mediante esta división, se desacopla el contenido de los documentos de su formato, lo que nos permite implementar la generación / procesamiento de la información por separado del manejo de formato, el cual será delegado a BizTalk.

La solución queda entonces dividida en dos partes:

- **QflowB2B**: Implementa la interfase “Paso Externo” con la lógica del paso B2B, generando y procesando la información que necesita intercambiarse con el Agente. Se comunica con BizTalk mediante esquemas estándar B2B_Request y B2B_Response.
- **Configuración BizTalk**: Se encarga de darle el formato y estructura adecuada a la información generada por QflowB2B para que sea “entendida” por el Agente. Maneja el envío y recepción de documentos en los protocolos acordados con el Agente.

6.2 Paquete QflowB2B



TemplateStep: Configuración del paso dentro del diseño del Template.

FlowStep: Información de la ejecución del paso. Hay uno por cada vez que un flow ejecuta el paso (mas de una en caso de que el flow tenga loops).

B2B_Agreement: Configuración del Agreement al que hace referencia el paso.

B2B_Instance: Información acerca de la ejecución del proceso externo.

6.2.1 ExecStep

El método ExecStep se encarga de iniciar la ejecución del proceso externo. El mismo confecciona el documento B2B_Request a partir de los datos del flujo. Luego, identifica el Agreement asociado al paso, y de éste el nombre de un canal BizTalk, a donde despacha el documento. Finalmente, crea una nueva B2B_Instance que permitirá vincular los documentos intercambiados con el paso.

6.2.2 Procesamiento de B2B_Response

Para el procesamiento de las respuestas generadas por el Agente Externo, el paquete aprovecha el procesamiento de documentos QFLOWRESPONSE que ofrece Q-flow. Para esto, **se hace coincidir el esquema B2B_Response con el esquema QFLOWRESPONSE**, y se configura BizTalk para que despache los documentos directamente a la casilla del motor.

El procesamiento de los documentos QFLOWRESPONSE actualiza el estado del paso actualizando el objeto FlowStepsTo (de Q-flow) asociado al mismo. Para aprovechar esto, el método ExecStep crea un objeto de este tipo y lo asocia al paso B2B. (Conceptualmente, el objeto FlowStepsTo es parte de B2B_Instance).

6.2.3 WaitStep

El método WaitStep es invocado periódicamente por el motor para consultar el estado de ejecución del paso. El mismo responde si el paso sigue en ejecución o si ha sido completado. En nuestro caso, el método decide a partir de la información mantenida en el objeto FlowStepsTo asociado al paso por ExecStep, el cual es actualizado cada vez que se recibe una respuesta del Agente.

6.2.4 Interfaz QFlowB2B - BizTalk

La comunicación entre QflowB2B y BizTalk es implementada invocando el componente “IInterchange” provisto por BizTalk. El mismo ofrece un método “Submit” que permite despachar documentos a un canal BizTalk.

6.3 Paquete Configuración BizTalk

6.3.1 Agreement

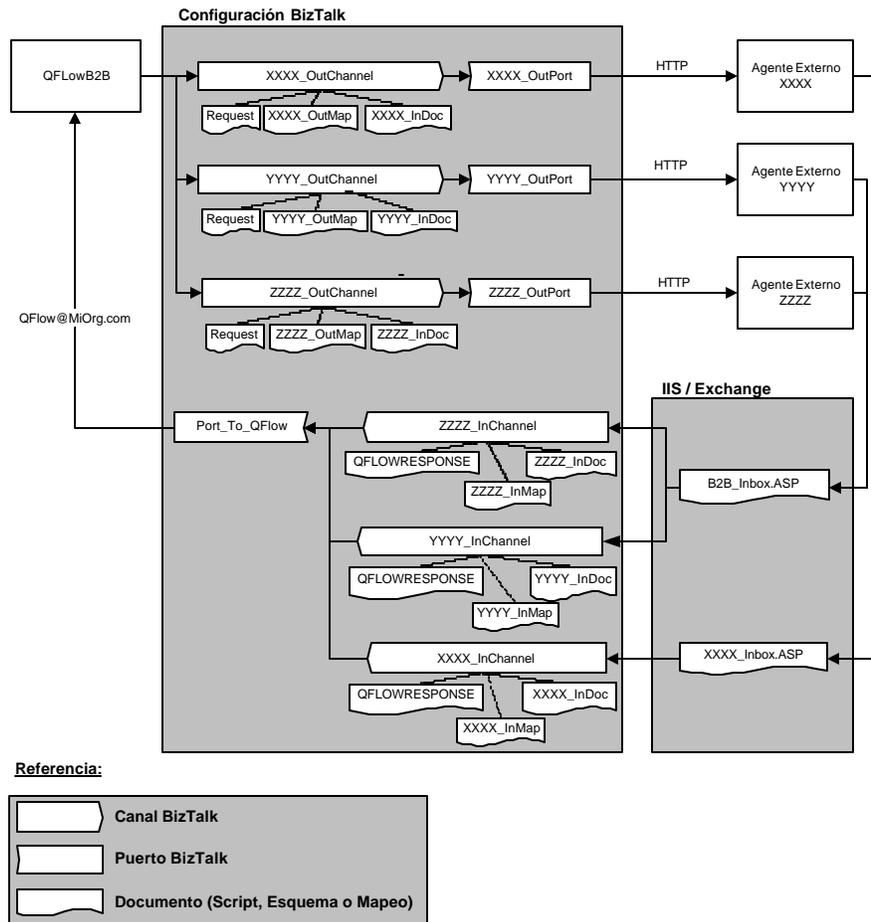
Un “Agreement” define las características de la interacción entre el Workflow y un Agente Externo particular. El mismo contiene la siguiente información:

- Esquemas “XX_Request” y “XX_Response”: Esquemas de los documentos que define el Agente Externo. Para el diseño de los mismos se podrá contar con la herramienta BizTalk Editor.
- Mapeos “XX_OutMap” y “XX_InMap”: Archivos de BizTalk Mapper que definen la conversión del esquema B2B_Request en XX_Request y XX_Response en QFLOWRESPONSE. Son parte fundamental de la lógica del intercambio. En ellos se define cómo los datos de aplicación del flow pasan como parámetro de la invocación, y cómo impactan los resultados en las variables del flow.
- “Esperar Respuesta”: Indica si el Agente Externo genera notificaciones de progreso, o el paso debe darse por finalizado inmediatamente. En este ultimo caso, no se requiere ni el esquema ni el mapeo de respuesta.
- “URL”: Direcciones de protocolo donde el Agente Externo recibe las solicitudes
- “Intercambio confiable” (“reliable”): Indica si los documentos son empacados en un envelope BizTalk al momento de ser enviados al Agente.

Cada Agreement debe ser “instalado” en el sistema antes que un paso pueda hacer referencia al Agente Externo. El proceso de instalación, crea los objetos de configuración en BizTalk que soportan el Agreement y los configura de acuerdo a los parámetros del mismo.

Un Agreement podrá ser llevado fuera de línea “desinstalándolo”. El proceso elimina los objetos de configuración de BizTalk, conservando el Agreement para futuras “reinstalaciones”.

6.3.2 Arquitectura de Configuración



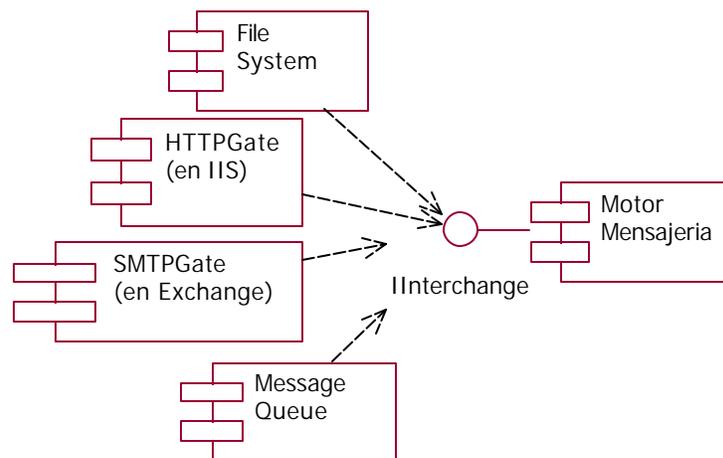
Cada Agreement XX es implementado por los siguientes objetos en BizTalk:

- Envío de documentos:
 - Canal XX_OutChannel: Recibe los documentos de QFlowB2B, y los mapea al esquema de salida. Configuración:
 - Esquema del documento de Entrada: “B2B_Request”
 - Esquema del documento de Salida: “AE_Request”
 - Mapeo aplicado: “AE_OutMap”
 - Puerto de salida: “XX_OutPort”
 - Puerto XX_OutPort: Entregar los documentos al Agente Externo. Configuración:
 - Protocolo: HTTP.
 - URL: Dirección donde atiende el Agente Externo.
 - Reliable?: Indica si el puerto empaqueta los documentos en un envelope BizTalk para el envío.
- Recepción de documentos:
 - Canal XX_InChannel: Recibe los documentos AE_Response, los mapea al esquema QFLOWRESPONSE, y los pasa al puerto Port_to_Qflow. Configuración:
 - Esquema del documento de Entrada: “B2B_Request”
 - Esquema del documento de Salida: “AE_Request”
 - Mapeo aplicado: “AE_OutMap”
 - Puerto de salida: “XX_OutPort”
 - Puerto Port_to_QFlow: Es común a todos los Agreements. Esta configurado para entregar los documentos que recibe en la casilla de correo del motor.

6.3.3 Funciones “Receive” y Scripts de “Gate”

El motor de mensajería de BizTalk ofrece solamente la interfase COM “IInterchange” para la recepción de documentos. El resto de los middleware, son manejados por componente incluidos en BizTalk llamados “Receive Functions”. Los mismos funcionan como pasarela entre el protocolo de origen y la interfase IInterchange de COM:

- **File System:** Revisa periódicamente el contenido de un conjunto de directorios configurados, y despacha los archivos recibidos como mensajes.
- **Message Queue:** Consume mensajes recibidos por un conjunto de colas configuradas
- **HTTP/SMTP:** Se instalan Scripts en los servidores correspondientes (IIS y Exchange).
- **FTP:** Se debe configurar el servicio de File System para que escanee las carpetas en donde se habilita la recepción de documentos vía FTP.



Para manejar HTTP y SMTP, BizTalk incluye los scripts “ReceiveStandard”, los cuales funcionan solamente para mensajes empacados en un “BizTalk Envelope”. Ya que ésta es una restricción que no podrán cumplir todos los Agentes Externos, debemos proveer una solución alternativa.

El aspecto principal del diseño de estas pasarelas, es la forma en la que se identifica el canal BizTalk a donde debe ser dirigido cada mensaje recibido. Se proponen tres soluciones, de las cuales cada Agreement podrá elegir la más adecuada:

- **ReceiveStandard.asp:** Los documentos se reciben en un BizTalk Envelope por lo que no es necesario identificar explícitamente el canal BizTalk al invocar a IInterchange.
- **B2BGate.asp:** El Agente pasa el Id. de Agreement en los parámetros del comando Post cuando transmite el documento. El script recupera el parámetro, y de este identifica el canal a donde finalmente despacha el documento recibido.
- **xxxxGate.asp:** El nombre del canal se define “Hard Coded” en el script. Se deberá tener un script particular para cada acuerdo (todos idénticos a diferencia del nombre del canal). Será útil para implementar acuerdos en los que fallen las soluciones anteriores.

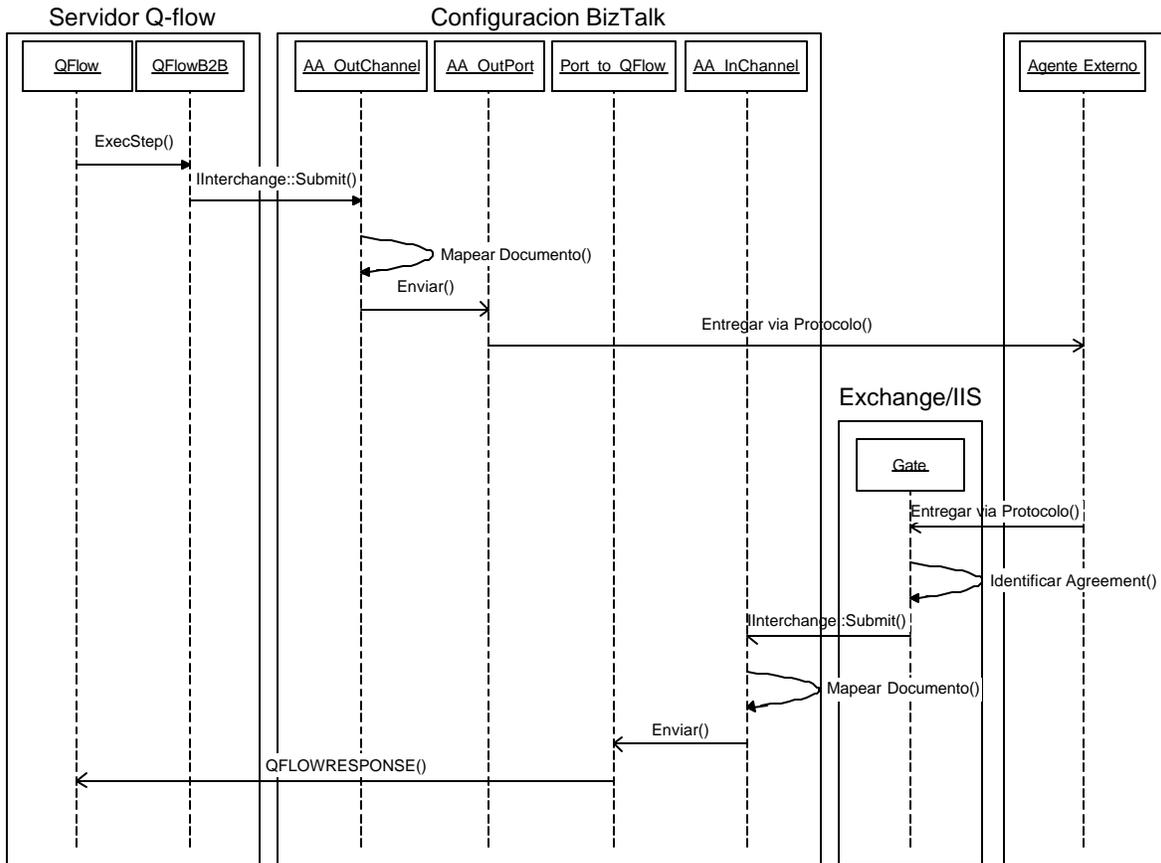
6.3.4 Inicio de Flows desde el Exterior

El inicio de un flow desde el exterior es similar al procesamiento de respuestas a un paso.

Los documentos “XX_StartFlow” son recibidos por el canal “XX_StartChannel”, el cual mapea el documento al esquema QFLOWSTARTREQUEST de Q-flow, y se lo pasa al puerto “Port_to_QFlow”.

6.4 Diagrama de Secuencia

El siguiente Diagrama de Secuencia describe el flujo de mensajes entre los distintos componentes del sistema, desde el inicio del paso hasta recibir la respuesta del Agente Externo. Luego de haber estudiado cada paquete por separado, el mismo permite tener una visión global del diseño de la solución.



1. Cuando el control de un flujo llega a un paso B2B, el motor invoca al método ExecStep de QflowB2B.
2. El método entonces, construye el documento “B2B_Request” con información del flujo (incluyendo los datos de aplicación) y del paso. Luego, a partir de la configuración del paso, identifica el Agreement asociado (digamos XX), y envía el documento por el canal “XX_OutPort”.
3. El canal recibe el documento B2B_Request, lo convierte al esquema del Agente Externo aplicando el mapeo XX_OutMap, y
4. Se lo pasa al puerto, quien transmite el documento al agente en el protocolo asociado.
5. Las respuestas del Agente Externo son recibidas por el script “Gate”, el cual (de ser necesario) identifica el canal XX_InChannel y despacha el documento.
6. El canal recibe el documento XX_Response y lo convierte al esquema QFLOWRESPONSE aplicando el mapeo XX_InMap, y finalmente
7. El puerto Port_to_QFlow entrega el documento en la casilla de correo del motor.

6.5 Resumen

El diseño de la solución se divide en dos partes: Elaborar una “arquitectura de configuración” para BizTalk que resuelva el intercambio de documentos, y diseñar un componente que implemente el paso B2B aprovechando la arquitectura configurada en BizTalk. Con esta división se logra que la implementación del paso maneje un único esquema de datos y delegue en BizTalk la conversión entre esquemas y el transporte de los documentos.

La configuración de BizTalk se compone, para cada Agente Externo, de un par canal-puerto para los documentos salientes y otro par simétrico para los documentos entrantes. Los canales son configurados para convertir los documentos de un esquema propio (B2B_*) al esquema aceptado por el agente (XX_*) mediante mapeos provistos por el administrador. Los puertos por su parte, entregan el documento recién procesado a su destinatario, ya sea el Agente Externo para los documentos salientes o la casilla de correo del motor para los documentos entrantes.

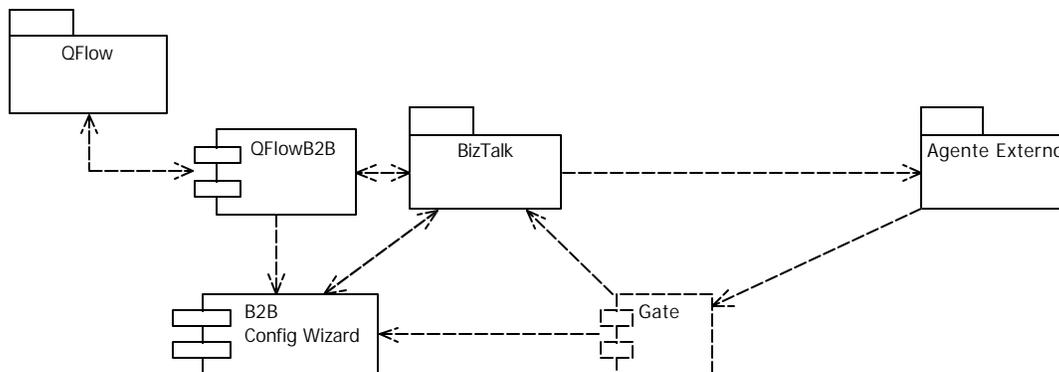
El componente QFlowB2B implementa el paso B2B implementando la interfaz “Paso Externo”. Cuando el motor inicia el paso invocando a la función ExecStep, la misma construye un documento XML (en formato B2B_Request) con toda la información que se desea exponer del flujo, y lo envía al canal adecuado en BizTalk. El paso puede entonces quedar en espera o continuar inmediatamente, de acuerdo a su configuración. En caso de quedar en espera, el motor invoca periódicamente a la función WaitStep, que consulta si la última respuesta recibida del Agente (procesada por el motor como una QFLOWRESPONSE) indica o no la finalización del proceso.

7 Implementación

7.1 Plataforma de Software

- Sistema Operativo: Windows 2000 Advanced Server
- Base de Datos: SQL 2000 (accedida utilizando ADO 2.6)
- Servidor Web: Microsoft Internet Information Server
- Servidor de Correo: Microsoft Exchange
- Entorno de desarrollo: Visual Studio 6. Lenguaje: Visual Basic.
- Microsoft BizTalk Server 1.0
- Q-flow 2.0

7.2 Vista de Componentes



- **QflowB2B**: Biblioteca ActiveX. Expone el objeto “QflowExtB2B.XMLStep”, que implementa el paso B2B. Tiene visibilidad sobre los Agreements definidos por “B2B Config Wizard”.
- **BizTalk**: Intercambia documentos con QFlowB2B y los procesa de acuerdo a los objetos de configuración generados por B2B Config Wizard
- **B2B Config Wizard**: Herramienta de configuración de Agreements. Instala / Desinstala los Agreements creando / eliminando los objetos de configuración BizTalk que soportan al mismo.
- **B2BGate**: Script ASP encargado de recibir las respuestas transmitidas por los agentes vía HTTP y rutearlas al canal adecuado en BizTalk. Tiene visibilidad sobre los Agreements definidos por “B2B Config Wizard”.

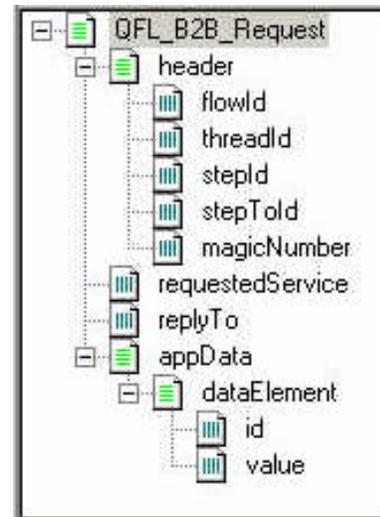
7.3 Documentos XML

7.3.1 B2B_Request

Como vimos en el diseño, la generación de documentos “personalizados” a cada Agente es hecha definiendo mapeos entre el esquema B2B_Request y el esquema que maneja el Agente (AE_Request). Por esto, el documento B2B_Request debe contener toda la información que pueda ser útil al momento de diseñar el mapeo. El mismo define la visibilidad que tendrán los sistemas externos sobre la información de cada flow.

La estructura del documento debe ser lo mas explícita posible, de forma de facilitar el diseño de los mapeos.

```
<QFL_B2B_Request>
  <header>
    <flowId >flowId_1 </flowId >
    <threadId >threadId_1 </threadId >
    <stepId >stepId_1 </stepId >
    <stepToId >1 </stepToId >
    <magicNumber>magicNumber_1 </magicNumber>
  </header>
  <requestedService >Service_1 </requestedService >
  <replyTo >replyTo_1 </replyTo >
  <appData >
    <dataElement >
      <id>id_1 </id >
      <value>value_1 </value >
    </dataElement >
    <dataElement >
      <id>id_2 </id >
      <value>value_2 </value >
    </dataElement >
  </appData >
</QFL_B2B_Request >
```

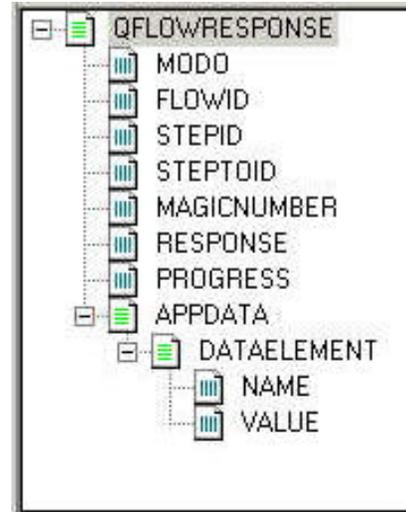


- **Header:** El header del documento incluye información necesaria para identificar el paso. La misma es copiada al documento QFLOWRESPONSE, lo que permite al módulo vincular las respuestas con el paso en ejecución.
- **FlowId, threadId, stepId, stepToId:** Requeridos. Identifican el flow, thread y paso que está siendo ejecutado. StepToId no es utilizado por la solución, pero es mantenido por consistencia con QFLOWRESPONSE, previendo que a futuro sea necesario. Siempre contiene el valor “1”.
- **MagicNumber:** Requerido. Hashing generado por el motor a partir de los valores anteriores que permite verificar que una respuesta es genuina.
- **RequestedService:** Opcional. Permite indicar el servicio que esta siendo invocado, en caso de que el agente ofrezca varios servicios en la misma dirección.
- **ReplyTo:** Opcional. URL donde el sistema procesa las respuestas. El motor completa este campo con el valor %B2BGate_URL%?IxId=nnn, donde %B2BGate_URL% es la URL del Script B2BGate, y “nnn” es el Id de la instancia asociada, lo que permite al B2BGate.asp identificar el canal a donde debe despachar el documento recibido. El campo puede o no ser aprovechado, dependiendo del Agreement.
- **DataElement:** Se incluye uno por cada dato de aplicación del flow. El campo “Id” contiene la identificación del dato, y “Value” el valor.

7.3.2 QFLOWRESPONSE

El formato de documento QFLOWRESPONSE es definido por el motor de Q-flow. El mismo incluye toda la información necesaria para actualizar el estado de ejecución de un paso de tarea.

```
<QFLOWRESPONSE >
  <MODO>MODO_1</MODO >
  <FLOWID>FLOWID_1</FLOWID>
  <STEPID>STEPID_1</STEPID>
  <STEPTOID>STEPTOID_1</STEPTOID>
  <MAGICNUMBER>MNMBR_1</MAGICNUMBER>
  <RESPONSE >RESPONSE_1</RESPONSE >
  <PROGRESS >PROGRESS_1</PROGRESS >
  <APPDATA >
    <DATAELEMENT >
      <NAME >NAME_1</NAME >
      <VALUE>VALUE_1</VALUE>
    </DATAELEMENT >
    <DATAELEMENT >
      <NAME >NAME_2</NAME >
      <VALUE>VALUE_2</VALUE>
    </DATAELEMENT >
  </APPDATA >
</QFLOWRESPONSE >
```



- **MODO**: Ignorado
- **FLOWID, STEPID, STEPTOID**¹⁶: Idem a B2B_Request.
- **MAGICNUMBER**: Ídem
- **RESPONSE**: Indica el estado de ejecución del paso:
 - " No iniciado" = 1
 - "En progreso" = 2
 - "Terminado" = 3
 - "Cancelado" = 4
- **PROGRESS**: Numero entre 0 y 100 que indica el porcentaje de progreso.
- **DATAELEMENT**: Idem B2B_Request.

¹⁶ Observar la omisión del ThreadId.

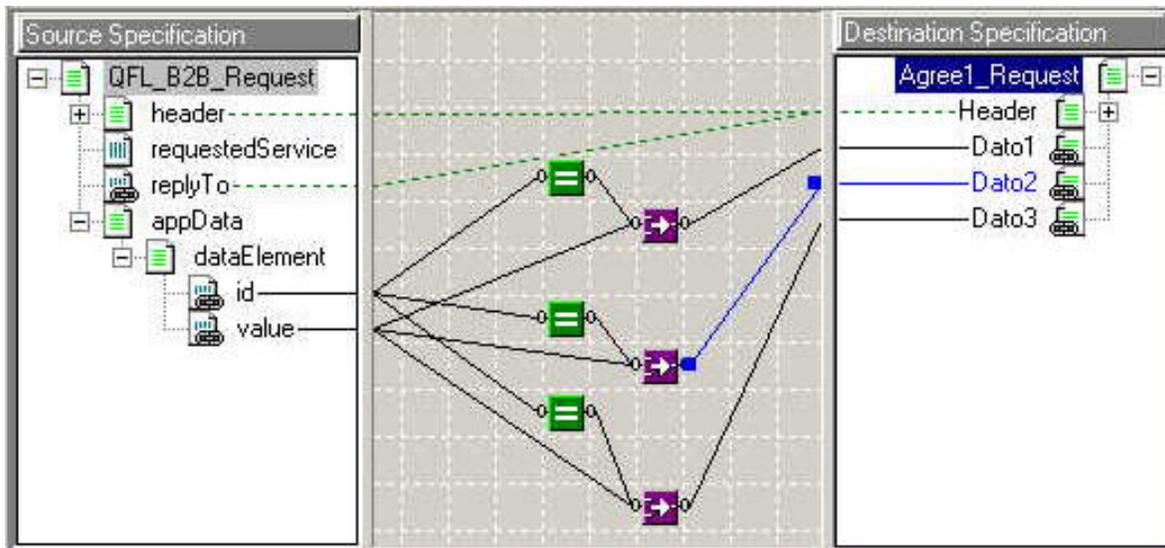
7.3.3 Mapeos

Los mapeos XX_InMap y XX_OutMap definen la relación entre los datos del flow y el contenido de los documentos. Los mismos son diseñados de acuerdo a las reglas de cada Agreement y deben ser provistos por el administrador como parte de la configuración.

El ejemplo de abajo muestra como el mapeo “Agree1_OutMap” convierte el documento B2B_Request en el formato “Agree1_Request”. La información del header del documento de origen es hecha coincidir con los campos del header de destino. En caso de que el documento de destino sea muy restrictivo, y no permita incluir suficientes campos, la información del header puede ser concatenada (incluyendo separadores de algún tipo)utilizando functoides, y reconstruida en el mapeo Agree1_InMap mediante functoides de Script. Para simplificar el ejemplo, el subárbol del header se presenta contraído (línea punteada).

El mismo recibe como parámetro los valores Dato1, Dato2 y Dato3. Observar la estrategia con la que se convierten instancias múltiples del tag “dataElement” a valores únicos:

-  Se configura para que compare el valor de entrada con el nombre de la variable. Devuelve “True” si son iguales.
-  Si recibe “True” del functoides de igualdad, pasa el valor de la variable generando el tag correspondiente con el dato contenido en el tag de origen “value”

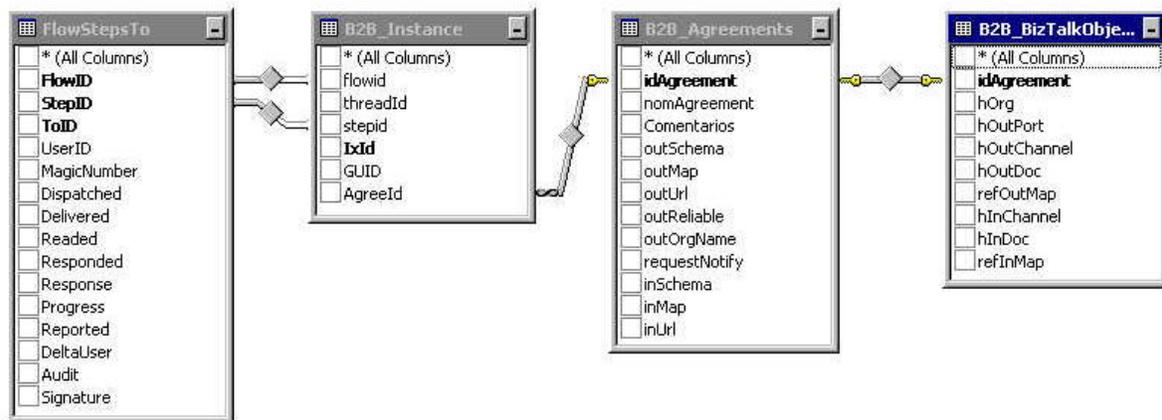


7.4 Persistencia

El módulo de B2B, al igual que los motores de BOT, almacena su información dentro de la propia base de datos de Q-flow, aun siendo implementados de forma externa al sistema. Para esto, se agregan a la base tres nuevas tablas:

- B2B_Agreements
- B2B_Instance
- B2B_BizTalkObjects

El acceso a la base de datos es hecho a través del componente “OleDbCompat” de la biblioteca QSoftData.dll de Q-flow.



7.4.1 Tabla B2B_Instance

La tabla B2B_Instance almacena la información de las instancias en ejecución. Permite vincular un IxId (id de intercambio) con el paso que lo generó. Registra también el GUID retornado por IInterchange al despachar el documento, lo que permitirá en el futuro consultar el estado de procesamiento del documento en BizTalk. El AgreeId se mantiene en la tabla para evitar buscarlo en la configuración del template del paso, donde se almacena como XML. La misma es accedida por QflowExtB2B y por B2BGate.asp

	Nombre	Tipo	Largo	Null
	flowid	decimal	9	No
	threadId	decimal	9	No
	stepid	decimal	9	No
Clave -->	IxId	decimal	9	No
	GUID	char	50	Si
	AgreeId	int	4	Si

7.4.2 Tabla B2B_Agreements

La tabla B2B_Agreements almacena la información de los Agreements registrados en el sistema. La misma es accedida por QflowExtB2B y por la herramienta de configuración.

	Nombre	Tipo	Largo	Null
Clave -->	idAgreement	int	4	No
	nomAgreement	char	16	No
	Comentarios	text	16	Si
	outSchema	text	16	Si
	outMap	text	16	Si
	outUrl	text	16	Si
	outReliable	int	4	Si
	outOrgName	text	16	Si
	requestNotify	int	4	Si
	inSchema	text	16	Si
	inMap	text	16	Si
	inUrl	text	16	Si

7.4.3 Tabla B2B_BizTalkObjects

La tabla B2B_BizTalkObjects almacena las referencias de los objetos de configuración BizTalk asociados a un Agreement, lo que permite eliminarlos al desinstalar el Agreement.

	Nombre	Tipo	Largo	Null
Clave -->	idAgreement	int	4	No
	hOrg	int	4	No
	hOutPort	int	4	No
	hOutChannel	int	4	No
	hOutDoc	int	4	No
	refOutMap	text	16	Si
	hInChannel	int	4	Si
	hInDoc	int	4	Si
	refInMap	text	16	Si

7.5 Módulo QflowB2B

La biblioteca QFlowB2B.dll exporta el componente “QflowExtB2B.XMLStep”, que implementa las funciones de la interfaz “Paso Externo”.

7.5.1 EditStep

Invocado por el diseñador de formularios al momento de editar el paso.

Sintaxis

```
Public Function EditStep( ByRef stCaption As String, _
    ByVal lnStepID As Long, _
    ByRef stXMLInfo As String, _
    ByVal TemplateID As Long, _
    ByVal VersionID As String, _
    ByVal Editmode As Long, _
    ByRef objKernel As QSoftKRN.Kernel, _
    ByRef QFlowDBConn As ADODB.Connection
) As Boolean
```

Parámetros

- **StCaption**: Nombre del paso. String de hasta 30 caracteres. Editable
- **LnStepID**: Identificador del paso.
- **StXMLInfo**: Información del paso. Toda la configuración del mismo se guarda aquí. String en formato XML
- **Tempalteid**: Identificador del template
- **Versionid**: Número de la versión de template que se esta editando.
- **Editmode**: Modo de edición en el que esta trabajando el Designer.
 - 0 = Full
 - 1 = Editar compatible
 - 2 = Editar Review
 - 3 = Browse (read only)
- **Kernel**: Contiene una colección de los datos de aplicación, otra con los roles y otra con los pasos del flow.
- **QFlowDBConn**: Conexión a la base de Q-flow. Viene abierta y no debe cerrarse.

Valor de Retorno

- “True” : Si se deben salvar los cambios
- “False”: Si no se deben salvar los cambios

Ejecución

Despliega un cuadro de diálogo que permite ingresar la configuración del paso. El usuario deberá seleccionar el nombre de Agreement de un combo cargado con los Agreements instalados en el sistema. El valor seleccionado es almacenado en el string “stXMLInfo” en formato XML.



7.5.2 DocStep

Invocada por el sitio para desplegar la información del template de paso.

Sintaxis

```
Public Function DocStep( ByVal stCaption As String, _
                        ByVal lnStepID As Long, _
                        ByVal stXMLInfo As String, _
                        ByVal TemplateID As Long, _
                        ByVal VersionID As String, _
                        ByRef QFlowDBConn As Variant
                        ) As String
```

Parámetros

Ídem EditStep

Valor de Retorno

Tabla HTML con las propiedades del paso para incluirlo en el sitio Web

7.5.3 ShowStep

Invocada por el sitio para desplegar la información del paso en ejecución. Construye en un string la tabla HTML con la información y la devuelve como resultado.

Sintaxis

```
Public Function ShowStep(ByVal TemplateID As Long, _
                        ByVal VersionID As String, _
                        ByVal TemplateStepID As Long, _
                        ByVal stXMLInfo As String, _
                        ByVal stCaption As String, _
                        ByVal FlowID As Long, _
                        ByVal StepID As Long, _
                        ByRef QFlowDBConn As Variant) As String
```

Parámetros

Ídem EditStep

Valor de Retorno

Ídem DocStep

7.5.4 ExecStep

Invocado por el motor para lanzar la ejecución de una instancia de paso. Se encarga de la construcción y despacho a BizTalk de documentos B2B_Request.

Sintaxis

```
Public Function ExecStep(ByVal TemplateID As Long, _
    ByVal VersionID As String, _
    ByVal TemplateStepID As Long, _
    ByVal stXMLInfo As String, _
    ByVal stCaption As String, _
    ByVal FlowID As Long, _
    ByVal ThreadId As Long, _
    ByVal StepID As Long, _
    ByRef objKernel As QSoftKRN.Kernel, _
    ByRef QFlowDBConn As ADODB.Connection
) As String
```

Parámetros

Ídem EditStep.

Valor de Retorno

- "R": Continúa inmediatamente la ejecución del proceso.
- "W": El paso queda en espera
- "E": Error que amerita detener el workflow
- "S": Estado de hibernación. no se esperan novedades pronto. El motor reduce a la mitad la frecuencia de interrogación. (Ver WaitStep)

Ejecución

- Construye el documento B2B_Request iterando sobre las variables del flow.
- Recupera la configuración del agreement de la base de datos.
- Instancia un objeto IInterchange y despacha el documento invocando al método Submit con el canal y el documento como parámetros.
- Inserta un registro en la tabla FlowStepsTo con la identificación del paso en ejecución.
- Inserta un registro en la tabla B2B_Instance
- Responde al motor el estado en que queda el paso:
 - Ready: Si la configuración del paso indica no esperar respuestas del Agente
 - Wait: Si se deben esperar respuestas del Agente
 - Error: Ante cualquier excepción. (Ej.: No poder despachar el documento a BizTalk)

7.5.5 WaitStep

Devuelve el estado de ejecución de una instancia de paso. Es invocada periódicamente por el motor mientras el paso esta en espera, para que este le indique la acción a seguir.

Sintaxis

```
Public Function WaitStep(ByVal TemplateID As Long, _
    ByVal VersionID As String, _
    ByVal TemplateStepID As Long, _
    ByVal stXMLInfo As String, _
    ByVal stCaption As String, _
    ByVal FlowID As Long, _
    ByVal ThreadId As Long, _
    ByVal StepID As Long, _
    ByRef objKernel As QSoftKRN.Kernel, _
    ByRef QFlowDBConn As ADODB.Connection
) As String
```

Parámetros

Idem EditStep

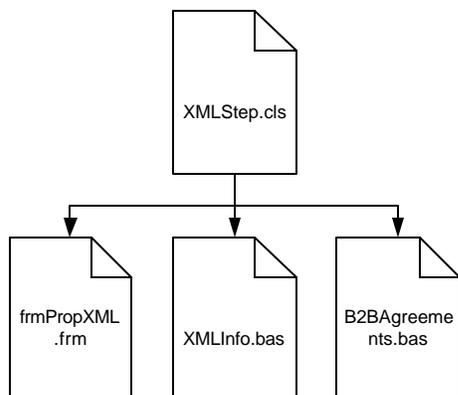
Valor de Retorno

- "R": Continúa la ejecución en el paso siguiente:
- "W": Queda en Espera
- "E": Error que amerita detener el workflow
- "S": Estado de hibernación, no se esperan respuestas pronto
- "N": Queda en estado de espera, con nivel de recordatorio.
- "A": Queda en Espera con nivel de Alerta
- "D", "O": Sale por el conector de Timeout

Ejecución

- Recupera de la tabla "FlowStepsTo" el registro asociado al paso y decide el valor de respuesta de acuerdo al valor de progreso:
 - "Terminado": Responde "R"
 - "No Iniciado" o "En progreso": Responde "W"
 - "Cancelado": Responde "E"

7.5.6 Mapa del Código Fuente



- XMLStep.cls: Implementa el objeto XMLStep con las funciones del paso.
- FrmPropXML.frm: Formulario de edición del paso. Invocado por EditStep.
- B2BAgreements.bas: Funciones auxiliares para el manejo de Agreements.
- XMLInfo.bas: Funciones auxiliares para el manejo de la configuración del paso en XML.

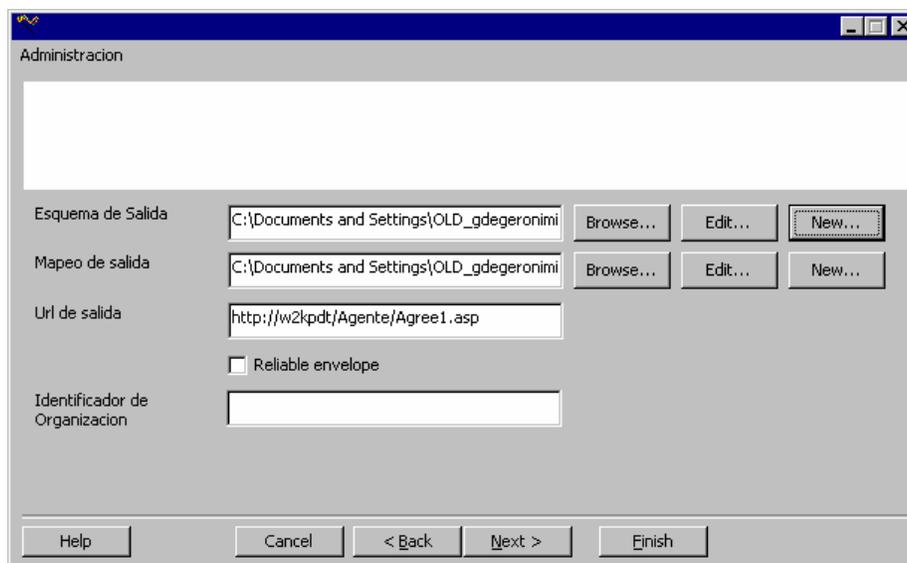
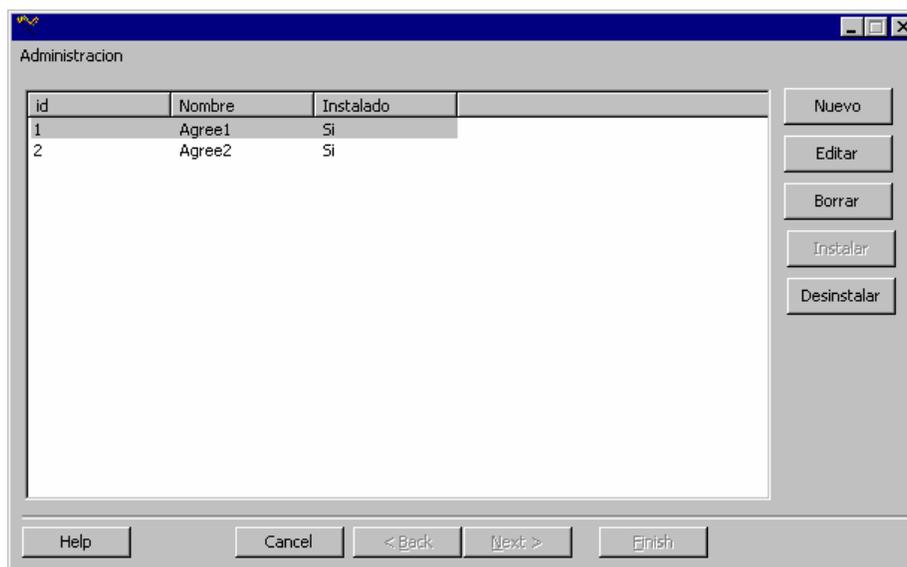
7.6 B2B Config Wizard

El “B2B Config Wizard” es la herramienta de administración de Agreements. Las funciones del mismo son:

- Mantenimiento de Agreements: Altas, Bajas y Modificaciones de los Agreements registrados en el sistema
- Instalación / Desinstalación: Permite instalar / desinstalar los Agreements, generando / eliminando la configuración necesaria en BizTalk para soportar el Agreement.

7.6.1 Interfase

La interfase de la herramienta fue construida con el “Add-In” “Wizard Manager”, provisto con Visual Basic. El mismo simplifica el desarrollo de aplicaciones tipo Wizard, encargándose de la lógica de las transiciones entre cada paso de configuración (“Adelante”, “Atrás”, “Cancelar”, etc.).



7.6.2 BizTalk Configuration Object Model

La herramienta “B2B Config Wizard” accede a la configuración de BizTalk mediante la biblioteca “BizTalk Configuration Object Model”. La misma ofrece un conjunto de objetos que permiten configurar BizTalk de forma programática.

La “BizTalk Configuration Object Model” incluye un objeto COM para cada objeto de configuración de BizTalk: BizTalkChannel, BizTalkPort, BizTalkDocument, BizTalkOrganization y BizTalkEnvelope. Cada uno de ellos implementa la interfase *IbizTalkBase*, la cual define de forma común los métodos para altas, bajas y modificaciones.

Además de estos, la biblioteca incluye la clase “BizTalkConfig”, la cual contiene la configuración completa de BizTalk. La misma contiene todos los objetos configurados en colecciones según el tipo (Channel, Port, Document, etc.).

7.6.3 Instalación de un Agreement

La rutina de instalación de un Agreement debe generar los objetos de configuración que soportaran el Agreement y configurarlos de acuerdo a los parámetros del mismo.

Los pasos a seguir para instalar el Agreement “XX” son:

1. Crear una organización “**XX_Org**”.
2. Crear un puerto “**XX_OutPort**” con destino a “**XX_Org**”, asignándole:
 - a. Dirección de destino: URL donde recibe las solicitudes el Agente Externo.
 - b. Protocolo HTTP.
 - c. Envelope: Según configuración del Agreement, sin envelope o con el envelope “**RELIABLE**”, creado durante la instalación del producto.
3. Copiar los esquemas de documento del Agreement a la carpeta en el repositorio de documentos de BizTalk.
4. Crear el documento “**XX_Request**”, pasándole la URL del esquema de Request copiado en el repositorio.
5. Crear el canal “**XX_OutChannel**” con la siguiente configuración:
 - a. Origen: Aplicación “QFL” de la “Home Organization”¹⁷, creada durante la instalación del producto.
 - b. Documento de Entrada: “QFL_B2B_Request”, creado durante la instalación del producto.
 - c. Documento de Salida: “**XX_Request**”.
 - d. Mapeo: URL del mapeo “XX_OutMap.xml” en el repositorio de BizTalk.
 - e. Puerto: **XX_OutPort**

Si se esperan respuestas del Agente Externo:

6. Crear el documento “**XX_Response**”, pasándole la URL del esquema de Response copiado en el repositorio.
7. Crear el canal “**XX_InChannel**” con la siguiente configuración:
 - a. Origen: **XX_Org**.
 - b. Documento de Entrada: **XX_Response**.
 - c. Documento de Salida: “**QFLOWRESPONSE**”
 - d. Mapeo: URL del mapeo “XX_InMap.xml” en el repositorio de BizTalk.
 - e. Puerto: “Port_to_QFlow”, creada durante la instalación del producto.

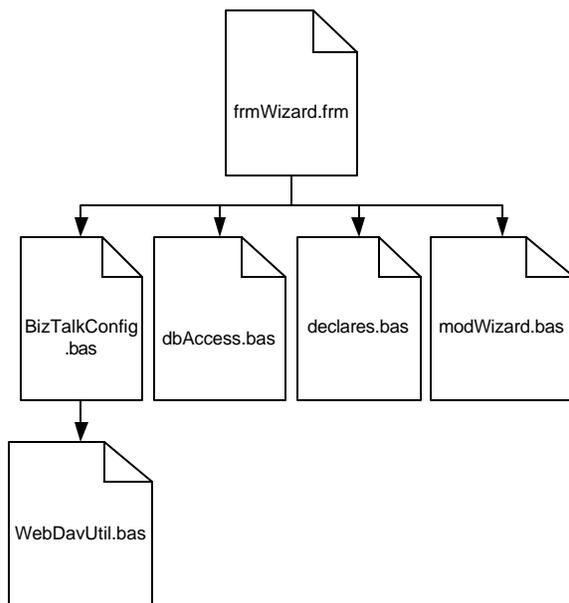
Luego, se registran los handles de los objetos en la tabla “B2B_BizTalkObjects” para su futura identificación.

7.6.4 Desinstalacion de un Agreement

Para desinstalar un Agreement, solamente se necesita recuperar los handles de los objetos configurados de la tabla “B2B_BizTalkObjects”, y borrarlos del servidor.

¹⁷ La Home Organization representa la organización local. La misma no puede configurarse como destino de documentos. Los mismos deben ser dirigidos a las aplicaciones configuradas en la misma.

7.6.5 Mapa del Código Fuente



- FrmWizard.frm: Formulario con la interfase de la herramienta.
- ModWizard.bas: Código generado por el Wizard Manager. Lógica de la interfase.
- BizTalkConfig.bas: Genera / Elimina los objetos de configuración en BizTalk.
- WebDavUtil.bas: Permite acceder al repositorio de documentos de BizTalk.
- dbAccess.bas: Maneja el acceso a los Agreements en la base de datos.
- declares.bas: Declaración de funciones del sistema de acceso a la Registry.

7.7 B2BGate

B2BGate es una página ASP, que permite recibir documentos por HTTP de los Agentes Externos. La misma identifica el Agreement al que están asociados los documentos a partir de un identificador de instancia que se pasa como parámetro del comando POST. La página debe ser instalada en el servidor Web.

Ejecución

- 1) Recupera el documento XML del comando POST.
- 2) Carga el mismo en un objeto DOM, verificando que sea bien formado.
- 3) Recupera el parámetro "IxId" del "Query string"
- 4) Obtiene el nombre del canal, haciendo un "join" de las tablas B2B_Instance y B2B_Agreements y filtrando por IxId
- 5) Instancia un objeto IInterchange e invoca al método "Submit" con el documento y el canal como parámetros.

8 Testing

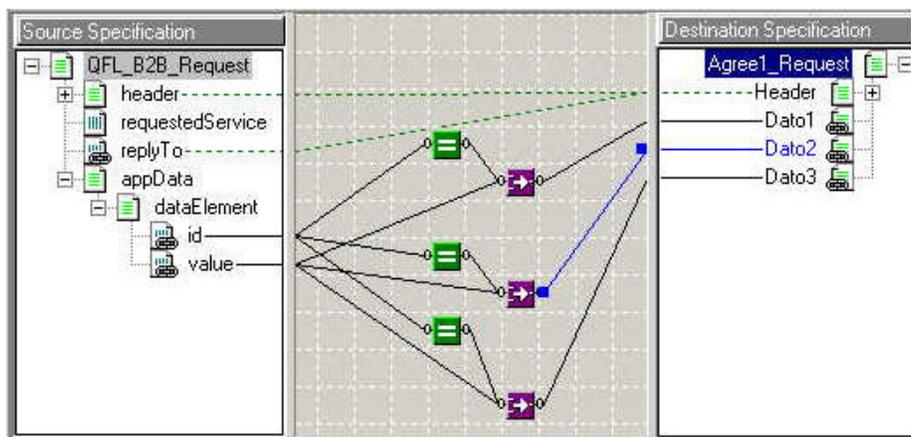
El Testing del módulo fue hecho implementando un flujo con dos pasos de B2B invocados secuencialmente. Cada paso fue asociado un Agreement distinto: “Agree1” y “Agree2”, los cuales fueron implementados por páginas ASP.



El paso de pregunta, permite ingresar valores para los datos de aplicación “Dato1”, “Dato2” y “Dato3”, los cuales serán pasados como parámetro en ambos pasos.

8.1 Agree1

El Agree1 fue configurado para continuar inmediatamente. El Agente Externo asociado, “Agree1.asp”, recupera la información del documento y simplemente la registra en un archivo de Log.



8.2 Agree2

El Agree2 utiliza un esquema de Request similar al de Agree1. A diferencia del mismo, espera a que el Agente Externo responda la solicitud para continuar con el flujo. El Agente Externo asociado, “Agree2.asp” incrementa el valor de Dato1, multiplica Dato2 por 2 y genera un registro en un archivo de Log.

La estructura del documento de Response es similar al documento de Request. Los valores “Response” y “Progress” del documento QFLOWRESPONSE son fijados a “Finalizado”, “100%” en el mapeo, por lo que la primer respuesta recibida terminara el paso.

Las respuestas son recibidas por el script “B2BGate.asp”.

9 Conclusiones

El “objetivo de los objetivos” planteados para el proyecto, era investigar la potencialidad de Microsoft BizTalk Server, para resolver el problema de integrar Q-flow con aplicaciones externas. En este sentido, y tal como se indica en la propuesta del proyecto, no se esperaba obtener una solución acabada, pronta para la puesta en producción, sino un prototipo que probara la factibilidad del enfoque propuesto.

El proyecto cumple con los objetivos planteados: Se logra una evaluación práctica del producto, que permite medir el verdadero alcance de la herramienta BizTalk, y se presenta una solución al problema planteado, implementando un prototipo que permite validar el diseño propuesto.

La evaluación de BizTalk Server permite concluir que si bien la herramienta ofrece gran potencia, el producto muestra cierta falta de madurez que es característico de las primeras versiones. Estas pequeñas carencias no le quitan funcionalidad, pero hacen que la implementación de soluciones exija mayor esfuerzo del que debería:

- Las herramientas de configuración presentan meros listados de los objetos, donde no es posible visualizar la relación entre ellos. Los mismos no pueden ser agrupados en conjuntos, lo cual es comparable a no poder organizar en bases de datos las tablas de un DBMS.
- El sistema ofrece pobres o nulas herramientas de “debugging”, lo cual hace muy difícil identificar la fuente de errores en el procesamiento de documentos. Ante una falla, el error puede ser causado por la aplicación, el canal, el puerto, o los documentos asociados.

La solución presentada explota bien los beneficios de BizTalk. Delega completamente el manejo de protocolos y la conversión entre esquemas de documento, permitiendo que la programación se concentre en la lógica del paso, evitando “reinventar la rueda” y logrando una solución mas **simple** y mucho mas **potente** que las otras alternativas.

Aunque el prototipo fue limitado en implementación, tanto el análisis como el diseño observan toda la funcionalidad de BizTalk: Si BizTalk lo soporta, lo soporta el módulo. No es de esperar que la implementación final del módulo se enfrente a nuevos problemas causados por la interacción con BizTalk.

9.1 Trabajos Futuros

- Inicio de flows desde el exterior: Esta funcionalidad fue analizada y diseñada, pero no se llegó a implementar por ser similar al procesamiento de documentos de respuesta y no aportar al prototipo.
- Control de permisos sobre las variables: Se deben insertar en el documento B2B_Request solamente las variables con permisos de lectura o edición.
- Otros protocolos: El prototipo permite configurar intercambios solamente a través de HTTP. Esto se limito para simplificar el proceso de instalación de los Agreements.
- Otros deployments: Todos los aspectos de instalación están “Hard Coded”. El prototipo exige que BizTalk y Q-flow estén instalados en el mismo equipo, lo cual resulta muy restrictivo para un entorno de producción. Se deberá permitir configurar la ubicación del servidor BizTalk como parámetro de instalación.

Apéndices

Apéndice A – Tecnologías de Internet

A.1 XML

Aunque formalmente sea solamente un estándar para definir lenguajes de demarcación de textos, XML(eXtensible Markup Language) representa un conjunto de tecnologías que habilitan el intercambio de información estructurada de forma estándar y portable sobre Internet.

Para hacer posible el intercambio de documentos estructurados, es necesario que los participantes acuerden un lenguaje de demarcación de textos que haga explícita la estructura del mismo. Asimismo, deberá formalizarse la semántica de la estructura compartida (Observe por ejemplo, que la fecha 8-7-97 admite dos interpretaciones).

Históricamente el problema ha sido resuelto desarrollando estándares de demarcación propios de cada industria (ANPA 1312, GenCode, GML). Teniendo un conjunto limitado de tags, todos estos estándares debían negociar entre generalidad y claridad, teniendo como resultado un enorme conjunto de formatos cada uno especializado en un pequeño grupo de documentos.

Con la intención de atacar el problema de la diversidad es que la ISO desarrolla SGML (Standard Generalized Markup Language). En este se define una sintaxis de demarcación estándar y un lenguaje para declarar la estructura del documento (DTD). La inclusión de esta facilidad permitió separar los datos de los procesos que actúan sobre ellos. Por primera vez se podían escribir rutinas que actuaran sobre estructuras de datos variables.

SGML fue desarrollado pensando en máquinas lentas y costosas. Implementa complejas mecanismos de minimización que resultan en parsers extremadamente complejos y desarrollos costosos. Es entonces que se decide desarrollar una versión liviana de SGML para la Web: XML.

A.1.1 Sintaxis

- **Declaración:** Un documento XML bien formado comienza con la declaración opcional:

```
<?xml version="1.0" ?>
```

Pueden incluirse también atributos de codificación e información extra para el parser.
- **Estructura:**
 - Un documento XML consiste en texto y tags.
 - Los tags se distinguen del texto por estar encerrados en <>.
 - Cada elemento esta delimitado por un tag de apertura y uno complementario de cierre. El tag de cierre contiene el texto del tag de apertura precedido de un "/"
 - Los elementos tienen atributos asociados, ente los cuales se encuentra el nombre del mismo.
 - Los datos entre tags son el contenido del elemento. Puede ser texto plano así como otros elementos con sus propios tags.

```
<ej> Info de ejemplo </ej>
```
- **Elemento Raíz:** Todo documento XML debe tener una única raíz. El primer tag encontrado por el parser será considerado raíz, y se dará por terminado el documento al llegar al tag de cierre correspondiente.
- **"Case Sensitivity":** Los nombres de elementos y atributos en XML son Case Sensitive.
- **Anidamiento:** Los elementos en un documento correctamente formado deben mantener una jerarquía arborescente. El tag de cierre de un elemento debe aparecer antes del tag de cierre del objeto que lo contiene.
 - Mal: <a> DATOS
 - Bien: <a> DATOS

- **Caracteres especiales:** Ya que los caracteres especiales de demarcación(<, >, ‘, “, /) no pueden aparecer dentro del contenido, éstos deben ser sustituidos por cadenas de referencia.

Referencia	Carácter
<	<
>	>
&	&
'	'
"	"

- **Atributos:** Cuando no alcanza con el nombre para describir un elemento, puede agregarse información acerca del mismo con atributos. Los atributos aparecen como pares nombre–valor dentro del tag de apertura

```
<ej atrib1="Valor1" atrib2="Valor2"> Datos </ej>
```
- **Elementos vacíos:** En los casos en donde se desea insertar elementos sin contenido, puede evitarse el uso de dos tags <ej> </ej> reemplazando ambos por el tag <ej/>
- **Comentarios:** Los comentarios pueden aparecer en cualquier parte del texto mientras aparezca dentro de

```
<!-- Comentario -->
```

Se dice que un documento XML que cumple con las reglas anteriores es “**bien formado**”; cualquier parser podrá interpretar el documento sin problemas.

A.1.2 Esquemas

La clasificación de “bien formado”, no nos permite a priori determinar si un documento XML podrá ser procesado por una aplicación. Por mas que sepamos que podrá ser “parseado”, nada nos indica que el contenido del mismo sea el que espera recibir la aplicación. No hay forma de predecir la forma en que el remitente organiza la información dentro del documento, o los tags que la delimitan.

Para asegurar que un documento sea procesable, tendremos que poder comparar su estructura contra una definición de la misma. Para esto se definen los Esquemas XML.

Un Esquema XML define:

- Los elementos que pueden aparecer en un documento
- Los atributos que tiene asociado un elemento
- Que elementos son hijos de otro.
- La secuencia en la que pueden aparecer los elementos
- El número de hijos que puede tener un elemento
- Si un elemento debe ser vacío o si contiene texto
- Valores por defecto para los atributos

Si un documento XML tiene un esquema asociado, y cumple con él, se dice que es “válido”.

A.1.3 Document Object Model (DOM)

El DOM expone un documento XML como una estructura arborescente en memoria, ofreciéndole al programador acceso a los elementos de la misma forma que accede a estructuras en memoria. DOM es una recomendación de la W3C. En lo que sigue nos referiremos a la implementación de Microsoft.

Microsoft DOM es parte del parser XML. Se distribuye en la biblioteca MSXML2.dll . Esta se registra como un objeto COM bajo el nombre “MSXML2.DOMDocument” o puede accederse mediante automatización como “Microsoft.XMLDOM”

Esencialmente la API de DOM consiste en un árbol cuyos nodos, dependiendo de su tipo, representan elementos, atributos, texto, etc. El siguiente código JavaScript muestra un ejemplo de uso típico.

```
var objDocument = new ActiveXObject("MSXML2.DOMDocument");
objDocument.async = false; // especifica modo asíncrono
objDocument.load("mi_archivo.xml");
var rootElem= objDocument.documentElement;
...
...
```

A.1.4 eXtensible Stylesheet Language

“Estilizar” (styling) un documento significa convertir el mismo a un formato adecuado para cierta audiencia. Ya que esta audiencia puede ser muy diversa, es probable que debamos aplicar muchos estilos al mismo documento, para acomodarlo a las necesidades de los destinatarios de la información.

El proceso de estilizar un documento se divide en dos partes:

1. **Transformación** del esquema de origen al esquema del destinatario, y
2. **Formateo** de la información para que sea procesable por el destinatario.

Veamos un ejemplo:

- El documento original:

```
01 <?xml version="1.0"?>
02 <purchase id="p001">
03   <customer db="cust123"/>
04   <product db="prod345">
05     <amount>23.45</amount>
06   </product>
07 </purchase>
```

- El resultado, desplegado (rendered) para una persona y formateado para un Web Browser (HTML):

```
01 <p>From: <i>(Customer Reference) <b>cust123</b></i>
02 </p>
```

Para describir este procesamiento de forma estándar, la W3C elabora **XSL**. De forma de conservar la independencia entre ambas etapas, XSL fue dividida en dos recomendaciones: XSLT y XSL.

A.1.5 XSLT

XSLT es un lenguaje principalmente declarativo. Esto permite concentrarse en el resultado deseado y no en la forma en que se logrará.

Una “especificación de transformación” en XSLT debe proporcionar:

- Las reglas de cómo se formula cada componente en el lenguaje destino, y
- Las condiciones en las fuentes que determinan qué componente es agregado a continuación a la salida.

En XSLT la información se manipula como árboles de nodos abstractos. XSLT incluye construcciones que permiten iterar sobre la estructura de nuestras fuentes. La información que estamos procesando puede ser dispuesta en el orden y cantidad que se quiera, y podemos referirnos a ella en mas de una oportunidad si el resultado de la transformación lo requiere.

Técnicamente, podemos mencionar que XSLT es “Turing Complete”, lo cual garantiza que cualquier algoritmo puede (en teoría) ser implementado con los constructores que se disponen.

A.2 SOAP(Simple Object Access Protocol)

A.2.1 Introducción

SOAP, como su nombre lo indica, es un protocolo que permite el acceso a objetos distribuidos a lo largo de la red. Cuando se diseña un sistema, las funciones del mismo son distribuidas en los nodos del sistema que mejor las albergan. Tales funciones son materializadas en tiempo de implementación como procedimientos que son accedidos de forma remota por RPC. Habitualmente, un conjunto de procedimientos relacionados se agrupan en objetos, utilizando el estándar de moda (CORBA, DCOM, etc.). El problema de esto, es que las técnicas para implementar estos objetos dependen en mayor o menor grado de la plataforma: Un objeto corriendo en una plataforma no es fácilmente accesible desde otras. Por otra parte, acceder a objetos que están distribuidos sobre Internet es aún mas difícil. Las tecnologías usadas hasta ahora exigen el uso de puertos especiales para transmitir los datos, lo cual choca la mayoría de veces con las políticas de seguridad de las organizaciones, las cuales evitan cualquier tráfico que no sea al puerto 80 (HTTP).

De lo anterior se desprende que existe la necesidad de:

- Un mecanismo que sea **independiente de plataforma** para invocar métodos ubicados en sistemas distribuidos.
- Un método **Firewall - friendly** de hacer RPC sobre Internet.

Es deseable también, que la solución a estos problemas sea **ligera y fácil** de entender.

SOAP resuelve estos requerimientos utilizando:

- XML para lograr independencia de plataforma.
- HTTP como protocolo de transporte (entre otros) para cruzar los Firewalls sin riesgos de seguridad.

A.2.2 Características

- Transmisión de mensajes en un estilo pedido / respuesta (Request / Response).
- Esquemas definidos para invocar un procedimiento, pasarle parámetros y devolver el resultado
- Esquema para retornar errores de forma estándar
- Medio de transporte no especificado. La forma en que se transmiten los mensajes no es parte del protocolo.
- No se especifica cómo, luego de recibido un mensaje, se crea la instancia del objeto que ejecutará el procedimiento.

A.2.3 Estructura del Mensaje SOAP



A.2.4 Diálogo Cliente – Servidor

La secuencia típica para la invocación de un procedimiento consistiría en los siguientes pasos:

1. [CLI] Construir el mensaje SOAP, el cual contendrá el nombre del procedimiento, y los parámetros.
2. [CLI] Enviar el mensaje
3. [SERV] Recibir el mensaje
4. [SERV] Instanciar el objeto referido en el mensaje
5. [SERV] Invocar el procedimiento, pasándole los parámetros incluidos en el mensaje.
6. [SERV] Al retornar el procedimiento, construye un documento de respuesta y se envía al cliente. EL mensaje contiene el nombre de la respuesta y el valor de retorno.

A.3 BizTalk Framework

A.3.1 Introducción

La iniciativa BizTalk fue propuesta por Microsoft, como forma de resolver el intercambio de documentos de negocio en un ambiente de e-commerce. BizTalk ofrece una forma elegante de intercambiar documentos de negocio a través de Internet.

BizTalk promueve el uso de XML en las transacciones B2B al proponer una estructura para crear documentos de negocio en XML. Dada la enorme flexibilidad de XML, es muy poco probable que dos desarrollos independientes de un mismo documento terminen siendo compatibles. XML nos asegura que el destinatario podrá validar el documento y recuperar los datos de su estructura, pero no asegura que este sea capaz de procesarlos, convirtiendo datos en información. Para esto último, es todavía necesario acordar cual será el contenido del documento: Como se identificará la información, que estructura seguirá, etc.

Para resolver estos problemas, BizTalk define un conjunto de tags, guías de diseño y soporte para el intercambio de esquemas estándar propios de cada industria. Con todo esto, uno puede determinar fácilmente que tipo de documento recibió y que información contiene.

Disponer de esta plataforma, permite que los desarrolladores se enfoquen en resolver los requerimientos de información de cada industria y eviten los detalles comunes a todas.

A.3.2 Componentes

La iniciativa BizTalk se divide en tres aspectos:

- Estándar de Documento y Mensaje BizTalk
- BizTalk.org
- BizTalk Framework Compliant (BFC) Server

A.3.3 Documento BizTalk

Un documento BizTalk es un mensaje SOAP 1.1, donde el Body del mensaje contiene uno o varios documentos de negocio, y el Header contiene entradas específicas de BizTalk que definen la semántica de manipulación del documento. En este sentido, un documento de negocio es un documento XML bien formado conteniendo información propietaria del negocio.

Los componentes del Header son:

Tag Name	Mandatory	Kind	Type	Occurs
Del i very	Yes	element	Compl exType	Once
Properti es	Yes	element	Compl exType	Once
Mani fest	No	element	Compl exType	Once
Process	No	element	Compl exType	Once

- **Delivery:** Contiene información acerca del origen y destino del documento BizTalk. Opcionalmente puede incluir información para entrega segura. Se compone de los siguientes elementos:

Tag Name	Mandatory	Kind	Type	Occurs
To	Yes	element	ComplexType	Once
From	Yes	element	ComplexType	Once
Reliability	No	element	ComplexType	Once

- **Properties:** Contiene información descriptiva y de identificación del documento. Sus componentes son:

Tag Name	Mandatory	Kind	Type	Occurs
Identity	Yes	element	uriReference	Once
SentAt	Yes	element	timeInstant http://www.w3.org/TR/xmlschema-2/#timeInstant	Once
ExpiresAt	Yes	element	timeInstant	Once
Topic	Yes	element	uriReference	Once

- **Manifest:** Permite catalogar el conjunto de documentos que son cargados por el mensaje BizTalk. El tag, que es opcional, permite hacer chequeos de integridad y consistencia, y verificaciones como ser firmado digital.

Tag Name	Mandatory	Kind	Type	Occurs
reference	Yes	Element	complexType	One or more times

- **Process:** Incluye información acerca del proceso que ofrece soporte al documento.

Tag Name	Mandatory	Kind	Type	Occurs
Type	Yes	Element	uriReference	Once
instance	Yes	Element	uriReference	Once
handle	No	Element	uriReference	Once

A.3.4 Servidor BizTalk

Un servidor BizTalk es un sistema capaz de leer e interpretar documentos BizTalk. Según la especificación, esto es lo único que se necesita para cumplir con la norma BFC. Mencionaremos a continuación, los requerimientos que son deseables para un sistema de este tipo:

- **Entrega confiable:** El servidor BizTalk debe ser capaz de realizar la entrega de documentos mediante los canales mas comunes (HTTP, FTP, SMTP, etc.). Algunos de estos, son por naturaleza no confiables, por lo que el servidor BizTalk deberá implementar las rutinas necesarias para asegurar la misma.
- **Seguridad:** Que el transporte de documentos sea hecho a través de redes públicas (Internet) presenta un riesgo de seguridad para el sistema. Para solucionar el problema, debemos ser capaces de manejar canales de comunicación seguros. La especificación BT Framework 2.0 ofrece las opciones de usar S/MIME o PKCS, aunque puede optarse por cualquier otra plataforma de seguridad, como ser HTTPS.

- **Ruteo:** La funcionalidad mas importante de un servidor BizTalk es entregar nuestros documentos en el sistema del destinatario. Esto puede ser tan simple como hacer un POST a un servidor Web por HTTP, o implicar complejos pasos de ruteo para lograr atravesar Firewalls u otras medidas de seguridad.
- **Definición de procesos:** Luego de que nuestro sistema recibe un documento y descubre el propósito del mismo y la información que incluye, debe realizar cierto proceso para cumplir con él. Este proceso exige típicamente la interacción con el resto de los sistemas de Back-end de la organización. Es deseable entonces, que el servidor BizTalk ofrezca las herramientas necesarias para diseñar estos procesos; típicamente como un grafo de ejecución (à la SDL).
- **Comunicación Síncrona y Asíncrona:** La forma mas simple de entregar documentos de forma confiable es realizar la transferencia y esperar “conectado” la respuesta. Este modo, no obstante, puede dar lugar a cuellos de botella ante alta carga. Por esta razón, es que un servidor BizTalk debería ofrecer mecanismos de acuse asíncrono de recepción de documentos.
- **Encolado:** Un servidor BizTalk debe ser capaz de entregar un documento a una cola de procesamiento en el servidor remoto y de forma asíncrona consultar por el estado de procesamiento del documento.
- **Soporte Batch:** Para poder implementar de forma elegante ciertos acuerdos de negocio, es deseable que el sistema soporte la definición de procesos batch.
- **Monitoreo de Actividad:** En ambientes de procesamiento distribuido, es importante poder intercambiar información acerca del progreso de las tareas en cada sistema. En nuestro caso deberíamos observar:
 - Tracking de documentos.
 - Éxito en la entrega de documentos.
 - Logeo de información propia de los documentos para posterior reporte de actividad.
- **Administración de Acuerdos de Negocio:** Debemos ser capaces de definir políticas de procesamiento según quien sea el solicitante. En particular, debemos diferenciar los remitentes con los que tenemos algún tipo de acuerdo de los que no.
- **Escalabilidad:** Nuestro sistema debe ser suficientemente fácil de escalar como para que esto no sea freno para incorporar nuevos acuerdos de negocio.
- **Transformación de documentos:** Es muy probable que en la práctica terminemos recibiendo documentos de agentes de diversas industrias, por lo que inevitablemente terminaremos manejando múltiples esquemas para nuestra información. Un sistema de BizTalk debe estar preparado para enfrentar estos problemas y ofrecernos herramientas para mapear múltiples esquemas entre si.

Apéndice B – WfMC: Modelo de Referencia

B.1 Introducción

A pesar de la diversidad de escenarios en los que se desarrollan, los sistemas de WF conservan muchos aspectos comunes. Esto despierta el interés por el desarrollo de estándares, que eviten que esfuerzos descoordinados resulten en sistemas incompatibles. Con este objetivo en mente, es que surge la “Workflow Management Coalition”. El objetivo de la misma es coordinar esfuerzos entre los actores de la industria para lograr sistemas de Workflow capaces de interoperar a todo nivel.

El eje del trabajo de la WfMC es el “Workflow Reference Model”. El mismo funciona como un análisis de requerimientos normalizado del problema. Cubre conceptos básicos, terminología y estructura general de un sistema de Workflow. Define interfases con los agentes externos y plantea los posibles escenarios de interoperabilidad. El modelo se complementa con la documentación específica de cada interfase.

B.2 Sistemas de Workflow

B.2.1 Que es un Workflow?

Un Workflow es la **automatización total o parcial de un proceso de negocio** donde **actividades, información y documentos** son pasadas entre los participantes según **reglas preestablecidas**. La definición de un Workflow debe especificar:

- Las tareas que deberán ser realizadas y la **secuencia** en que serán ejecutadas (las reglas de flujo),
- La forma en que serán asignadas las **tareas a los recursos (humanos)** disponibles,
- La **información** que debe estar disponible y ser producida en cada etapa,
- Las **herramientas** que podrán ser utilizadas para ejecutar cada tarea.

El ciclo de vida de un Workflow puede ir de minutos hasta días o meses, dependiendo de la complejidad y la duración de las actividades que lo constituyen. Mientras que en teoría, un Workflow podría ser implementado manualmente, lo mas común es que se desarrollen sobre medios electrónicos.

B.2.2 Áreas Funcionales

Un Sistema de Workflow es un sistema capaz de **definir, crear y manejar la ejecución de workflows a través del uso de software**. Según el enfoque de la WfMC, un sistema de Workflow puede dividirse en tres áreas funcionales:

- **Tiempo de Construcción:** Definición y modelado del workflow. Se refiere a la “programación” del sistema para representar el proceso objetivo. Debe dar como resultado una versión digital del proceso de negocio.
- **Control de Ejecución:** Administración de los workflows en ejecución, secuenciamiento de las actividades que componen cada proceso y asignación de tareas a recursos. Estas funciones se concentran en un componente llamado “**Motor**” del sistema de Workflow.
- **Interacción con el Medio:** Interacción con Recursos Humanos (personas) o tecnológicos (aplicaciones, herramientas) para ejecutar cada etapa del mismo. Pasaje de datos de/a aplicaciones externas.

B.2.3 Tiempo de Construcción

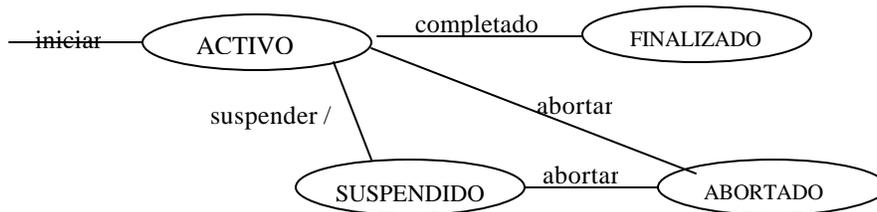
Las funciones de Tiempo de Construcción tienen como fin llevar el proceso del mundo real a un modelo formal, y éste luego en uno procesable por computadora. Una definición de proceso se compone de un conjunto de **pasos de actividad** y de **reglas de progreso**. Lo más común es que los workflows sean formalizados como diagramas de flujo que se construyen mediante herramientas visuales. De esta manera, el “lenguaje de definición” del Workflow es cercano al modelo “en papel”, y la formalización puede ser realizada por el propio administrador de procesos. Finalmente, la representación gráfica del proceso es “compilada” en otra más adecuada para la ejecución.

B.2.4 Tiempo de Ejecución

Las funciones de Tiempo de Ejecución son interpretar las definiciones de proceso, determinando la secuencia de ejecución de las etapas y despachar las tareas que genere la interpretación a los recursos disponibles para que las ejecuten. Internamente maneja el concepto de “Instancia de proceso”, el cual corresponde a un trámite, o expediente iniciado en la organización.

El flujo de control de un proceso puede verse como una máquina de estados, que va cambiando en respuesta a eventos externos y decisiones de control, tomadas internamente por el Workflow.

La culminación de las actividades que constituyen un proceso, implica la culminación del mismo.



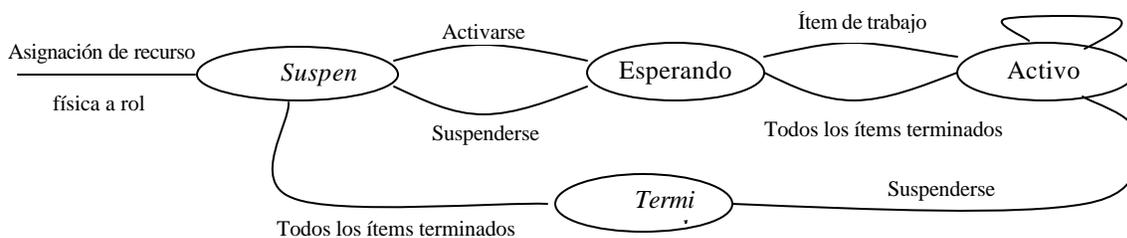
B.2.5 Interacción con el Medio

Un sistema de Workflow es por naturaleza un componente más de un sistema de mayor tamaño. Aislado de los demás componentes, no ofrece gran utilidad. Es en la interacción con otros recursos (los verdaderos objetivos) que desarrolla completamente su potencialidad. Por ejemplo, una empresa puede usar un sistema de Workflow para automatizar sus procedimientos de Calidad. En este caso, los verdaderos recursos del sistema son los especialistas en Calidad, las bases de datos, los sistemas de producción y los funcionarios que deben trabajar con el sistema. El Workflow es solo el “orquestrado” de todo el conjunto.

Es por esto, que poder interactuar de forma elegante con el resto de los actores del ambiente es fundamental para el éxito de un sistema.

- **Interacción con RR.HH:** Acceso Web, e-mail online y offline
- **Interacción con IT:** Definición de pasos de :Invocación a sistemas externos, consultas sobre BD, scripting, pasaje de datos a herramientas de escritorio, etc.
- **Interacción con sistemas “de papel”:** Escaneo/Impresión de documentos, faxeo, etc. (Ej.: documentos legales en sistemas de expediente electrónico, envío de solicitudes de compra por fax).

Los recursos disponibles para el sistema pueden modelarse con la siguiente máquina de estados:



B.3 Workflow Reference Model

El Workflow Reference Model (WfRM) es un modelo de arquitectura de Workflow genérica. El objetivo del mismo es disponer de una estructura en donde puedan formalizarse estándares de interoperabilidad para la industria. El mismo esta estructurado en torno a un conjunto de interfases, cada una de las cuales identifica un área de interoperabilidad potencial:

- **Interfase 1:** Definición de procesos.
- **Interfase 2:** Clientes del Workflow
- **Interfase 3:** Aplicaciones externas
- **Interfase 4:** Interoperabilidad con otros sistemas de Workflow
- **Interfase 5:** Administración y Monitoreo.

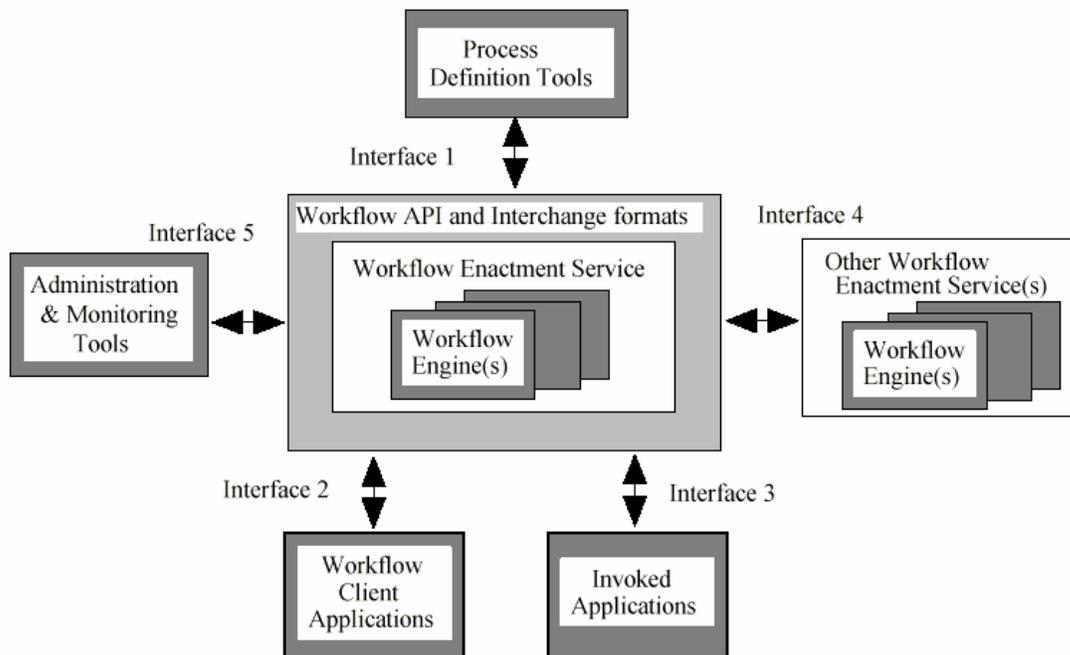


Fig 6 Workflow Reference Model - Components & Interfaces

B.4 *Enactment Service*

El **Enactment Service** resume la lógica de orquestación del sistema. Sus tareas son:

- Interpretar las definiciones de Workflow
- Instanciación y activación de procesos,
- Despacho de las tareas a recursos (RR.HH. y de IT) para su ejecución,
- Pasaje de información entre etapas de un proceso (incluido control de acceso),

La interpretación de definiciones consiste en recuperar una estructura tipo grafo de algún medio de almacenamiento y recorrerla de acuerdo a la semántica de cada nodo. Estas funciones dependen directamente de la Interfaz 1.

El despacho de tareas a recursos es resuelto mediante las interfases 2 y 3. Conceptualmente, cada recurso tiene asociado una **lista de tareas pendientes** (Work List). Para despachar una tarea a un recurso, el ES genera una **instancia de tarea** (Work Item) y la inserta en la cola del recurso. A medida que este termina las tareas, retira la siguiente de la cola, la procesa y devuelve el resultado al ES, el cual entonces avanza el proceso al siguiente paso.

El pasaje de información ente etapas de un proceso es resuelto por el sistema mediante el concepto de **variables del proceso**. Las variables del proceso son unidades de información tales como: Nombre y apellido del cliente, fecha de nacimiento, monto de la deuda, un archivo, etc. Se dividen en tres categorías:

- **Variables internas** (o de control): Información solo accesible por el motor. Ej.: Identificadores de instancia
- **Variables relevantes**: Información del proceso que incide en la secuencia de ejecución del proceso. Ej.: Monto de la deuda: Si es mayor que ZZZ, se requiere autorización especial, y sino no.
- **Variables externas** (o de aplicación): Información que es distribuida por el motor, pero que no es interpretable por el mismo, y por lo tanto no afectan el flujo de procesamiento. Ej. Un campo de observaciones.

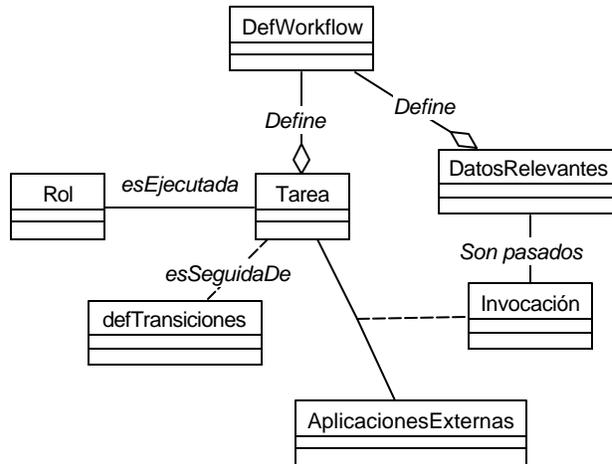
B.5 *Módulo de Definición de Procesos*

El módulo de definición de procesos tiene como objetivo la construcción de una versión del workflow interpretable por el motor del sistema. Según la complejidad de la herramienta, puede cubrir el análisis, modelado y documentación del proceso, o limitarse a la edición de la estructura final.

El formato de los datos intercambiados por la herramienta y el motor son habitualmente propietarios. Estos dependen en gran medida de aspectos de implementación, por lo que es difícil contemplar estándares. Para ofrecer algún nivel de independencia entre el sistema y las definiciones, (por ejemplo, para reutilizar las definiciones de un sistema en otro), las herramientas incluyen funciones de exportación / importación a formatos estándar, aptos para el intercambio pero no para interpretación. Estas últimas son quienes constituyen la interfase de definición de procesos (Interfase 1).

B.5.1 Interfase 1

La Interfase 1 consiste de un formato de intercambio (un meta-modelo de WF) y una API para el importación /exportación de definiciones. La misma debe ser capaz de soportar el intercambio de procesos enteros así como de fracciones del mismo. Ej. Exportar la lista de Roles de un proceso.



[WFMC]

B.6 Módulo Cliente

El módulo cliente es el componente del sistema encargado de interactuar con los recursos humanos del proceso. El mismo puede ser parte del producto de Workflow, o ser desarrollado a medida, por ejemplo para integrarlo a otros sistemas de la empresa. Puede desarrollarse “desde cero”, o ser aprovechar otras aplicaciones del ambiente usuario; por ejemplo, clientes de correo electrónico.

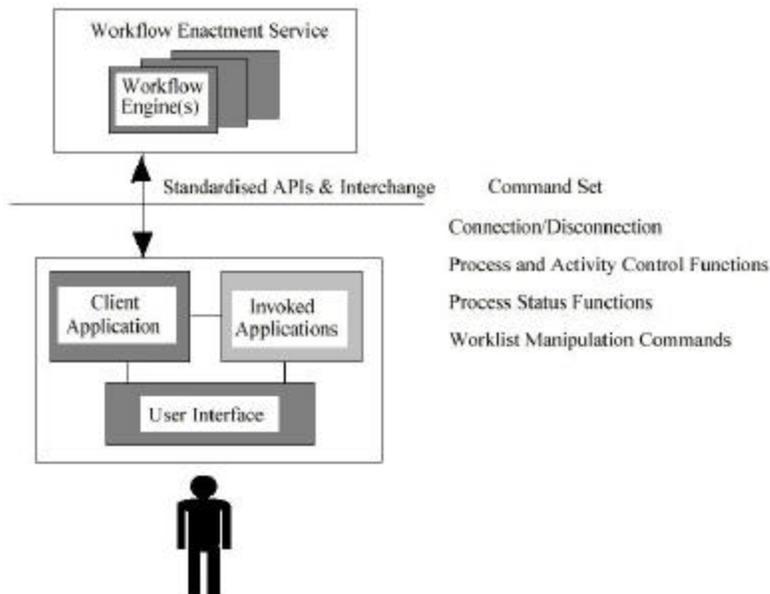


Figure 11 - Client Application Interface

[WFMC]

Dentro del modelo de referencia, la comunicación con el cliente se modela entorno al concepto de “Worklist”. La Worklist es el conjunto de tareas que tiene pendientes de ejecutar un recurso. Cuando el sistema desea

despachar una tarea a un usuario, encola la tarea en la lista asociada al mismo. Cuando el usuario se libera, inspecciona su lista de pendientes, retira la de mayor prioridad, la ejecuta y devuelve al sistema el resultado. La activación de las tareas pendientes puede estar bajo el control del programa cliente o del propio usuario.

Los modelos de implementación posibles son los siguientes:

- **Modelo basado en el Host:** Toda la aplicación cliente (incluida la worklist) reside en el ambiente del motor. Del lado cliente solo se conserva la IHM en forma de un cliente delgado (emulador de terminal, Web Browser, etc.).
- **Modelo de almacenamiento compartido:** El cliente y el servidor se comunican leyendo y escribiendo en espacio de almacenamiento compartido, donde reside la worklist.
- **Modelo de correo electrónico:** La comunicación se realiza utilizando correo electrónico. La bandeja de entrada se convierte en la worklist. Esta se almacena fuera del motor de workflow, ya sea en el servidor de correo o en el propio cliente dependiendo del sistema de correo utilizado (POP3, IMAP, etc.).
- **RPC o Mensajería:** LA comunicación es lograda mediante RPC u otro modelo de mensajería. La Worklist puede residir en cualquiera de los dos ambientes, según la implementación.

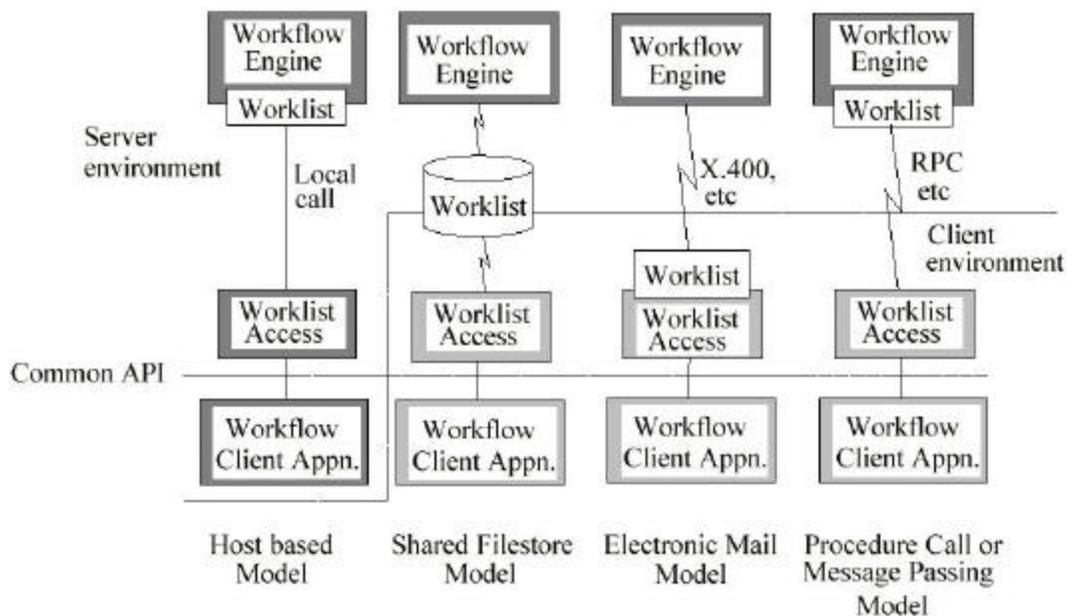


Figure 5 - Alternative client worklist handler implementations
[WFMC]

B.6.1 Interfase 2

La interfase 2 esta compuesta por un conjunto de APIs que cubren las siguientes áreas:

- Establecimiento de sesión: Login, Logout, etc.
- Consulta de procesos definidos en u motor,
- Control de Procesos: Creación / Inicio / Terminación de instancias de proceso
- Consulta del estado de una instancia de proceso:
- Manejo de la Worklist y su contenido: Acceso a las tareas pendientes, notificación de selección
- Intercambio de datos: Pasaje de datos relevantes y entrega de resultados
- Funciones de administración
- Invocación de Herramientas.

Las mismas son convertidas a cada mecanismo de comunicación particular siguiendo estándares particulares. Ej. MIME bindings para el modelo de e-mail.

B.7 Invocación de aplicaciones externas

Existen ciertos tipos de tarea, en donde el rol principal no es el del usuario, sino un recurso de IT. En actividades como imprimir un formulario, consultar una base de datos o un sistema de producción, el usuario se limita a servir de nexo entre ambos sistemas, acarreando información de un sistema a otro.

Para obtener el máximo provecho de los recursos del ambiente, un sistema de Workflow debe ser capaz de gestionar los recursos de IT de la misma forma en que gestiona los recursos humanos. Por ejemplo: Si una tarea implica recuperar información de un cliente de un sistema transaccional, podemos dejar que el sistema de Workflow realice la consulta, y le pase los datos ya seleccionados al usuario. De esta forma, los usuarios pueden concentrarse en la esencia de su trabajo y delegar al sistema los aspectos automatizables del proceso.

Para lograr que un Workflow utilice eficientemente los recursos de IT, el motor debe ser capaz de invocar aplicaciones externas. La forma de hacer esto depende tanto de la plataforma como de la propia naturaleza del recurso invocado. Por esto, no se puede esperar que un sistema sea capaz de comunicarse con cualquier recurso externo de forma nativa. Algunas estrategias de interacción posibles[WFMC]:

Tipo de Interfaz	Acceso a Datos
Llamada a proceso local	Archivo local
Script de Shell	Archivo local
CORBA / COM / etc.	Por referencia
RPC	Por referencia
Mensajería (ej. X.400)	Incrustado o por referencia
Transaccional	Incrustado o por referencia

Para resolver este problema, se puede optar por dos estrategias:

- **Restringir el tipo de recursos** que pueden ser manejados, en particular aquellas donde los datos son fuertemente tipados y pueden ser asociadas directamente con alguna aplicación (Ej. MIME)
- **Utilizar “Agentes de Aplicación”**, módulos tipo “plug-in” que contienen toda la lógica de invocación del tipo de recurso y que exportan una interfaz estándar hacia el motor.

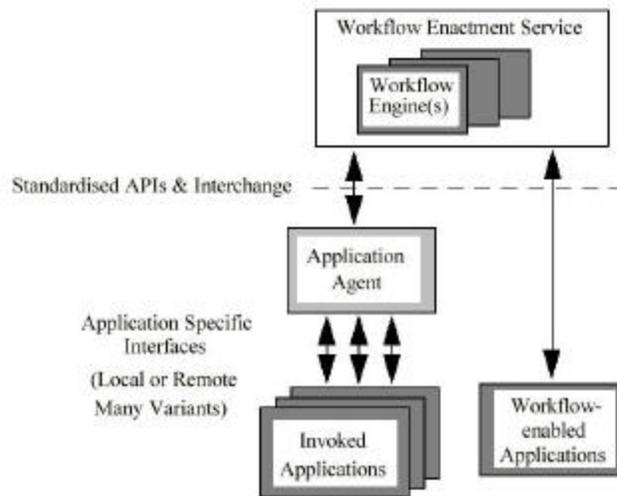


Figure 12 - Invoked Application Interface [WFMC]

B.7.1 Interfase 3

Como se ve en la figura, el motor usa la interfase 3 para interactuar tanto con los agentes de aplicación como con las aplicaciones. Las APIs que componen la interfase se dividen en los siguientes grupos:

- **Establecimiento de sesión:** Conexión / Desconexión con la aplicación o el agente correspondiente
- **Control de Actividad:** Iniciar / Suspender / Terminar / Abortar una actividad, (de la aplicación al WF) Notificar terminación, eventos, etc.
- **Pasaje de datos:** Envío de parámetros, recuperación de resultados

B.8 Interoperabilidad

El objetivo principal de las funciones de interoperabilidad es permitir que motores de Workflow desarrollados por distintos vendedores intercambien actividades para ejecutar.

Para formalizar los requerimientos de interoperabilidad, se han definido 4 escenarios posibles:

- **Conexión Discreta:** El modelo contempla un único punto de conexión entre los dos procesos. El motor A provoca la creación y ejecución de una instancia de proceso en el motor B. No hay comunicación posterior.
- **Jerárquico:** Un paso de actividad en el motor A genera la ejecución de un proceso en el motor B. El flujo de control es pasado al proceso hijo; el padre no prosigue hasta que termine el hijo. Se debe poder intercambiar parámetros y resultados mapeando las variables de un proceso en las del segundo. La jerarquía de invocaciones puede ser generalizada a mas niveles. Puede o no permitirse recursión.
- **Peer-to-Peer:** Los dos motores trabajan completamente integrados. Se comparte la definición del workflow, mientras que algunas actividades son ejecutadas por un motor y el resto por el otro. Esto último debe lograrse de forma transparente al usuario; los distintos motores deben ser vistos como un único motor. El nivel de interoperabilidad requerido es máximo.
- **Paralelos Sincronizados:** Cada motor ejecuta procesos distintos simultáneamente. Cada proceso tienen definidos puntos de sincronización con el otro. El primer motor que alcance el punto de sincronización debe esperar a que el segundo motor alcance el suyo para continuar la ejecución (Rendez-Vous). Eventualmente puede intercambiarse información durante la sincronización.

B.8.1 Interfase 4

La interfase 4 esta dividida en las siguientes funcionalidades:

- Invocación de actividades o subprocesos
- Control de progreso/estado de las actividades/subprocesos
- Transferencia de datos
- Sincronización
- Lectura/escritura de definiciones de proceso

B.8.2 Intercambio de Definiciones de Proceso

Si ambos motores pueden interpretar un formato común de definición de proceso, generadas por una única herramienta de definición por ejemplo, se puede tener una vista común de los objetos de definición y sus atributos desde los dos ambientes. Esto permite que motores heterogéneos intercambien referencias a actividades y subprocesos para su ejecución dentro de un mismo esquema de identificación.

B.8.3 Control de Ejecución

Durante la ejecución, la interfase 4 es utilizada para transferir el control de ejecución de un proceso al motor encargado de ejecutar la siguiente actividad del proceso. Si ambos motores soportan al mismo nivel la API, la interacción se podrá dar de forma directa entre motores (previo acuerdo de un protocolo común de implementación de las primitivas de la API).

En caso de que no pueda darse la comunicación directa entre motores, la API puede utilizarse para construir un módulo de Gateway que mapee los objetos y datos de la vista de un motor en los objetos de la vista del otro. Ya que el Gateway interactúa con ambos motores, debe soportar los protocolos de comunicación de ambos.

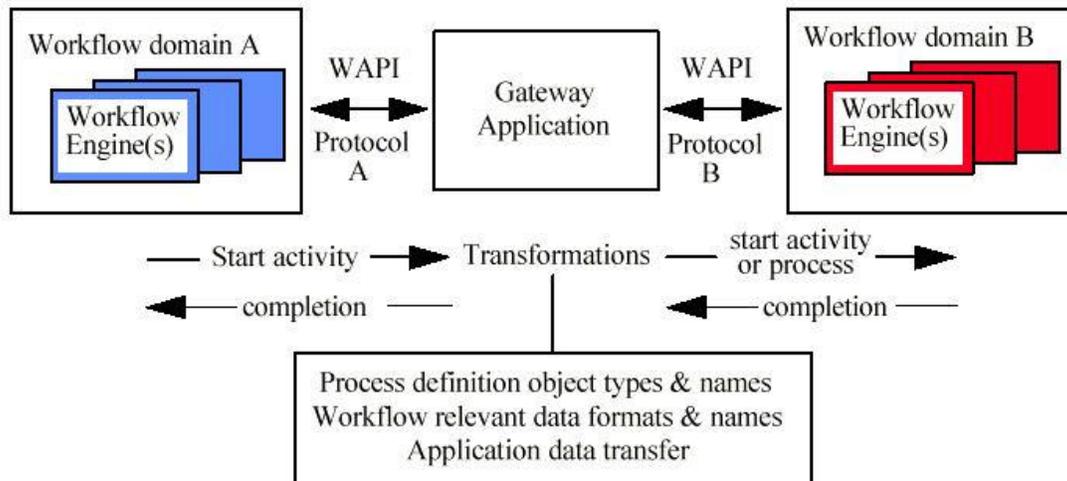


Fig 18 - Gateway operation using WAPI

B.9 Administración y Monitoreo

El objetivo de definir un conjunto estándar de funciones de administración y monitoreo es:

- Permitir que la herramienta de administración de un proveedor controle el motor de otro
- Permitir que una herramienta controle varios motores heterogéneos.

Aunque muchas de las funciones de administración ya están incluidas en el resto de las interfases (ej. Control del estado de una actividad), estas tienen visión sobre un fragmento del estado del motor (solo una actividad, o un proceso, o un usuario) y no describen el estado general del sistema. Las funciones provistas por esta interfase tienen en cambio propósito administrativo, por lo que incluyen además aspectos como seguridad, control y autorización.

B.9.1 Interfase 5

El modelo presenta una herramienta de administración independiente operando diferentes sistemas.

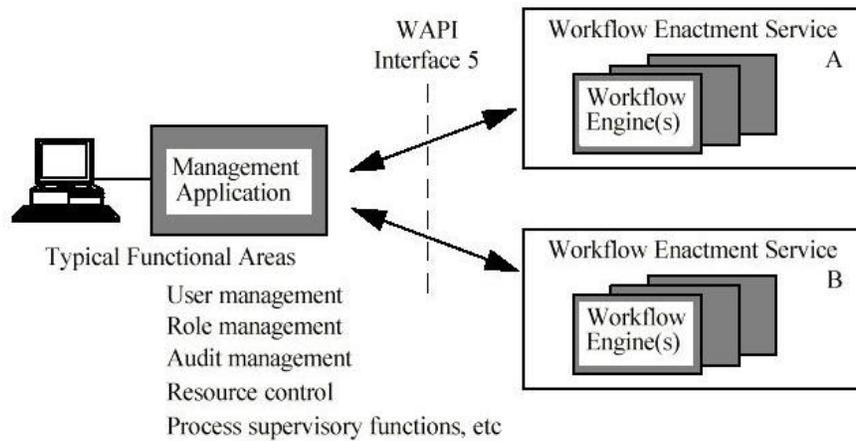


Fig 19 - Systems Administration & Monitoring Interface

Además de las áreas de administración mencionadas, la herramienta de administración puede manejar la definición de procesos, funcionando en casos como repositorio y distribuyendo las definiciones a los dominios que las requieran vía Interfase 1.

Apéndice C - BizTalk Server

C.1 Introducción

La interacción de sistemas en el contexto de Business-to-Business presenta grandes desafíos para arquitectos y desarrolladores. Las organizaciones utilizan una gran variedad de sistemas para soportar su gestión, y generan información en gran variedad de formatos. Por esto, lograr un nivel de interoperabilidad adecuado a los requerimientos de B2B resulta un problema para nada trivial.

Microsoft BizTalk Server 2000 ofrece una solución completa al problema de integrar aplicaciones y datos, permitiendo a las empresas integrar y automatizar sus procesos de negocio.

C.2 Servidores .NET Enterprise

BizTalk Server es parte de la plataforma de servidores empresariales .NET de Microsoft, concebida para proveer soluciones de e-Business. La misma fue diseñada con el objetivo de ayudar a las organizaciones a integrar sus servicios y aplicaciones en una única solución consistente, y se basa en estándares de WEB libres (Ej. XML) para fomentar la interoperabilidad.

La .NET Enterprise Server Platform esta compuesta por:

- **Application Center 2000:** Herramienta de administración y deployment de aplicaciones WEB.
- **BizTalk Server 2000:** Orquestación y mensajería. Resuelve la coordinación de procesos de larga duración y bajo acoplamiento.
- **Commerce Server 2000:** Sistema para construir soluciones Business-to-Consumer (B2C) personalizados.
- **Exchange 2000 Server :** Servidor de mensajería y colaboración.
- **Host Integration Server 2000:** Plataforma de integración que permite adaptarse a nuevas oportunidades de negocio manteniendo la infraestructura existente.
- **InternetSecurity and Acceleration 2000:** El ISA Server provee conectividad con Internet de forma segura y fácil de manejar, mientras que acelera el despacho de contenido Web mediante caché.
- **Mobile Information Server 2000:** Servidor de información para usuarios móviles
- **SQL Server 2000:** Sistema de Base de Datos para soluciones Web.

C.3 BizTalk Server, ¿para que?

Muchas organizaciones utilizan redes de intercambio de datos (EDI, de Electronic Data Interchange) para soportar su gestión (Ej. Red bancaria SWIFT). Las mismas son de gran tamaño, complejas y caras de establecer. Normalmente se basan en múltiples formatos de documento tales como ANSI-X12 o EDIFACT.

BizTalk Server provee una plataforma confiable para el intercambio de documentos y la integración de procesos de negocio. El mismo soporta una gran cantidad de formatos de documento y protocolos, lo que permite a las organizaciones implementar nuevas tecnologías reutilizando el software existente.

BizTalk Server ofrece un entorno de ejecución y desarrollo basado en la Web para integrar procesos levemente acoplados¹⁸ y de larga duración¹⁹ entre y dentro de las organizaciones.

¹⁸ Proceso débilmente acoplado (lightly coupled): Las tareas que lo componen están débilmente acopladas.

¹⁹ Proceso de larga duración (long term): Procesos que pueden durar de días a años, por lo que no pueden ser ejecutados en memoria (como un programa común). El estado del mismo debe ser persistente. Se implementan utilizando arquitecturas de tipo motor / intérprete sobre definición y datos persistentes.

C.3.1 Problemas al integrar aplicaciones y datos

La industria no ha ofrecido soluciones al problema de integrar aplicaciones levemente acopladas de larga duración, que son comunes en muchas industrias(Ej.: Banca). Como resultado, integrar este tipo de aplicaciones resulta difícil para organizaciones con una gran base de clientes y proveedores.

Por otro lado, el uso de diversos formatos de EDI, protocolos y aplicaciones complican la interoperabilidad dentro y a través de las organizaciones.

Los siguientes factores causan problemas al integrar aplicaciones y datos:

- **Aplicaciones:** Se construyen utilizando diversos lenguajes metodologías de programación.
- **Datos:** Son representados en varios formatos de EDI, lenguajes de programación y modelos, y la plataforma de EDI no es consistente al cambiar de industria.

C.3.2 Integración de Aplicaciones Empresariales

Las organizaciones han utilizado redes de EDI para intercambiar y procesar información de negocio electrónicamente. Sin embargo, las mismas exigen rediseñar las aplicaciones existentes para implementar nuevos procesos de negocio.

BizTalk Server permite ofrecer soluciones de Integración de Aplicaciones Empresariales(EAI) sobre redes de B2B. Las mismas ofrecen los siguientes beneficios:

- Permiten reutilizar aplicaciones existentes e integrarlas a nuevas
- Logran mejorar la productividad mediante un manejo mas eficiente de información y recursos.
- Ahorro en costo al utilizar recursos compartidos.
- Retención de clientes al mejorar el servicio.

C.3.3 BizTalk Framework

La BizTalk Framework es un conjunto de especificaciones para el intercambio de documentos de negocio, construida sobre el protocolo SOAP. La misma provee un conjunto de guías para la elaboración de esquemas XML de forma consistente e interoperable, basadas en los estándares de la W3C: HTTP, MIME y SOAP.

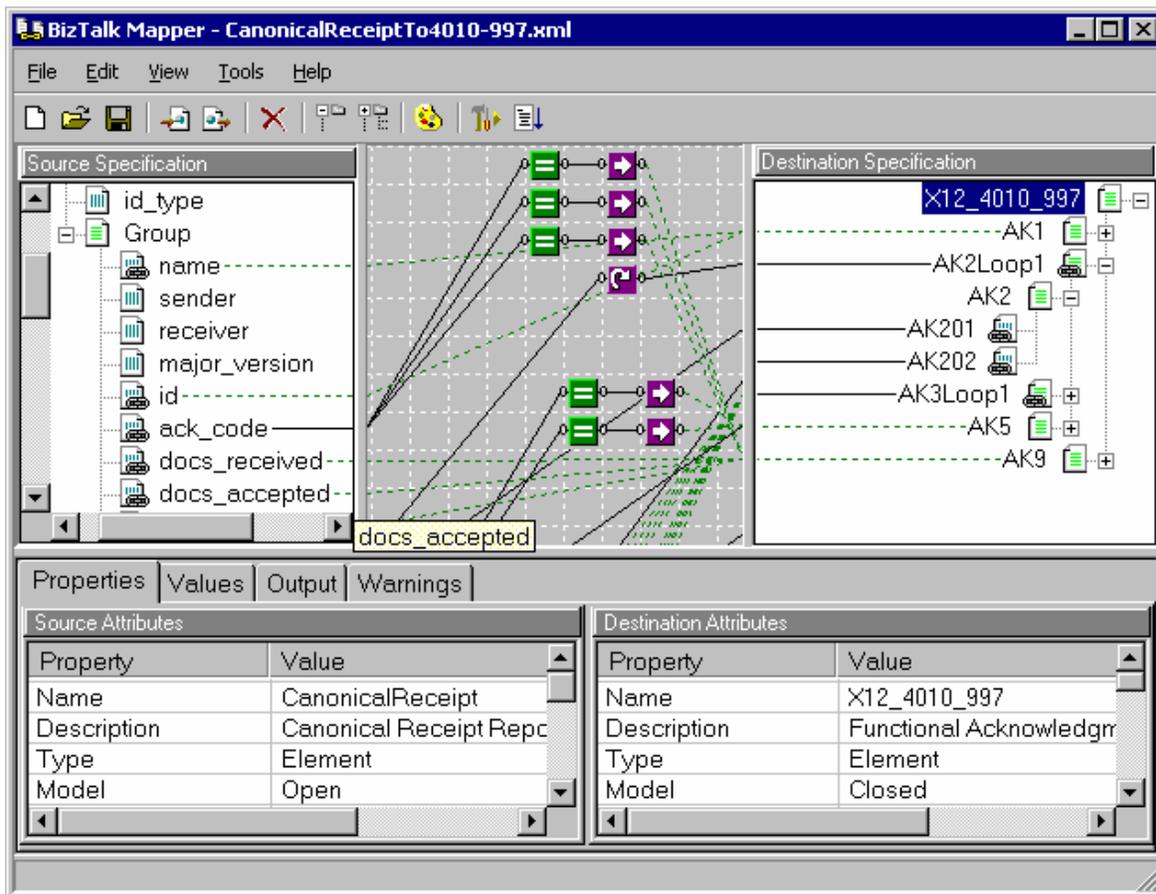
La BizTalk Framework especifica:

- **Documento BizTalk:** Un documento BizTalk es un mensaje SOAP 1.1. El body contiene uno o mas documentos de negocio; el Header contiene entradas específicas de BizTalk.
- **BizTags:** Son un conjunto de tags XML utilizados para construir el header SOAP, y permiten especificar el manejo del mensaje.
- **Mensaje BizTalk:** El mensaje BizTalk se utiliza para transmitir documentos BizTalk. Su estructura depende del protocolo de transporte²⁰ utilizado.

²⁰ En el contexto de intercambio de documentos de negocio, el termino “protocolo de transporte” refiere a los protocolos utilizados para el transporte de documentos. Los mismos son protocolos de aplicación del modelo OSI/ISO (HTTP, SMTP; FTP, etc).

C.4.2 BizTalk Mapper

Siguiendo el estilo de BizTalk Editor, BizTalk Mapper nos permite definir visualmente, transformaciones (mappings) de un esquema de datos en otro. Estos mappings serán utilizados por el motor de mensajería para traducir documentos de un esquema a otro cuando se deba transmitir información entre sistemas no consistentes.



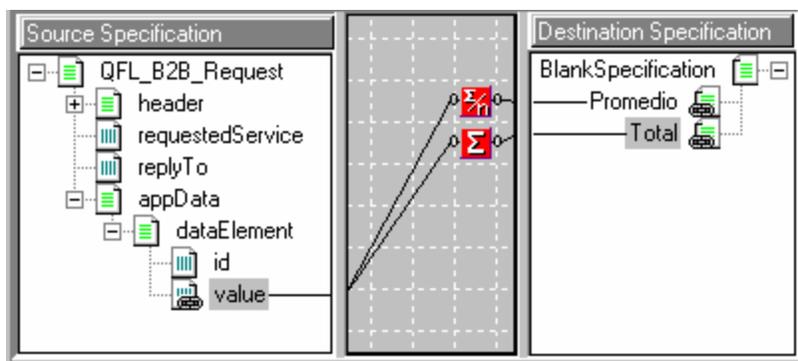
BizTalk Mapper utiliza una representación gráfica del mapeo, el cual puede incluir conversiones de tipo "copia del valor", representadas por líneas, y conversiones estructurales más complejas, representadas por **functoides**.

La cardinalidad de los functoides es variable. Algunos generan un resultado para cada valor de origen que se encuentre en el documento de origen, mientras que otros generan un único valor para todas las instancias de un dato.

Existen 9 categorías de functoides:

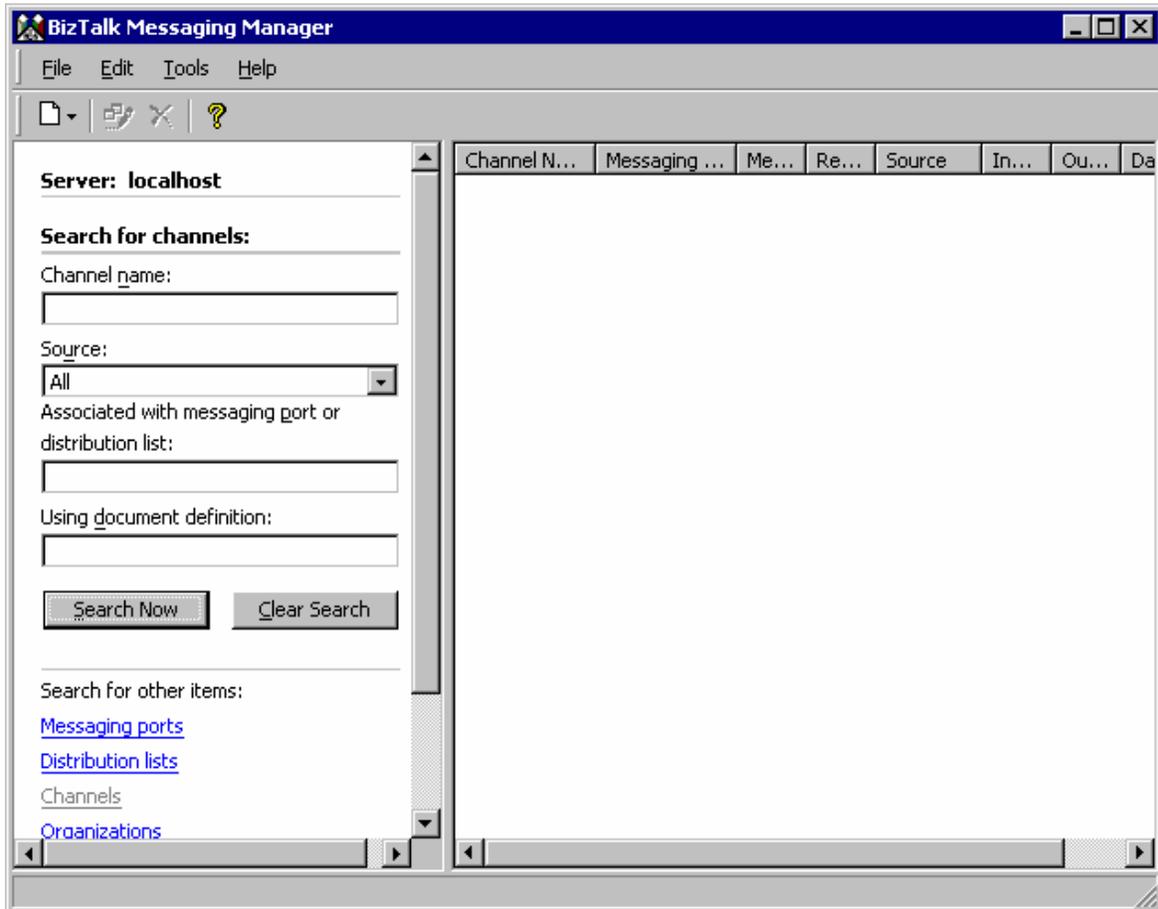
- Manejo de Strings: Manipulación de strings. Por Ej. Buscar un string dentro de otro, convertir a mayúsculas, “trimming”, etc.
- Funciones matemáticas: Valor absoluto, máximo de los parámetros, raíz cuadrada, etc
- Operadores Lógicos: Mayor, menor, esNumero?, etc.
- Fecha / tiempo: Fecha de hoy, hora de hoy, sumar días a una fecha.
- Funciones de Conversión: Conversiones de ASCII a char, y entre bases(octal, decimal, hexadecimal)
- Científicos: Trigonometría, exponenciación y logaritmos.
- Acumulativos: Sumatoria, Promedio, Mínimo, Máximo, etc.
- Acceso a Bases de Datos: Recuperar un registro, un campo, etc. Accede vía ADO.
- Avanzados: Paso de Script, Functoides para iterar sobre instancias, etc.

Los functoides acumulativos operan sobre el conjunto de instancias de un tag. En el ejemplo, se calcula el promedio y la suma de todos los campo “value” de todos los “dataElement” hijos de “appData”



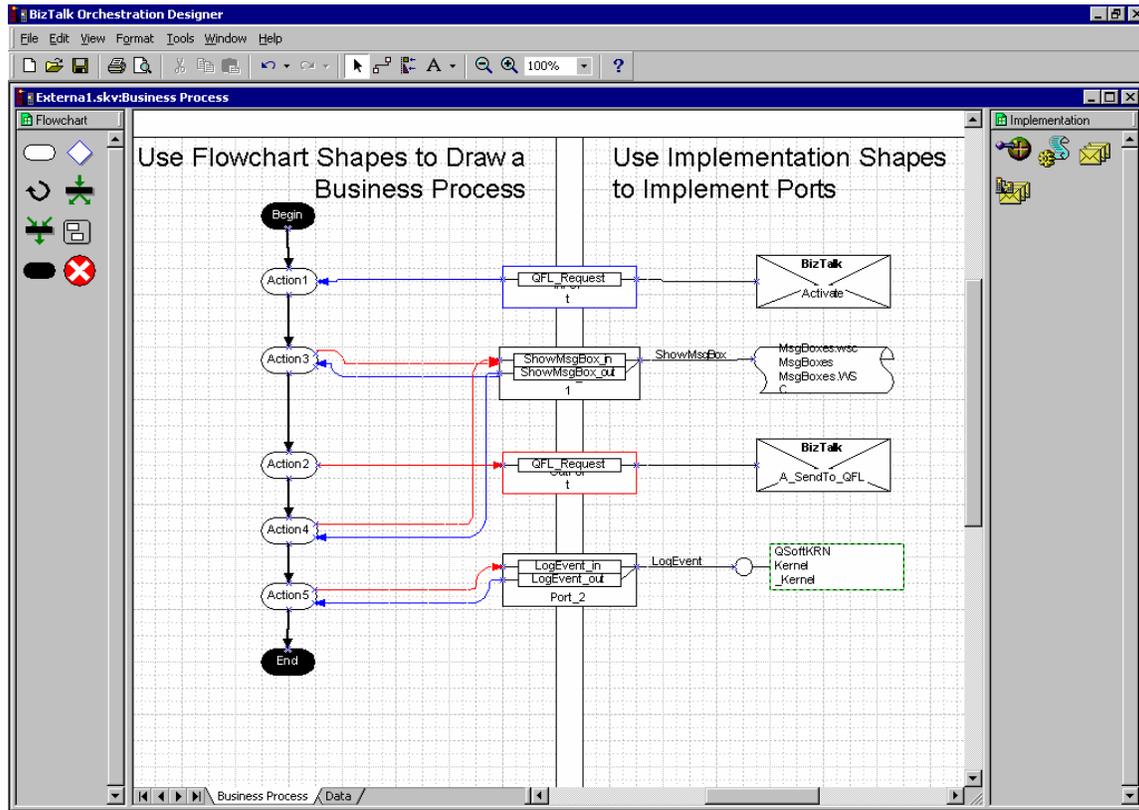
C.4.3 BizTalk Messaging Manager

El administrador de mensajería permite administrar el intercambio de documentos vía BizTalk. Ofrece un conjunto de objetos de configuración (Canales, Puertos, Organizaciones, Aplicaciones, Definiciones de Documentos, Envelopes y Listas de Distribución) que definen el ruteo y procesamiento que se le dará a los documentos que lleguen al motor. La (bastante pobre) interfaz se basa en visualizar listas de estos objetos y editarlos mediante un “wizard”.



C.4.4 BizTalk Orchestration Designer

El BizTalk Orchestration Designer es el diseñador de schedules. Basado en VISIO 2000, permite modelar procesos de negocio como un diagrama de flujo y definir los puntos donde las aplicaciones involucradas intercambiarán documentos. Permite visualizar de forma separada el flujo de ejecución y el flujo de datos del proceso.



C.4.5 BizTalk Server Administration

El Administrador de BizTalk Server ofrece una interfase de usuario única para el mantenimiento de los servidores BizTalk en uso. El mismo permite:

- Crear y administrar grupos de servidores (para clustering)
- Configurar propiedades globales y particulares de cada grupo/servidor.
- Crear funciones de recepción (receive functions)

El administrador también permite acceder a los documentos que están en procesamiento. BizTalk ordena los documentos en cuatro colas según el estado de procesamiento de los mismos:

- Work Queue: Documentos que están siendo procesados
- Scheduled Queue: Documentos que no pueden enviarse inmediatamente por cuestiones de horario
- Retry Queue: Documentos con errores recuperables; típicamente reintentos de comunicación
- Suspended Queue: Documentos con errores o fallas irreversibles de procesamiento. El administrador puede ver, borrar o reintentar procesar los mismos.

C.5 Servicios de Mensajería de BizTalk

El servicio de mensajería de BizTalk se ocupa de enviar y recibir documentos, “parsearlos”, interpretar la información de ruteo contenida en ellos y asegurar su entrega. Sus tareas incluyen el mapeo de esquemas, generación y correlación de confirmaciones (receipts), control de integridad y seguridad.

C.5.1 Receive Functions

La forma básica de despachar documentos hacia BizTalk es mediante COM (utilizando la interfaz IInterchange). Para las aplicaciones que no puedan utilizar COM, BizTalk ofrece Receive Functions. Las mismas oficián de “traductores” entre la comunicación que soporta la aplicación y COM, la soportada por el motor. La versión actual de BizTalk ofrece las siguientes funciones²¹:

- File Receive: “Pollea” periódicamente una carpeta del sistema de archivos. Cuando una aplicación escribe en ella, borra el archivo y despacha el documento contenido a BizTalk.
- Message Queue: De forma semejante, es activada cuando una aplicación envía un documento por una cola de mensajes

C.5.2 Servicios de Transporte

BizTalk es capaz de transmitir documentos utilizando: HTTP, HTTPS, SMTP, Archivos, Message Queue, o Application Integration Components. Los últimos son componentes desarrollados a medida, que permiten enviar documentos hacia sistemas complejos como SAP.

C.5.3 Parsers y Serializers

BizTalk 2000 incluye parsers y serializers para varios estándares, como X12, EDIFACT y XML. Los mismos son capaces de recuperar la información de ruteo contenida tanto en el documento como en los headers que reconoce. También se pueden desarrollar parsers particulares para requerimientos específicos.

C.5.4 Entrega confiable de documentos y Seguridad

Los servicios de transporte pueden configurarse para asegurar la entrega de los documentos. Se puede especificar la hora en que se desea transmitir un documento, el número de reintentos y el intervalo de tiempo entre ellos. BizTalk 2000 incorpora encriptación de clave pública y firma digital de documentos salientes y entrantes.

C.6 Servicios de Orquestación de BizTalk

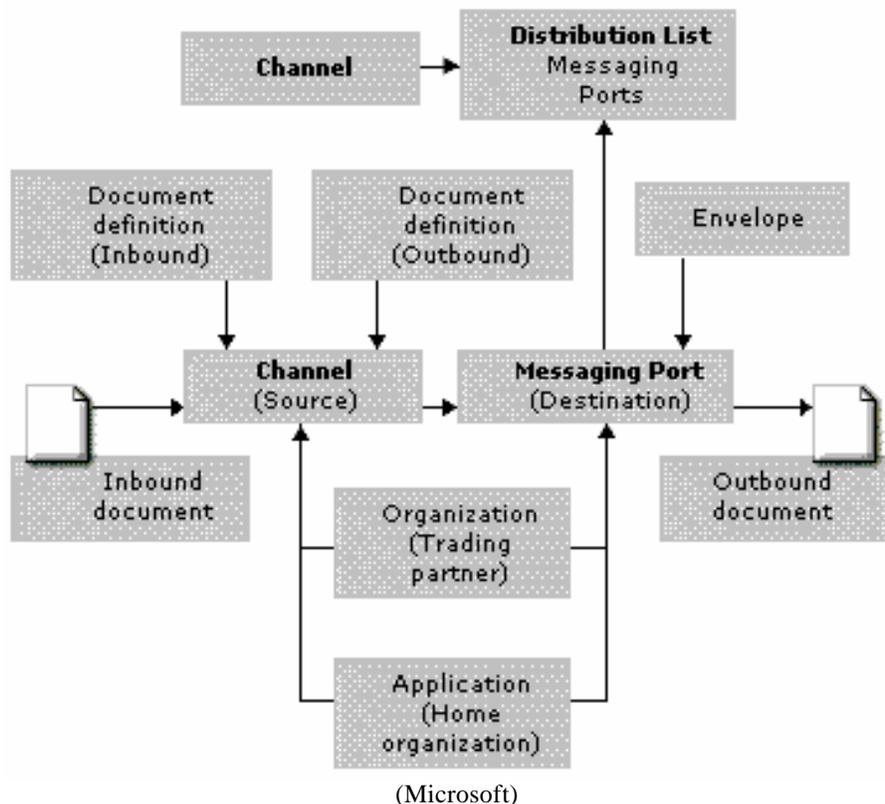
Los servicios de orquestación de BizTalk guían los procesos de una empresa, secuenciando las tareas de las aplicaciones que soportan el negocio. Los mismos permiten manejar complejos procesos que duran entre días y meses. Los procesos de la empresa se modelan utilizando rutinas escritas en XLANG (o XLANG Schedules). Los XLANG Schedules pueden verse como diagramas de flujo, donde los pasos de tarea implican interacción con aplicaciones del ambiente. La interacción con las aplicaciones es hecha vía:

- Mensajería de BizTalk
- Componentes COM
- Message Queue
- Windows Script Component

²¹ La recepción de documentos desde HTTP y SMTP no es hecha por BizTalk directamente. Deben instalarse manualmente scripts en los servidores IIS y Exchange que procesen las solicitudes entrantes. Los mismos son, notoriamente Receive Functions: Obtienen un documento y lo despachan vía COM. Sin embargo, la documentación del producto no hace referencia a esto.

C.7 Servicios de Mensajería: Modelo de Objetos

El servicio de mensajería se configura creando un conjunto de “objetos de configuración”. Los mismos definen la forma en que son procesados los documentos que pasan por el servidor. El esquema describe la relación entre los mismos:



C.7.1 Canales

Los canales representan las reglas usadas para procesar los documentos que llegan al servidor. Cuando estos llegan a BizTalk, son procesados por el canal que coincide con el origen, destino y formato del documento. Luego, el documento es dirigido al puerto asociado al canal (o lista de distribución) para su transporte.

Cada canal específica:

- Origen: Puede ser una aplicación o una organización.
- Documentos de origen y destino
- Mapeo utilizado para convertir el documento entrante en el saliente. Es obligatorio si los documentos son distintos.
- Información de registro y seguimiento

C.7.2 Puertos

Un puerto de mensajería define la forma en que son transportados los documentos a su destino. El mismo podrá ser una organización externa o una de nuestras aplicaciones. Existen dos tipos de puertos: Hacia una organización y hacia una aplicación, dependiendo del tipo de destino. Un puerto puede ser compartido por varios canales. Cada puerto específica:

- Destino: Una organización, una aplicación, o un schedule XLANG.
- Transporte: HTTP, HTTPS, SMTP, Loopback o AIC (Application Integration Components, desarrollados “a medida” del sistema de destino)
- Envelope: El envelope que se utilizará para encapsular los documentos antes de transportarlos.

- Seguridad: Se podrán especificar certificados de encriptación para utilizar con HTTPS y las extensiones s/MIME

C.7.3 Aplicaciones y Organizaciones

Las Organizaciones representan partners con las cuales existen relaciones de EDI. Existen dos tipos de organizaciones:

- Home Organization: Representa nuestra propia organización. Se crea durante la instalación y no puede ser eliminada, pero puede renombrarse. Puede tener asociadas Aplicaciones, las cuales representan los sistemas que dan soporte a nuestro negocio. La Home Org no puede ser origen o destino de documentos, pero si las aplicaciones asociadas a ella.
- Partner Organization: Representa las organizaciones con las que se mantienen vínculos de EDI. No pueden tener aplicaciones asociadas²².

Los intercambios posibles resultan:

- Org – Org : Nuestra organización actúa como intermediaria del documento.
- Org – App: Un partner nos envía un documento, y es procesado por uno de nuestros sistemas
- App – Org: Uno de nuestros sistemas envía un documento a una organización externa.
- App – App: Dos aplicaciones internas colaboran vía BizTalk.

Las organizaciones no están asociadas a protocolos de transporte ni direcciones. Las mismas pueden recibir documentos en varias direcciones y múltiples protocolos, dependiendo de los puertos que se utilicen.

Cada organización tiene asociada un conjunto de identificadores, los cuales son utilizados para asegurar que los documentos son entregados al destinatario correcto. Cada identificador consiste de un nombre, un calificador y un valor. Para compararlos solo se utiliza el calificador y el valor.

Ejemplos son:

Nombre	Calificador	Valor
PorRUC	RUC	123456789123
PorURL	URL	www.urudata.com

Todas las organizaciones contienen al ser creadas el identificador, el cual no puede ser editado.

“Organization”	“OrganizationName”	Nombre_de_la_Org
----------------	--------------------	------------------

C.7.4 Definiciones de Documentos

Una Definición de Documento representa un tipo de documento que será procesado por BizTalk. La misma contiene:

- Una referencia a una especificación de documento creada en BizTalk Editor. El archivo de especificación debe estar almacenado en un espacio WebDAV²³ para que sea accesible desde la Web.
- Parámetros de registro y seguimiento del tipo de documento.

C.7.5 Envelopes

Un envelope consiste de un header y un footer que se agregan al documento antes de ser enviado. El header contiene información útil para el ruteo y procesamiento del documento en el servidor de destino. En particular, el envelope “Reliable”, incluido en BizTalk 2000 implementa la especificación de la BizTalk Framework.

²² Como son externas, no tenemos control sobre como procesan los documentos que les enviamos

²³ WebDAV es una extensión de HTTP que permite editar y administrar archivos en un servidor Web.

C.7.6 Listas de Distribución

Las listas de distribución son conjuntos de puertos que pueden utilizarse para enviar el mismo documento a un conjunto de destinatarios. Toda lista debe contener por lo menos un puerto.

C.7.7 Puertos y Canales Abiertos

Se dice que un puerto o un canal es “abierto” si no contiene información explícita acerca del origen o destino de los documentos. En este caso, la información faltante es obtenida del header BizTalk del documento (del Envelope) o de los parámetros de la función Submit utilizada para pasar el documento a BizTalk.. Un puerto a una aplicación no puede ser “abierto”.

Si un canal es definido como abierto (“Open Source”), aceptara cualquier documento que coincida con la especificación y con el destino del puerto asociados.

Un puerto abierto puede utilizarse para enviar documentos a múltiples destinatarios sin crear un puerto particular para cada uno. El envelope, la seguridad y el transporte es igual para todos.

D Apéndice 4 – Q-flow

Q-flow es un conjunto de herramientas desarrollado por Urudata, que permite modelar procesos de negocio en forma electrónica. El paquete esta compuesto por:

- El motor de Workflow
- El sitio Web de la aplicación
- Herramientas de diseño de procesos y formularios
- Herramientas de administración, configuración y mantenimiento
- Herramientas de estadística e integración con sistemas de información gerencial (OLAP).

D.1 Modelo Conceptual

Q-flow maneja básicamente cuatro elementos:

- **Usuarios:** Representan a los Recursos Humanos que participan en los procesos.
- **Roles:** Representa a las funciones y los cargos definidos por la organización. Permiten referirse indirectamente a los usuarios.
- **Templates:** Son la definición electrónica de un proceso; el diseño de un proceso.
- **Workflows (o simplemente Flows):** Son instancias de proceso, en ejecución o terminados.

D.1.1 Usuarios

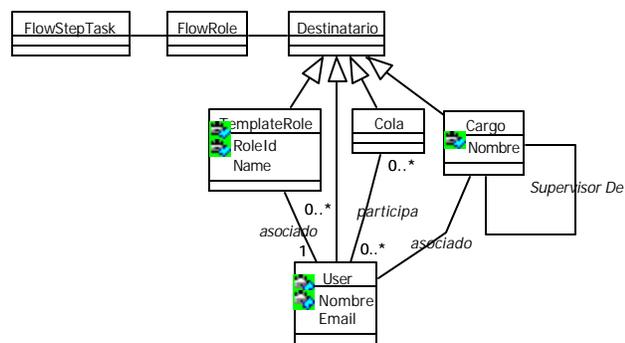
Cada **Usuario** es representado en el sistema por una cuenta (un “**Login**”) en alguno de los proveedores de seguridad configurados, y una **cuenta de correo**. De esta forma, el sistema es capaz de identificar a un funcionario tanto cuando este accede al sistema (Login) como cuando el sistema “accede” al usuario (vía mail).

D.1.2 Roles

Los **Roles** son referencias indirectas a los **Usuarios**. Los mismos permiten que el diseño de los procesos sea independiente de los cambios que puedan darse en el organigrama de la organización, abstrayendo la estructura organizacional de los usuarios que la desempeñan.

Existen 6 tipos de roles:

- **Roles de la Organización:** Representan a los cargos de una organización. Los mismos están asociados a un único usuario. Ejemplo: “Gerente financiero”
- **Colas de Trabajo:** Representa funciones de la organización que son desempeñadas por varios usuarios indistintamente. Ejemplo: Cajero. El trabajo despachado a una cola puede ser asignado a los usuarios por un moderador, o por los propios participantes, quienes pueden autoasignarse las tareas.
- **Roles del Proceso (o Template):** Representan funciones cuya definición tiene sentido solo dentro del contexto del proceso. Ejemplo: “Autorizador”, “Solicitante”. Los mismos pueden ser asociados a un usuario o rol durante el diseño o al momento de iniciar el proceso.
- **Roles de la Instalación:** Representan las funciones asociadas al propio sistema. Ej: “Administrador del sistema”
- **Roles de la Instancia:** Son roles definidos por el sistema, cuya asignación es relativa al template o flow en ejecución: Ejemplo: “Propietario del flow”, que representa al iniciador del proceso, quien sea que halla sido.



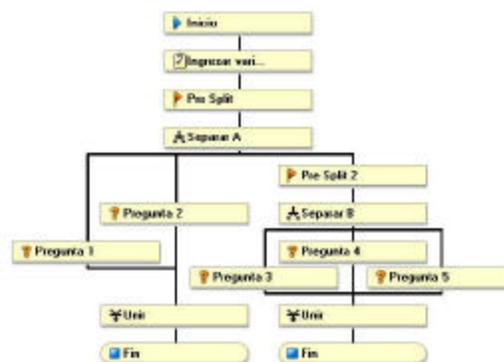
- **Roles Relativos:** El rol “Manager del Usuario”. permite hacer referencia al superior del usuario en la estructura de la organización.

D.1.3 Templates

Los Templates están compuestos por:

- Un **Grafo de Pasos**
- Un conjunto de **Datos de Aplicación**, y
- Un conjunto de **Roles de Template**.

Un **Paso** representa una unidad de ejecución del workflow, y las aristas indican el orden en que deben ser ejecutados. Existen 17 clases predefinidas de paso: Start, End, Ask, Task, Milestone, Notify, Eval, Script, StartFlow, Delegate, Fork y Join.



La cantidad de aristas entrantes y salientes depende de la clase de cada paso. Cada tipo de paso determina que aristas espera para iniciar y que aristas despacha al finalizar, por lo que, potencialmente, cualquier paso escapaz de controlar el flujo de ejecución.

Los pasos pueden tener asociadas **acciones proactivas** como **Alarmas, Recordatorios y Delegaciones**. Las mismas permiten detectar prematuramente el desvío de una tarea, y tomar automáticamente una acción correctiva.

Los **Datos de Aplicación** representan la información asociada al proceso. Los mismos permiten pasar la información resultante de una tarea a las siguientes, o utilizarla para controlar la ejecución del proceso (utilizando pasos de evaluación).

Los datos de aplicación pueden ser **multivaluados**. Un dato multivaluado permite que el usuario ingrese tantos valores como quiera, manteniendo cada valor como instancias separadas.

El sistema permite definir permisos (editable, visible, oculto) para cada dato en cada paso.

Los datos de aplicación tienen asociado un **Tipo de Datos Inteligente**, el cual define el comportamiento que tendrá el dato cuando sea desplegado al usuario.

Un tipo de datos inteligente permite definir:

- **Tipo:** El tipo de datos como será almacenado el dato. Texto, Texto libre, Número, etc.
- **Control de Entrada:** TextBox, CheckBox, Combo, Dropdown, etc.
- **Validaciones:** Permite definir rutinas para el manejo de los eventos del control elegido
- **Fuentes de datos:** Permite conectar un tipo de datos a una base externa, de forma de que el dato elegido sea siempre miembro de la misma. Ej: Un tipo de datos “País” que tome las opciones de una tabla.

D.1.4 Flows

Los **Flows** representan a las instancias de un Template. Cada vez que se inicia la ejecución de un proceso, el sistema instancia un Flow, el cual almacena toda la información de “tiempo de ejecución” del proceso.

Cada Flow contiene²⁴:

- Un conjunto de **Pasos**
- Un conjunto de Hilos de Ejecución o “**Threads**”
- Un conjunto de **Datos**
- Un conjunto de **Archivos Adjuntos**
- Un conjunto de asignaciones (**Rol, Usuario**)

²⁴ Los componentes del flow y sus correspondientes en el Template tienen la misma denominación. Ej.: “Pasos” del Template y “Pasos” del Flow. Podrá referirse al correspondiente en diseño o en ejecución diciendo “el Template de XX”, o “la Instancia de YY”. Cuando no se haga, quedará claro por el contexto a que concepto estamos refiriendo.

Los **Pasos** del Flow almacenan la información generada durante la ejecución de cada Paso del Template. Ej: Respuestas de los usuarios, valores de las condiciones de decisión, fecha y hora de inicio y fin de la tarea, etc. Ya que un flow puede tener bucles, un Template de Paso puede tener varias instancias asociadas.

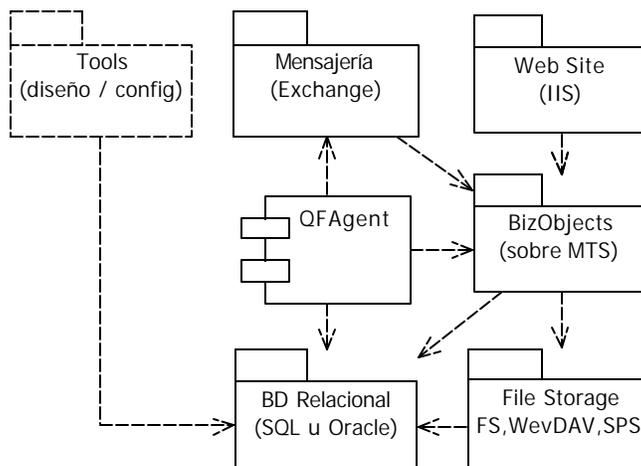
Los **Threads** representan el control de ejecución dentro del flujo. Los mismos indican que pasos del flow están activos (en ejecución). Al iniciar, cada flujo tiene un único thread asociado, y este indica como “Activo” al paso inicial del Flow. Durante el procesamiento, el motor recorre los threads activos y verifica que el paso asociado este en ejecución. Si el paso asociado a un thread fue terminado, el motor interpreta el Template del proceso en base a la información del Flow, instanciando el paso consecutivo al recién finalizado y haciendo que el thread apunte al mismo.

Cuando el paso siguiente al recién finalizado es de tipo “Fork”, el sistema genera otros tantos threads en estado activo y les asocia los pasos consecutivos al Fork. Al contrario, cuando un Thread llega a un paso de “Join”, el sistema “desactiva” el thread, y solo activa un nuevo thread cuando se cumple la condición configurada (“Algún thread alcanza el paso”, “Todos los Threads alcanzan el paso”, etc.)

Los **Datos** del Flow son las instancias de los Datos de Aplicación del Template. Los mismos almacenan el valor que toman las variables del Template en tiempo de ejecución.

Los **Archivos Adjuntos** permiten almacenar e intercambiar entre los usuarios grandes volúmenes de información no estructurada. Ej: Documentos de texto, planillas de cálculo, digitalizaciones de documentación legal, etc. El sistema provee de control de concurrencia y versionado mediante una estrategia de “Check-In / Check-Out” de los archivos.

D.2 Arquitectura



Q-flow se basa en una arquitectura en tres capas.

- **Front End**: Se compone de un acceso Web sobre IIS y Exchange como servicio de mensajería.
- **Middle Tier** dividida en dos partes:
 - **Agente Qflow**: El agente Qflow funciona como motor / intérprete, sobre los datos(flows) y metadatos(templates, etc) almacenados en la base relacional.
 - **BizObjects**: Implementan los objetos de la aplicación. Incluye los componentes QSoftXXX
- **Data Tier**: Base relacional y File Storage. El file storage es utilizado para el almacenamiento de los archivos adjuntos. La implementación contempla almacenamiento sobre el sistema de archivos local, Sharepoint Portal Server, o cualquier almacenamiento de documentos que soporte WebDAV²⁵

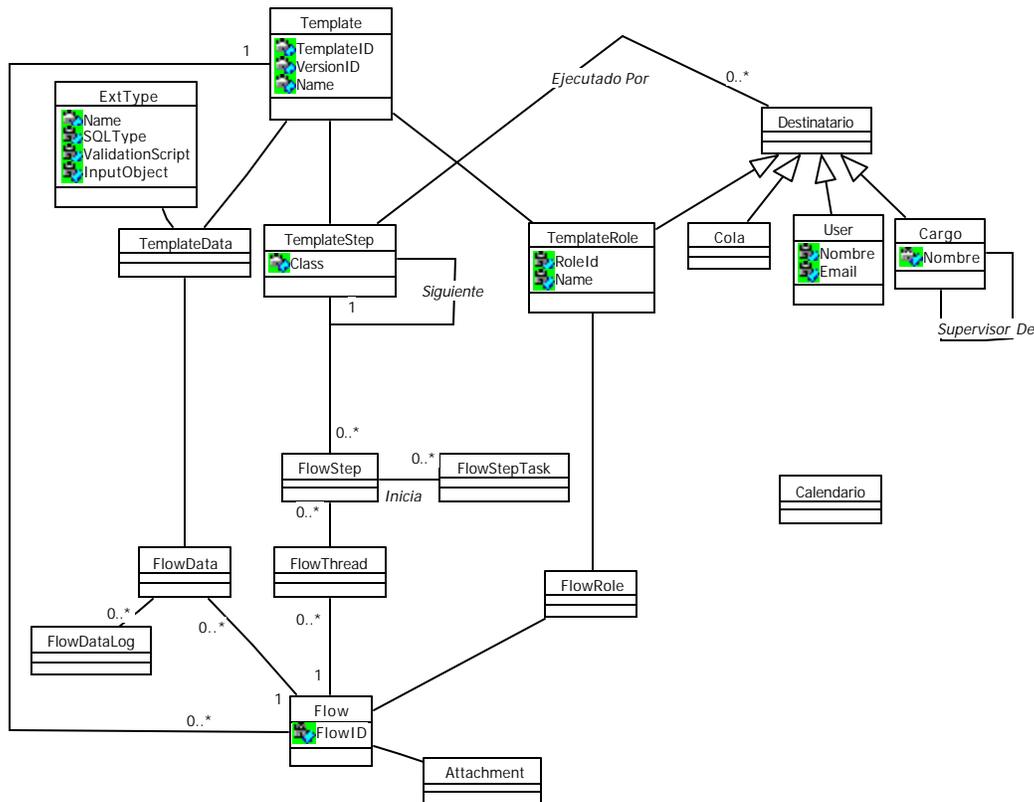
A diferencia de otras soluciones, toda la información del estado de ejecución de los flows es mantenida sobre la base relacional y no sobre el sistema de correo. Esto nos permite reconstruir los mails en caso de problemas con el sistema de correo.

La información no estructurada es almacenada en repositorios de documentos sobre el mismo sistema de archivos, web folders, Sharepoint Portal Server o bases documentales con acceso WebDAV.

La arquitectura de Q-flow es escalable, vertical y horizontalmente. La carga del sistema puede ser distribuida especializando servicios (Mail en un equipo, Motor(es) en otro(s), BD en otro, etc) , y/o implementando una granja de servidores, para distribuir los accesos.

²⁵ WebDAV: “Web Distributed Authoring and Versioning”, es una extensión del protocolo HTTP 1.1 que habilita la publicación y edición de contenido en servidores Web.

D.2.1 Estructura Estática



- **Template:** Representan una definición de proceso.
- **TemplateStep:** Representa un paso de ejecución perteneciente a un template. La relación “Siguiete” representa las aristas del diagrama de flujo.
- **TemplateData:** Representan cada uno de los Datos de Aplicación (las Variables) definidas en el Template. Define como será desplegada la variable en los formularios y vistas (Etiqueta, Encabezado de columna, Tooltip, etc).
- **ExtType:** Define un Tipo de Datos Extendido.
- **Flow:** Modela una instancia de proceso. Incluye información de tiempo de ejecución, como fecha de inicio y fin, estado (en espera, corriendo, en error, etc), progreso, etc.
- **FlowThread:** Representa un hilo de ejecución de un flow. Cada thread puede engendrar otros, por ejemplo, al ejecutar un paso de fork, lo que genera una relación de tipo Padre / Hijos, representada por la relación “Spawns”.
- **FlowStep:**
- **FlowStepTask:**
- **FlowRole:** Representa los roles propios del flow.
- **FlowData:** Representa cada variable del flow. Tiene un valor, y esta asociada a un TemplateData, de donde obtiene el tipo.
- **FlowDataLog:** Registra los cambios efectuados sobre una variable. Incluye fecha de modificación, el nuevo valor, y esta asociado al usuario que ejecuta el cambio.
- **Attachment:** Archivos Adjuntos del flow.

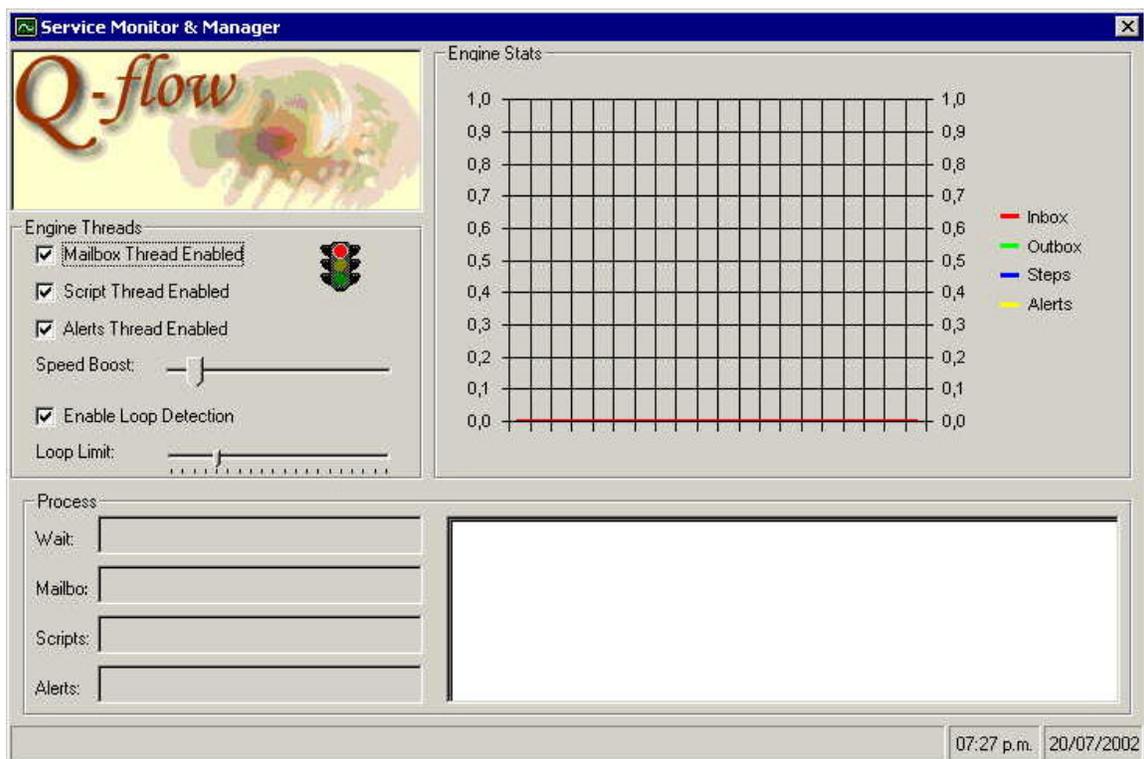
D.3 Componentes Principales.

D.3.1 Agente

El Agente de Q-flow es la parte proactiva del sistema. El mismo tiene cuatro funcionalidades:

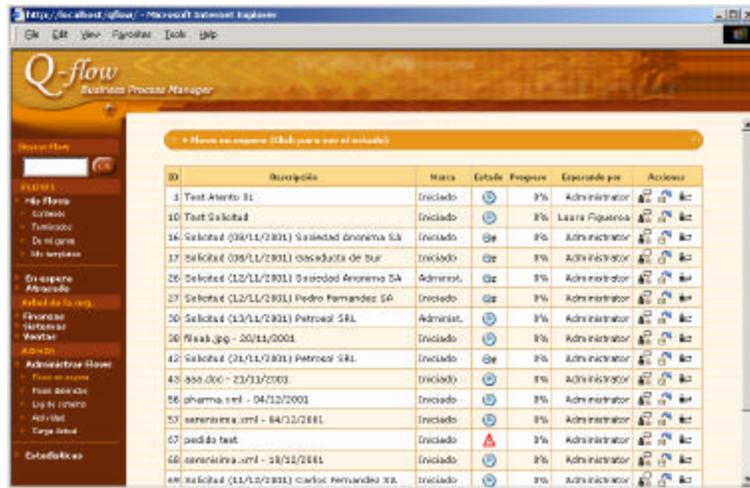
- Ejecutar los flows interpretando el template asociado: Identifica los pasos que están listos para ser ejecutados de acuerdo a secuencia indicada por el template y la evaluación de los pasos de control de flujo: Eval, Split, Join y Fin.
- Ejecutar efectivamente los pasos: Interpretar cada paso de acuerdo a su tipo y configuración:
 - Ask, Task, Notify: Interactuar con el sistema de correo para despachar las tareas a los usuarios
 - Script, BOT: Ejecutar el paso, instanciando el motor de Scripting, o despachando la tarea al agente de BOT asociado.
- Procesar las alarmas, recordatorios y delegaciones: Recorre los pasos en espera, evaluando las condiciones de tiempo configuradas y ejecutando las acciones proactivas correspondientes.
- Procesar las respuestas recibidas via mail: Recupera las respuestas recibidas por correo, y actualiza la base de datos.

El agente de Q-flow se instala como servicio, por lo que no dispone de interfase propia. A cambio, se provee una aplicación de monitoreo que se comunica con el mismo mediante sockets.



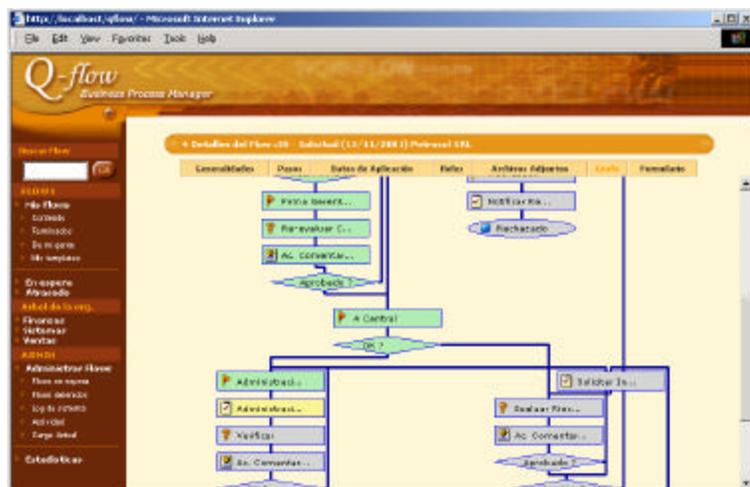
D.3.2 Sitio Web

El sitio Web de la aplicación es la interfase principal del sistema. El mismo ofrece todas las funcionalidades que necesita un usuario, y las funciones más habituales de administración.



El sitio Web de Q-flow se compone de un conjunto de “vistas” de los flows del sistema. Tal cual indica la barra de navegación a la izquierda, el mismo se organiza en tres secciones:

- **Flows:** Despliega los flows vinculados al usuario actual.
 - **Mis Flows:** Flows **iniciados** por el usuario.
 - **Mis Templates:** Dentro de “Mis Flows” se puede acceder a una lista de los templates diseñados por el usuario, desde donde se puede iniciar flows.
 - **En espera:** Flows que esperan intervención del usuario (la “bandeja de entrada”)
 - **Atrasados:** Flows que esperan intervención del usuario y que están retrasados
- **Árbol de la Organización:** Ordena los flows en carpetas según la dependencia o repartición a la que pertenecen: “Los flujos de Compras” “Los flujos de Help Desk”, etc. También permite acceder a los templates de cada área.
- **Vistas:** Permite acceder a las vistas personalizadas de los flows.
- **Administración:**
 - **Flows en Espera:** Despliega todos los flows que estén esperando una respuesta.
 - **Flows detenidos:** Despliega los flows detenidos o en estado de error.
 - **Log del sistema, Actividad y Carga Actual:** Información administrativa.



D.3.3 Interfase vía Mail

Q-flow maneja tres tipos de paso para interactuar con los usuarios:

- **NOTIFY**: Permite enviar una notificación sin esperar una respuesta por parte del destinatario. No se requiere que el mismo sea un usuario de Q-flow; puede ser un contacto externo.
- **ASK**: Permite solicitarle a un usuario que elija una respuesta de un conjunto de opciones. El paso puede estar asociado a varios destinatarios, caso en el que se realiza una votación entre ellos
- **TASK**: Permite modelar una tarea que puede tener una duración prolongada. Durante la misma, el usuario puede actualizar el progreso de la tarea, cambiando el estado entre:
 - No Iniciado.
 - En ejecución, indicando un porcentaje de progreso.
 - Terminada.
 - Cancelada.

Al ejecutar un paso de este tipo, el motor genera un mail para cada uno de los destinatarios de la tarea. El mismo contiene un formulario Web con:

- Información del flujo y del paso: Describen al usuario la tarea encomendada.
- Respuesta: Permite indicar el progreso / resultado de la tarea.
- Datos de aplicación: Permite tanto ver como editar la información mantenida por el flow.
- Archivos Adjuntos: Permite agregar / eliminar / modificar los Archivos Adjuntos al flow.

Paso ASK

• Respuesta del Flow (1/2)

ID	Descrip
Flow:	176 - rwrwr
Paso	43 - Pregunta (4)
Esperando desde:	23/04/2002 19:37
Respuesta:	<input type="text" value="Si"/>

• Datos de la aplicación

Item	Valor
Dato1	<input type="text" value="10"/>
DATO2	<input type="text" value="1024"/>
dato3	<input type="text" value="3"/>

• No hay archivos adjuntos

Paso TASK

• Respuesta del Flow (1/2)

ID	Descrip
Flow:	179 - test1
Paso	2 - Tarea (3)
Esperando desde:	23/05/2002 18:15
Estado:	<input type="text" value="No iniciado"/>
Progreso:	<input type="text" value="0"/> %

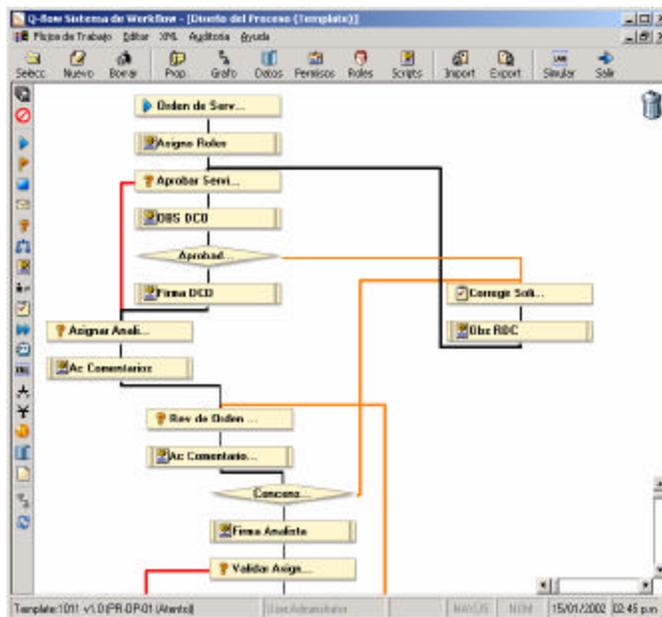
• Datos de la aplicación

Item	Valor
a	b
<input type="text" value="123"/>	<input type="text" value="123"/> *

• No hay archivos adjuntos

D.3.4 Diseñador de Plantillas

El diseñador de Plantillas de Q-flow permite definir en forma gráfica los procesos de negocio a automatizar, expresándolos en forma de tareas genéricas (preguntas, notificaciones, evaluaciones, puntos de control y de finalización, etc). Cada una de estas tareas es representada por un tipo de paso particular. Actualmente existen 19 de estos tipos. Los pasos que requieran intervención de Recursos Humanos, pueden tener configuradas acciones proactivas tales como Alertas, Recordatorios y Delegaciones, programadas en base a reglas de tiempo (dos horas antes de terminar, de empezar, el día xx/yy/zz, etc).



La herramienta permite también definir la información estructurada (y no estructurada) que maneja el proceso (número de factura, moneda, importe, documento de especificaciones, etc). La misma es representada en forma de “Variables de Proceso”. Cada variable tiene asociado un “Tipo de Dato Extendido”, el cual permite definir la forma en que las variables son desplegadas en los formularios y las validaciones que serán aplicadas a la entrada del usuario.

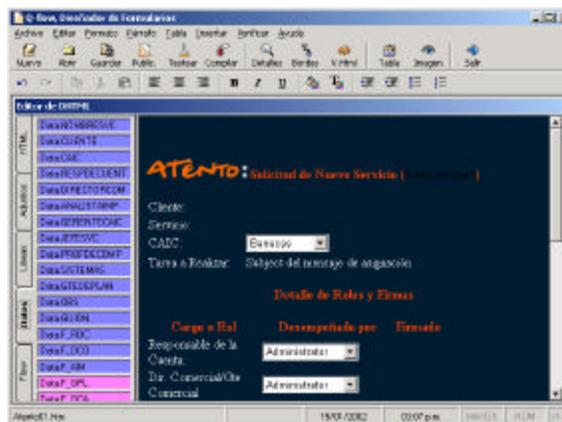
El diseñador también permite definir la visibilidad que tendrán las variables en cada paso del proceso, las cuales podrán estar ocultas, visibles o editables.

Por último, el diseñador permite definir roles propios del proceso, los cuales podrán ser asociados a un usuario tanto en tiempo de diseño como en tiempo de ejecución.

Toda la configuración de un Template puede ser exportada / importada entre instalaciones de Q-flow mediante archivos XML.

D.3.5 Diseñador de Formularios

Qflow genera automáticamente todos los formularios de ingreso de información necesarios para cada paso. Sin embargo, estos pueden ser rediseñados para adaptarse a las necesidades estéticas y funcionales del usuario.



Un formulario personalizado es un archivo HTML, que contiene tags de la forma “<tag %>” que representan información del flow. Antes de desplegar el formulario, el sistema sustituye estos tags por el valor de la variable o un campo de entrada adecuado al tipo de la variable. De esta forma, los formularios personalizados pueden ser diseñados con cualquier aplicación de desarrollo Web.

El Diseñador de Formularios provisto por Q-flow, permite construir estos formularios de forma asistida, simplemente arrastrando y soltando datos de aplicación sobre una hoja en blanco.

Bibliografía

- [ISO] – “Fundamentos y vocabularios” Norma UNIT-ISO 9000:2000.
- [WfMC] – “The Workflow Reference Model”. Código TC00-1003. Workflow Management Coalition. www.wfmc.org.
- [QFlow] – “Desarrollando soluciones de Workflow” QFlow 2.0 Urudata. 2001
- [Qpaper] – Documento “QPaper.doc”. <http://qflow.urudata.com/Documentos/Qpaper.doc>. Urudata. 2001
- [Biztalk] – “Microsoft BizTalk Server 2000 Documented”. Microsoft Press. 2001
- [BTFR] – BizTalk Framework 2.0 Document and Message Specification Microsoft Corp. 2001.
- [BTCurso] – Guía del Instructor del Curso 2379A: “Developing and Deploying Microsoft BizTalk Server 2000 Solutions. MSDN Training. Microsoft. 2001
- [Ult_ EAI] – White Paper “Ultimus for EAI: Bridging the Islands of Automation” - Ultimus Workflow Software. Abril 2001. www.ultimus.com/ultwhite/wp_bridging_islands.pdf
- [Ult_BT] – White Paper “Ultimus + BizTalk Server: Bringing Human and Application Process Integration to the Internet-driven Enterprise”. Ultimus Workflow Software. Junio 2001. www.ultimus.com/ultwhite/wp_ultimus_biztalk.pdf
- [T5WF] – “Implementación de un motor de Workflow”. Gonzalo Fernández. Proyecto de Taller 5. Facultad de Ingeniería, UdelaR, Montevideo, Uruguay. 2000.
- [Mohr] – Material de la presentación “Introduction to Microsoft BizTalk Server” de Stephen Mohr para las Wrox Conferences. Stephen Mohr, Omicron Consulting
- [Steffen] – Presentación “Workflow: Una herramienta para la optimización de procesos administrativos”. Hermann Steffen. Curso “Taller de Ingeniería de Software”. Facultad de Ingeniería, UdelaR, Montevideo, Uruguay.
- [GFI]. GFI – “Workflow Technology, an introduction”. <http://www.gficomms.com>
- [DocMan] – “DocMan EmailFlow White Paper”. DocMan A.G.
- [Travis] – Brian Travis, “XML and SOAP Programming for BizTalk Servers”. Microsoft Press.
- [W3Jour] – Conolly, Khare, Riffkin. <http://www.w3journal.com/xml/>. Volume 2, Number 4, Fall 1997, Pages 119-128
- [EBusiness] – Presentación “e-Business”. Curso “Administración y Seguridad de Sistemas” Jorge Merlino y Gustavo Degeronimi. Facultad de Ingeniería, UdelaR, Montevideo, Uruguay. 2001