Introducción

BIZUIT es una suite de productos diseñada para dar una solución integral a las necesidades de información de su organización. Las empresas, independientemente de su tamaño o mercado, enfrentan cuatro grandes problemáticas relacionadas a la administración de la información y la tecnología necesaria para lograr la eficiencia requerida para afrontar los retos y cambios que impone constantemente el mercado.
Estas cuatro problemáticas son:
a) Gestionar los procesos de negocios claves para la organización
b) Integrar de manera rápida, eficiente, escalable y flexible las aplicaciones existentes en el entorno empresarial así como nuevas herramientas que se incorporen al ecosistema informático organizacional
c) Desarrollar rápidamente nuevas aplicaciones robustas y con una sólida base arquitectónica que permitan hacer frente a cambios en estrategias de negocios
d) Gestionar la información resultante de los puntos anteriores.

Gestionar Procesos de Negocios Claves para la organización

Las organizaciones están en la constante búsqueda de herramientas, paradigmas o cualquier medio o mecanismo que colaboren a aumentar los beneficios, reducir costos, mejorar la productividad y la relación con los clientes.

Toda organización posee un conjunto de procesos que aportan el conocimiento necesario para desarrollar todas las tareas que requiere el funcionamiento de la compañía y las relaciones con el medio en donde se desarrolla (clientes, proveedores, socios, etc.). La eficiencia y efectividad en la ejecución de los procesos de negocios, es la diferencia entre un proceso altamente exitoso y lucrativo o un fracaso absoluto.
Orientar la Administración del Negocio hacia la Gestión basada en Procesos de Negocios es una de las mejores opciones para lograr los objetivos planteados. La Administración de Procesos de Negocios (BPM) ha emergido como una disciplina estratégica para ayudar a las empresas a mejorar sus procesos de negocios. Una iniciativa exitosa de BPM, debe sostenerse por medio de un equilibrio justo de dirección empresarial y tecnología, la cual permite una reducción significativa en costos de operación y los ciclos de tiempos, algunas veces hasta en un 90%. Esto se ve reflejado en todos los procesos de la organización, pero sobre todos en aquellos en los cuales intervienen varios departamentos, sistemas de información y personas.

BIZUIT brinda las herramientas necesarias para tomar el control de los procesos de su empresa, modelarlos, darles visibilidad y la posibilidad de tomar decisiones basadas en información real y oportuna del negocio a partir de ellos.
Esto, junto con la posibilidad de diseñar e implementar una arquitectura tecnológica acorde a la perspectiva estratégica de la organización, lleva al paradigma de BPM un paso más adelante.

Integrar de manera rápida, eficiente, escalable y flexible aplicaciones empresariales

EAI (Enterprise Application Integration) se define como la "Forma segura y orquestada de compartir procesos y/o datos entre aplicaciones dentro y fuera de una organización"
La implementación de una correcta estrategia de EAI permite a las organizaciones incorporar nuevas aplicaciones de manera mas eficiente y a menor costo, modificar a bajo costo procesos de negocios requeridos por la organización, automatizar procesos de negocios que requerían intervención humana, optimizar los procesos de negocios y obtener información consolidada y en línea para la toma de decisiones. En definitiva permite hacer mas eficiente la gestion de IT y sacar un mayor provecho de la infraestructura tecnologica existente en una organización, haciendo que todas sus aplicaciones hablen el mismo lenguaje: el de los negocios.

Cada proyecto de IT requiere obligatoriamente esfuerzos de integración, como ejemplo podemos citar que durante la implementación de un ERP, 2/3 del gasto en TI es de integración (Fuente: JDEdwards).
Estadísticamente el porcentaje del presupuesto de la Gerencia de IT asignado a integración es de un 40% a un 70%.

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

Otros datos de la industria nos revelan que sólo el 10% de las aplicaciones se encuentran interconectadas o integradas. De ese 10%, solo el 15% usa herramientas formales de integración, el resto utiliza mecanismos ad-hoc, mientras que el 85% restante utiliza arquitectura accidental. Analizaremos a continuación la problemática de la arquitectura accidental:

La arquitectura accidental es una estructura descentralizada donde cada aplicación se comunica de manera directa con las otras y es un modelo apropiado para organizaciones con pocas aplicaciones donde las interfaces entre las aplicaciones se escriben a medida que surgen las necesidades de integración.
Es un modelo que plantea una solución económica pero de corto plazo a la problemática de integración, ya que no representa un modelo escalable y mantenible, conduciendo inevitablemente a la creación de interfases de integración altamente acopladas que aumentan exponencialmente la complejidad de la solución haciendo prácticamente inviable las modificaciones a determinadas interfases debido al alto grado de acoplamiento entre ellas, ya que un cambio en una de las interfases puede provocar un efecto en cascada que provoque cambios a cientos de interfases. La formula para el calculo de interfaces altamente acopladas: $n(n-1)/2$, donde n es la cantidad de aplicaciones a integrar.

La arquitectura de integración accidental o punto a punto presenta una serie de desventajas muy importantes, que impiden su adopción en un entorno empresarial robusto, flexible y escalable:

- Rígida, Costosa y Difícil de Operar y Mantener
- Silos de información e islas de Integración
- Aplicaciones altamente acopladas, cero flexibilidad
- Difícil y lenta de monitorear o diagnosticar
- Complejidad creciente al aumentar las soluciones punto a punto.
- Baja Performance y Escalabilidad
- Lógica de ruteo embebida en la solución
- Ausencia de un modelo de seguridad uniforme
- Ausencia de estándares

Como alternativa a la integración accidental surge la arquitectura de integración basada en un hub que representa una estructura centralizada donde cada aplicación se comunica con las demás por medio de un intermediario o broker. Debido a que la complejidad en este modelo crece de manera lineal y no exponencial (Nro. de interfaces desacopladas es igual al nro. de conexiones) es un modelo apropiado para organizaciones con muchas aplicaciones que ofrece muy buena flexibilidad y escalabilidad.
Con la aparición del paradigma SOA (Service Oriented Architecture), surge una variante del modelo HUB denominada SOI (Service Oriented Integration), que logra los beneficios del modelo HUB desacoplando aplicaciones mediante interfases de servicios logrando de esa manera un footprint mayor que el ofrecido por el modelo HUB. BIZUIT utiliza este ultimo modelo de arquitectura de integración SOI.

Una plataforma de integración robusta como BIZUIT ofrece:

- Conectividad entre diferentes plataformas tecnológicas
- Procesamiento de reglas complejas de negocios y transformaciones de datos
- Soporte a procesos de negocio
- Adaptación a cambios en hard, soft y objetivos de negocio.
- Soporte a diferentes alternativas:
  - Web services
  - Extract, Transform, and Load (ETL)
  - Communications message protocols
  - Program calls
  - Direct data access
  - File transfer
  - Human involvement
- Exponer interfaz común para la comunicación de aplicaciones
- Utilizar protocolos de transporte existentes en la organización
- Aislar implementación de la tecnología mediante el uso de interfaces
- Permitir integrarse a nivel funcional o a nivel de datos

Desarrollar rápidamente nuevas aplicaciones para enfrentar cambios de estrategias

Los cambios casi constantes en las reglas de negocios de cualquier organización para adaptarse a las fluctuaciones del mercado, termina con la necesidad de adaptar ciertos procesos rápidamente, y con ello, modificar las aplicaciones informáticas que los soportan.
Las empresas, en su gran mayoría, recurren a programadores y/o personas con un alto conocimiento técnico para realizar las imperiosas adecuaciones. Esto trae aparejado tres inconvenientes:

1) Tener a disposición este conjunto de personas técnicas cada vez que la organización requiera un cambio en los procesos, en las reglas de negocio y/o en las aplicaciones existentes, o en el desarrollo de sistemas informáticos inexistentes.
2) Integrar las nuevas aplicaciones a la arquitectura existente.
3) El tiempo que demanda la puesta en marcha de los cambios.

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

Esto produce un ciclo incremental cada vez más complicado de administrar. Mientras más cambios se incluyan a nivel de código, más compleja se torna la actualización de los sistemas.
Entre otros aspectos negativos, uno de los más perjudiciales está vinculado al tiempo de respuesta, cada vez más lento del área de IT ante las necesidades del negocio.

Con la tecnología Zero Code de BIZUIT, las organizaciones prescinden de la programación para implementar cambios a nivel de procesos o reglas de negocios. De esta forma se minimiza la dependencia del departamento de IT, debido a que los cambios no se realizan en el código y normalmente no necesitan habilidades de programación o conocimientos técnicos avanzados.
Por otra parte, el departamento de IT, responde de manera mucho más rápida y eficiente ante el requerimiento de las áreas de negocio, construyendo interfaces, prototipos y hasta aplicaciones sin la necesidad de escribir código, enfocándose exclusivamente en la problemática del negocio.

Históricamente el concepto de RAD ha sido utilizado para categorizar a herramientas de programación que proveían mejoras en la productividad de los desarrolladores (Ej: Visual Basic) con respecto a la programación clásica procedural. Si bien estas herramientas efectivamente aceleraban el proceso de desarrollo, su mayor efectividad se encontraba en la generación de interfases graficas de usuario, mediante diseñadores visuales se podían generar pantallas de una manera muy fácil e intuitiva simplemente arrastrando controles de una barra de herramientas a nuestro formulario.
Sin embargo, el resto de nuestra aplicación -la codificación del acceso a datos, las reglas de negocio y lógica de procesos de negocios con interacción humana- debía ser codificado de la manera tradicional, lo que limitaba su efectividad como herramientas de desarrollo rápido.

Tampoco se contaba en estas herramientas con un soporte adecuado para el diseño, codificación y deploy de nuestras aplicaciones en entornos distribuidos.
E-RAD (Enterprise Rapid Application Development) en cambio, provee un enfoque visual para la creación de todas las capas de nuestras aplicaciones de una manera rápida y fácil. No solo permite la generación de interfases de usuario de manera visual sino que permite construir también de manera visual nuestros servicios, componentes de negocio y acceso a datos utilizando reglas, funciones y componentes prediseñados.
BIZUIT permite utilizar datos, componentes y reglas de negocios de diferentes fuentes y orígenes en una interfaz grafica intuitiva y, en cuestión de minutos, transformar esos datos en robustas aplicaciones SOA-enabled.

La barra de herramientas de BIZUIT incluye un conjunto de componentes y actividades de control de flujo que implementan las características típicas que un desarrollador necesita al momento de crear las diferentes capas de una aplicación. El paradigma BIZUIT E-RAD de codificación de reglas de negocio, servicios y acceso a datos consiste en modelar la logica seleccionando componentes de la barra de herramientas y arrastrandolos hacia nuestra superficie de modelado. Una vez alli, se procede a la configuración de estos componentes permitiendo su depuración grafica, lo que transforma a BIZUIT E-RAD Edition en un IDE (Integrated Development Environment) para la codificación, depuración y deploy de todas las capas de nuestras aplicaciones empresariales.
La tecnología BIZUIT Zero-Code, combinada con nuestro IDE provee resultados inmediatos. El proceso de modelado de nuestras aplicaciones no es solo un modelo util a los efectos de documentación, el modelo es el código fuente. La tecnología BIZUIT Zero-Code transforma ese modelo en código .NET 2.0 de manera automática. De esa manera, el modelo y el código fuente se encuentran sincronizados constantemente, facilitando no solamente las tareas de documentación sino también el mantenimiento de nuestras aplicaciones.

Gestión de la información

La constante y veloz evolución de todos los sectores de la industria lleva a que las organizaciones deban moverse en un entorno de gestión cada vez más complejo. La creciente competitividad de los mercados hace imprescindible mejorar la calidad del servicio ofrecido sin que ello signifique incrementar los costos y, que la detección de oportunidades o problemas en la actividad se realice lo más pronto posible. Para ello las empresas deben contar con información que les permita ejecutar acciones con rapidez, minimizando el riesgo y la pérdida de tiempo que implica la toma de decisiones basada en información desactualizada, incompleta o inconexa: la gestión de la información resulta fundamental para la aportación de valor.

Monitorear las actividades del negocio es un elemento clave en la mejora de los procesos y servicios de cualquier compañía. BIZUIT Dashboard es una herramienta de la suite que, mediante el uso de tecnología de BAM, aumenta las posibilidades del negocio a través de una estrategia de control y seguimiento proactivo y en tiempo real, que se alimenta de información proporcionada por la plataforma y arquitectura proporcionada por BIZUIT.

A partir de esta información se analizan Indicadores Clave de Rendimiento (KPIs por su sigla en inglés) para el óptimo funcionamiento del negocio, lo cual no solamente permite conocer el valor del indicador, sino también su estado en cuanto a lo deseado (objetivo, parámetro, etc.).

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

Además de información en tiempo real, es necesario contar con información histórica y de tendencia sobre el comportamiento de cada una de las variables establecidas.
Todo este volumen de **información**, debe ser **organizada** y presentada de forma **adecuada**, de lo contrario carecerá de sentido y practicidad.
BIZUIT Dashboard presenta la información en forma **clara** y **oportuna**, lo cual colabora en la toma de decisiones en forma **rápida** y **minimizando el riesgo** de dicha decisión.

Para resolver estas cuatro grandes problemáticas, BIZUIT se basa en una sólida arquitectura que combina de manera única características de

- Service Oriented Architecture (SOA)
- Business Process Management (BPM)
- Event Driven Architecture (EDA)
- Enterprise Rapid Application Development (ERAD).

El funcionamiento de BIZUIT se puede resumir mediante el concepto de Evento/Reacción. En BIZUIT todo evento recibido provoca una excitación del sistema que determina la ejecución de una reacción.

Un evento es un mensaje que le indica a BIZUIT que algo ha ocurrido en el exterior. Los eventos pueden ser de dos tipos: Manuales y Automáticos. Los eventos manuales son aquellos que son disparados por un usuario utilizando una aplicación. Los eventos automáticos son eventos disparados por aplicaciones sin que medie una interacción humana.

A su vez, los eventos manuales se pueden dividir en Iniciales y de Interacción. Los eventos manuales iniciales son los eventos que originan el comienzo de una reacción. Los eventos manuales de interacción son eventos que se disparan dentro de una reacción como consecuencia de la necesidad de requerir la interacción humana para continuar con el procesamiento de la reacción.

Ejemplos:

Evento	**Tipo**	**Disparado Por**
OrdenCompraSolicitada	Manual Inicial	Interfaz Grafica diseñada por BIZUIT Designer
OrdenCompraAprobada	Manual de Interaccion	Aplicación Legacy
StockMinimoAlcanzado	Automatico	Trigger de SAP

Proceso de Desarrollo BIZUIT

El proceso de desarrollo con la suite BIZUIT, sigue los lineamientos básicos de desarrollo SOA: Modelar, Ensamblar, Distribuir y Administrar.

Debido a que en toda reaccion BIZUIT hay una componente de conocimiento de la problemática del negocio que la reaccion intenta resolver, la metodología propuesta por BIZUIT sugiere un enfoque top-down partiendo desde el modelado a un nivel conceptual cercano a la problemática del negocio que se desea resolver.
Este modelado tendra diferentes formas en funcion de la reaccion que estemos desarrollando, asi si nuestra reaccion es un proceso de negocios, el modelado significara definir la secuencia logica del proceso de negocio en cuestion, sus roles, recursos, tareas y caminos alternativos. Si en cambio nuestra reaccion es una red de integración, se modelara a un nivel conceptual alto los grandes bloques funcionales que la integración implementara para solucionar una problemática de negocios. Por ultimo, si nuestra reaccion es un servicio,logica o regla de negocios con su correspondiente acceso a datos, el modelado consistira en disenar los componentes de negocio que resolveran la problemática en cuestion.

Una vez que la reaccion se ha modelado desde el punto de vista de negocios, es necesario comenzar con la tarea de implementar (ensamblar) la forma en la que el sistema informático va a comportarse, el sistema esta conformado por la lógica del proceso y por un conjunto de actividades que realizan tareas atómicas como ser interacción con bases de datos, conexiones con sistemas en plataformas heterogéneas, transformaciones de datos, interacción con usuarios, cálculos, etc.
Una vez implementado (ensamblado) el modelo, se da comienzo a las pruebas y la depuración del mismo. Una vez finalizada la etapa de depuración y pruebas se procede a la etapa de distribución del proceso de negocio y sus componentes de implementación, etapa que BIZUIT realiza de manera totalmente automática.
Por ultimo, una vez que el proceso de negocio ha sido puesto en funcionamiento, es necesario realizar un seguimiento de la ejecución del mismo, utilizando para ello BIZUIT Dashboard, herramienta que

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

permite analizar rendimientos, costos, tiempos, excepciones y demás características necesarias para la mejora en la eficiencia del proceso.

Durante el desarrollo de una reacción que implementa un proceso de negocio, podemos identificar los siguientes roles de usuarios:

- Analistas del Negocio (Information Workers)
- Implementadores Tecnicos (Developers)
- Equipo de Pruebas (Test)

Cada uno de estos roles cumple una funcion dentro de la metodología Modelar / Ensamblar / Distribuir / Administrar

Rol / Etapa	Modelar	Ensamblar	Distribuir	Administrar
Information Worker	●			●
Developer		●	●	
Test		●	●	

En cada una de las etapas se producen entregables para la etapa posterior, antes de dar por cerrada la etapa, el responsable de la etapa posterior debe dar su aceptación a los entregables recibidos.

La metodología Modelar/Ensamblar/Distribuir/Administrar sigue un proceso incremental e iterativo donde en cada iteración se cumplen las cuatro etapas de manera obligatoria. Al finalizar cada iteración se produce como entregable una nueva versión del proceso de negocios.

Componentes de BIZUIT

La suite de productos BIZUIT esta compuesta por cuatro herramientas que permiten resolver las problemáticas de negocios e IT anteriormente citadas:

- BIZUIT EventManager
- BIZUIT Designer
- BIZUIT Engine
- BIZUIT Dashboard

BIZUIT EventManager

Es uno de los componentes fundacionales de la arquitectura BIZUIT y el encargado de brindar la posibilidad de comunicarse con BIZUIT desde diferentes plataformas y tecnologías heterogéneas.

La función principal de BIZUIT EventManager es brindar una capa tecnológica para la recepción de eventos bajo diferentes protocolos de comunicación, utilizando para ello las características de Windows Communication Foundation (anteriormente conocida como Indigo).

Una aplicación o usuario que desea interactuar con BIZUIT lanza un evento, el cual es recibido por BIZUIT EventManager. Ejemplos de estos eventos son "OrdenCompraAprobada" , "PolizaIngresada", etc. Algunos de estos eventos pueden ser disparados por un usuario desde una interfaz legacy o confeccionada utilizando los Designers de BIZUIT, mientras que otros pueden ser disparados de manera automática desde aplicaciones existentes en la organización.

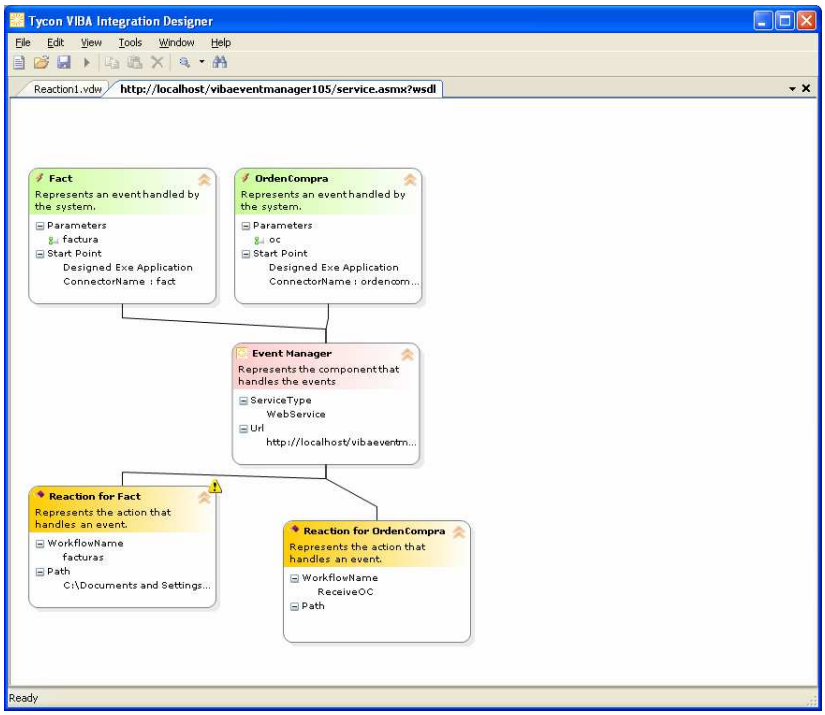
Los eventos recibidos por BIZUIT EventManager provocan una reacción, esta reacción puede representar la ejecución de procesos de negocios, integración de aplicaciones existentes o la ejecución de lógica y reglas de negocios.

Al utilizar WCF (Windows Communication Foundation) BIZUIT EventManager puede recibir eventos por medio de una amplia variedad de protocolos, como ser TCP, HTTP, FTP, SMPP y MSMQ.

Para determinar los eventos que BIZUIT puede recibir, que aplicaciones y/o usuarios pueden disparar estos eventos y cual será la reacción que BIZUIT ejecutará como respuesta al evento, se utiliza una interfaz grafica denominada EventManager Definition Designer, la cual nos permite rápidamente

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

visualizar los eventos configurados, su seguridad y reacciones, asi como también determinar el medio tecnológico y protocolo por el cual el evento llegara a BIZUIT.



Una vez que un evento llega a BIZUIT utilizando alguno de los protocolos anteriormente mencionados, la capa de seguridad de BIZUIT EventManager realiza tareas de autenticación y autorización, es decir que por un lado determina que las credenciales de la aplicación o usuario que originaron el evento son validas, luego de determinar la validez de las credenciales se procede a evaluar si con esas credenciales validas es posible ejecutar el evento solicitado.

Luego de aplicado este mecanismo de seguridad, la capa de Complex Event Processing (CEP) es la encargada de aplicar la lógica necesaria para determinar si el evento recibido junto con sus datos es de interés para BIZUIT.
La capa de CEP se utiliza como filtro de eventos no deseados, ya que BIZUIT puede estar subscripta a eventos disparados por aplicaciones externas, muchas veces no necesitamos que BIZUIT procese todos los eventos recibidos, sino solamente aquellos que debido a los datos que contienen son de interés para BIZUIT.
Un ejemplo de este filtro de eventos realizado por la capa de CEP es el caso en el que BIZUIT este subscripto a un evento disparado por una aplicación que cada 1 minuto indica el valor de la acción de una determinada compañía en la bolsa de valores. Nuestra regla de negocios puede requerir que reaccionemos solamente cuando el valor de la acción sea menor a U$S 5, descartando los demás eventos.

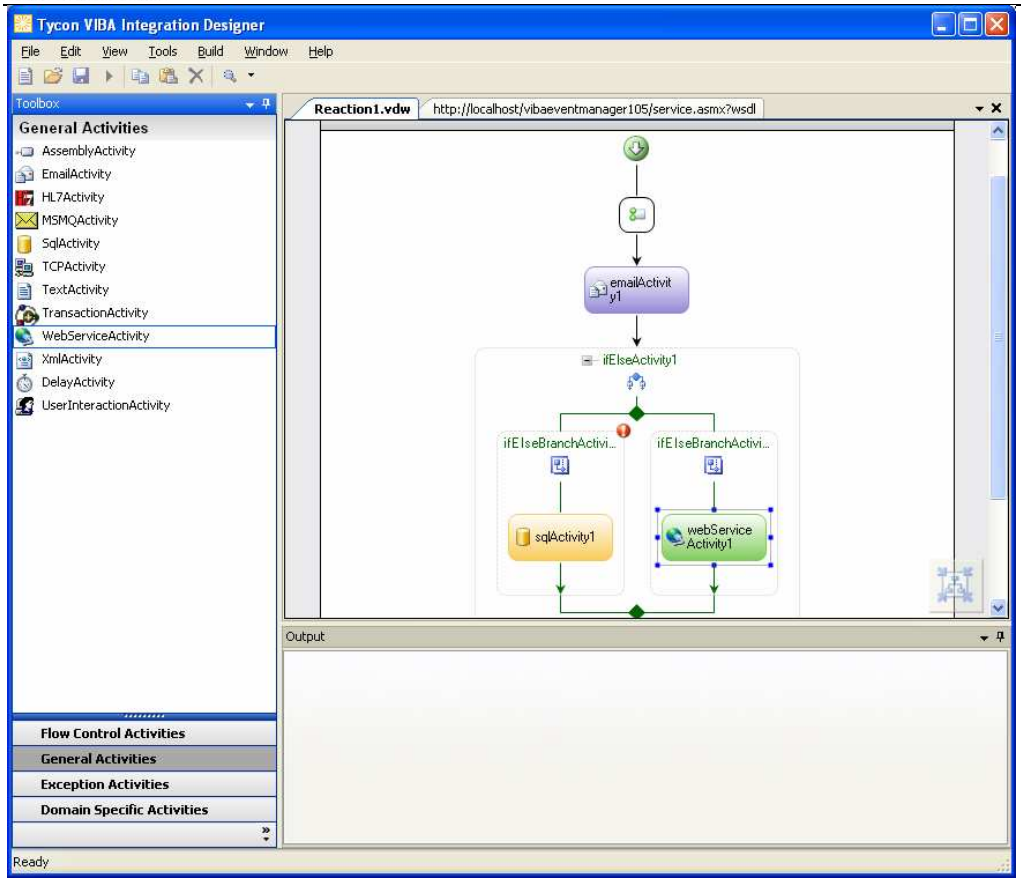
Luego de que la capa CEP determino que el evento recibido junto con sus datos, son de interés para BIZUIT, se ejecuta una capa denominada EventBinding and Dispatcher la cual es la encargada de inspeccionar la definición del evento realizada con EventManager Definition Designer y determinar la reacción a ejecutar. Una vez determinada la reacción a ejecutar, el control pasa a BIZUIT Engine, quien será el encargado de brindar el entorno de ejecución a la reacción configurada.

BIZUIT EventManager permite además la recepción asíncrona de eventos utilizando para ello mecanismos de encolado nativos a la plataforma Windows como MSMQ, lo cual permite la recepción asincrona de eventos, su entrega garantizada y ruteo mediante una arquitectura eficiente y escalable

BIZUIT Designer:

El diseñador de BIZUIT es una pieza clave dentro de la arquitectura BIZUIT ya que desde su interfaz es posible diseñar, probar y distribuir las reacciones a los eventos recibidos por BIZUIT EventManager. Estas reacciones representan la ejecución de procesos de negocios, integración de aplicaciones existentes y desarrollo rápido de aplicaciones.

Copyright © 2006 TYCON S.A. – Todos los derechos reservados



El designer de BIZUIT se basa en la premisa de que para adaptar la tecnología a los cambios en el negocio y mantener la competitividad comercial, es necesario elevar el nivel de abstracción del desarrollo de software en todas sus formas, confección de procesos de negocios, realización de integraciones de sistemas, desarrollo de interfases de usuario, confección de lógica de negocios, programación de servicios y acceso a datos.

Para cumplir con esta premisa, BIZUIT Designer ofrece características de diseño, desarrollo, pruebas y distribución únicas que permiten lograr drásticas disminuciones de tiempo en los proyectos informáticos. Un usuario típico de BIZUIT Designer es un analista de negocios que necesita desarrollar una pieza de código para dar soporte a una determinada problemática. Las características graficas de BIZUIT Designer posibilitan que un analista de negocios pueda desarrollar una solución para esa problemática a un nivel conceptual que esta dentro de su dominio del problema, sin necesidad de requerir conocimientos técnicos avanzados y logrando sin embargo una solución robusta, flexible y escalable basada en patrones, estándares y las mejores practicas del desarrollo de software.

BIZUIT Designer no solo es utilizada por analistas de negocios sino que es además una herramienta fundamental para el área de desarrollo de una organización.
Estadísticamente, durante el desarrollo de una determinada pieza de código el mayor tiempo se insume en plasmar en código fuente la imagen mental de la solución que el desarrollador arma en su mente. BIZUIT Designer es en si mismo un IDE de programación a un nivel de abstracción superior al de los IDEs tradicionales, donde el código no se escribe sino que se modela, logrando de esa manera plasmar mas rápida y fácilmente la intencionalidad del programa, es decir BIZUIT Designer mediante su tecnología de modelado de la programación permite trasladar rápidamente el modelo mental de la solución que el desarrollador confecciono a una pieza de código ejecutable. En BIZUIT, el modelo ES el código fuente, lo que permite un sinnúmero de ventajas con respecto a la programación tradicional entre las que podemos citar: velocidad de desarrollo, flexibilidad y adaptabilidad a cambios y documentación siempre sincronizada con el código fuente. Ese modelo es transformado en código ejecutable .NET de manera automática por la plataforma BIZUIT.

Con BIZUIT Designer podemos:

- Desarrollar visualmente Interfases de Usuario en tecnología WinForm y WebForm (ASP.NET 2.0) que estén enlazadas a datos del evento recibido por BIZUIT, así como a datos externos a BIZUIT.

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

- Desarrollar Servicios, Lógica de Negocios, Reglas de Negocio y Acceso a Datos de una manera simple, grafica e intuitiva
- Confeccionar redes de Integración de aplicaciones empresariales, utilizando para ello un conjunto de actividades específicas del dominio de integración prediseñadas.
- Modelar Procesos de Negocios utilizando actividades automáticas y manuales (aquellas que requieren intervención humana para continuar la ejecución del proceso)
- Asignar costos y recursos a las actividades manuales de un proceso de negocios de manera de luego analizar el costo total para la organización de la ejecución de un determinado proceso.
- Realizar transformaciones y mapeo de datos de manera grafica
- Ejecutar en modo de pruebas el modelo confeccionado con características de depuración visual que facilitan la detección de errores.
- Definir los eventos que BIZUIT recibirá, aplicarle seguridad y determinar cual será la reacción que se ejecutara ante la llegada del evento. Esta reacción puede ser un servicio, una lógica de negocios, una red de integración o la ejecución de un proceso de negocios.
- Construir de manera visual nuevas actividades especificas del dominio que se incorporen a la toolbox de actividades predisenadas de BIZUIT, permitiendo la reusabilidad de partes de nuestro modelo.
- Describir para servicios, redes de integración, procesos de negocios, lógica y reglas de negocios: el flujo de información, sus transformaciones y sus resultados.
- Documentar el modelo diseñado.

Ofrece:

- Mapeo gráfico de transformaciones basado en XSLT
- Transformaciones de datos predefinidas
- Motor de scripting para transformaciones complejas
- Interfaz orientada a usuarios sin conocimientos de lenguajes de programación.
- Control de errores y acciones compensatorias definidas y administradas de manera grafica.
- Motor Gráfico de Evaluación de Reglas de Negocio
- Debug Visual
- Documentador HTML de la solución
- Generación Automática de GUI multiplataforma con conexión a eventos de BIZUIT y fuentes de datos externas.
- Permite responder rápidamente a cambios en el modelo de negocios
- Mejora la productividad en las etapas de diseño e implementación al eliminar código.

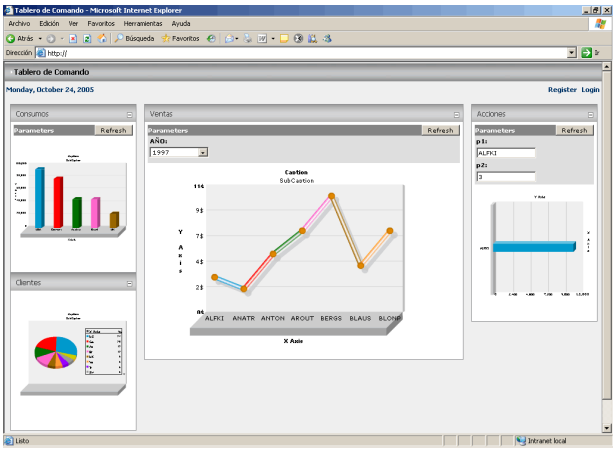
BIZUIT Engine:

Es el componente de ejecución de la reacción modelada por BIZUIT Designer. Es quien tiene a su cargo la ejecución de los servicios, lógica y reglas de negocios, procesos de negocios y redes de integración. Cuenta con una robusta plataforma 100% .NET 2.0 basada en Windows Workflow Foundation y con soporte a los principales estándares.
Posee robustos mecanismos de logueo de información, persistencia y seguimiento que permiten almacenar toda esa información para luego realizar tareas de seguimiento, optimizacion de procesos, cumplimiento con matrices SLAs, etc.

Mediante mecanismos de extensibilidad y configuracion es posible ampliar esta funcionalidad y cambiar el lugar donde se almacena esta información.

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

BIZUIT Dashboard:



- Capa de presentación basada en estándares para la arquitectura SOA de BIZUIT que ofrece información de negocios personalizada.
- Información obtenida de manera automática de sus aplicaciones empresariales integradas mediante Integration Plattform.
- Combina escalabilidad, seguridad y capacidades de Content Management.
- Permite monitorear y administrar proactivamente todos los aspectos de la plataforma de BIZUIT: Ejecución de Procesos de Negocios, Integración de aplicaciones.
- Mediante gráficos y reportes ofrece la información necesaria para la toma de decisiones en tiempo real: administración de eventos, alertas.
- Administra los KPIs (Key performance indicators) predefinidos y provee alertas cuando esos PKIS necesitan atención.
- La información analítica, reportes y alertas se encuentran integrados en una interfaz web-enabled amigable y flexible.

Resumen de Caracteristicas BIZUIT

- Modelado, ensamble, distribución y administración de procesos de negocios
- Creación de interfases visuales enlazadas a datos de procesos de negocio y redes de integración así como también a datos externos.
- Arquitectura
  - Orientada a Servicios con administración de eventos y mensajería
- Alternativas de Integración Soportadas
  - Web services
  - Extract, Transform, and Load (ETL)
  - Communications message protocols
  - Program calls
  - Direct data access
  - File transfer
  - Human involvement
- Actividades Predisenadas:
  - XML
  - DataBase
  - WebServices
  - MSMQ
  - FlatText
  - HL7
  - Assembly (COM y .NET)
  - Email
  - TCP
- Actividades de control de flujo:
  - If/Else
  - Paralell
  - Sequence
  - Do/While

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

- Actividades de Control de errores:
  - FaultHandler
  - ThrowException
- Diseñador visual de nuevas actividades
- Documentador de Servicios, Procesos de Negocios, redes de integración, lógica y reglas de negocios
- Integración a Nivel de Procesos de Negocios:

Característica	Descripción	Provista Por
Rules Processing	Permite definir reglas y procesarlas en tiempo de ejecución	Integration Engine, Conditional Activity, Calculation Functoid
Business Transaction Management	Asegura que si una actividad de negocio falla, las actividades relacionadas son canceladas.	Integration Fault Handler Engine, Transactional Activity y MSDTC
Workflow	Controla la interacción entre personas y procesos automatizados	Event Handler Activity, BIZUIT State Management
Orchestration	Coordina la ejecución de diferentes actividades que conforman un proceso de negocios	Integration Engine
State Management	Mantiene el estado en el contexto de aplicaciones pendientes	BIZUIT State management
Event Processing	Reconoce la ocurencia de eventos de negocio, almacena esos eventos y determina que proceso de negocio reacciona ante el evento.	BIZUIT Event Manager
Schedule	Genera eventos cuando transcurre un período de tiempo definido o en un momento en particular.	Delay Activity, Windows Server Schedule service and SQL Server Job Scheduler

- Integración a nivel de datos:

Característica	Descripción	Provista Por
Data Transformation	Modifica la apariencia o el formato de los datos (sin alterar su contenido)	BIZUIT Data Mapper, XSLT
Data Validation	Determina si un conjunto de datos cumple con un determinado criterio predefinido.	BIZUIT Data Mapper, XSLT, Custom Functions
Data Access	Determina cómo se recopilan los datos y se los presentan a las aplicaciones.	Integration Engine, BIZUIT Data Mapper, BIZUIT Activities
Schema Definition	Administra esquemas predefinidos.	Parameters, variables y activities manager. Third-parties tools
Mapping	Administra las relaciones entre aplicaciones y determina cómo transformar los datos entre las aplicaciones origen y destino.	BIZUIT Data Mapper, XSLT, Custom Functions
Schema Recognition	Verifica que un mensaje esté bien formado y que corresponda a un esquema predefinido de manera que pueda ser leído y procesado.	BIZUIT Activities, Event Manager

- Comunicaciones:

Característica	Descripción	Provista Por
Message Routing	Controla el ruteo de mensajes desde el origen a la aplicación destino	Event Manager, Integration Engine
Message Delivery	Determina cómo los mensajes pasan desde el origen a la aplicación destino	Event Manager, Integration Engine
Message Queuing	Implementa mensajeria garantizada	Event Manager, MSMQ, MQSeries

Copyright © 2006 TYCON S.A. – Todos los derechos reservados

Message Correlation	Asegura que los mensajes de respuesta se corresponden con los mensajes entrantes.	Event Manager
Message Instrumentation	Asegura que los mensajes son procesados en el orden en el que fueron enviados.	Event Manager, Integration Engine, UserInteraction Activity
Addressing	Determina adónde enviar los mensajes	Integration Engine
Transactional Delivery	Agrupa mensajes dentro de una transacción	Integration Fault Handler Engine, Transactional Activity y MSDTC
File Transfer	Mueve archivos entre aplicaciones	FileSystem Activity
Serialization/ Deserialization	Convierte datos desde y hacia una estructura plana para enviarlos a traves de la red.	Integration Engine, BIZUIT Activities
Batching/Unbatching	Agrupa mensajes para su transmisión y los separa en el destino	Event Manager, Integration Engine, Custom Activity, FileSystem Watcher Service
Encode/Decode	Posibilita la comunicación entre aplicaciones con diferentes code pages	BIZUIT Activities

- Seguridad:

Característica	Descripción	Provista Por
Authorization	Determina si un intento de conexion es permitido	Event Manager
Authentication	Verifica credenciales cuando una aplicación intenta realizar una conexión.	Event Manager y Dashboard Single Sign-On
Information Protection	Impide que usuarios no autorizados vean o modifiquen la información	Windows Server
Nonrepudiation	Utiliza firmas digitales para verificar la identidad	Digital signatures
Profile Management	Administración de roles	Dashboard User and Role Manager

- Administración Operacional:

Característica	Descripción	Provista Por
Business Activity Management	Administra, monitorea y analiza transacciones de negocios procesadas por la plataforma de integracion.	BIZUIT Dashboard, Event Monitor
Event Handling	Recibe eventos y procesa acciones	Event Manager, Integration Engine
Directory	Mantiene información de aplicaciones, suscripciones y servicios	Event Manager
System Monitoring	Determina si el hardware, SO y aplicaciones funcionan correctamente y dentro de los niveles de SLA establecidos.	Event Monitor

Copyright © 2006 TYCON S.A. – Todos los derechos reservados