



GeneXus y el ciclo de vida de las aplicaciones

Última actualización: Febrero de 2012

Copyright © Artech Consultores S. R. L. 1988-2012.

Todos los derechos reservados. Este documento no puede ser reproducido en cualquier medio sin el consentimiento explícito de Artech Consultores S.R.L. La información contenida en este documento es para uso personal únicamente.

Marcas Registradas

Artech y GeneXus son marcas o marcas registradas de Artech Consultores S.R.L. Todas las demás marcas mencionadas en este documento son propiedad de sus respectivos dueños.

Antecedentes: El viejo paradigma informático.....	3
GeneXus: un nuevo paradigma	5
El punto de partida	5
El diseño	6
La generación	6
La prototipación	6
El mantenimiento	7
La documentación y las ayudas	7
El trabajo en grupo.....	8
La reutilización del conocimiento (y su eventual comercialización)	9
Resumen.....	9

Antecedentes: El viejo paradigma informático

Cada vez existen más herramientas que, de alguna manera, pretenden aumentar la productividad en el desarrollo de aplicaciones.

El esquema original de desarrollo de programas consiste en combinar todas las acciones y reglas involucradas, organizarlos en un algoritmo y, luego, programar éste en un lenguaje de bajo nivel. La programación es procedural (o, más propiamente, procedimental).

Las primeras herramientas fueron los llamados “lenguajes de 4a. generación” que, si bien utilizaban el mismo esquema (procedimental), tenían una fuerte capacidad de expansión de código lo que permitía escribir mucho menos (y, también, equivocarse mucho menos) para obtener lo mismo. El impacto de estas herramientas sobre la productividad del desarrollo fue importante. En cambio, el impacto sobre los costos de mantenimiento fue muy pequeño: no existe comportamiento inteligente en la herramienta y, como consecuencia, es imposible un análisis integral del impacto de los cambios en la base de datos sobre los programas y, mucho menos, la propagación automática de esos cambios.

Otras herramientas importantes lo han sido los generadores de código. En este caso el sistema “entiende” las especificaciones y, en virtud de ello, genera el programa. Las primeras versiones eran muy rígidas pero, con el tiempo, se les ha ido agregando lenguajes procesadores de diagramas de acción (conceptualmente lenguajes procedimentales, muy semejantes a los lenguajes de 4a. generación).

Estas herramientas ofrecen una alta productividad para el desarrollo (mayor que la de los lenguajes de 4a. generación).

En lo relativo al mantenimiento, el aporte es pequeño: estas herramientas dependen de la base de datos (el diagrama E-R suele ser un “input”) y modificaciones en ese diagrama E-R pueden dejar inválidos múltiples procedimientos. O sea: tampoco aquí existe comportamiento inteligente de las herramientas y, como consecuencia, el análisis de impacto de los cambios y su eventual propagación es bastante limitado.

Si bien no han sido muy importantes (desde el punto de vista de su mercado) existen también herramientas similares a las anteriores que, en vez de generar código, interpretan las especificaciones. Esto no cambia nada: cuando una especificación pierde validez por cambios en la base de datos, el hecho de que se generen programas o se interprete la propia especificación no hace a lo esencial del problema.

El problema más importante de todos estos tipos de herramientas es que no son capaces de propagar automáticamente a las especificaciones los cambios sobre la base de datos. Entonces se usa, con ellas, la hipótesis "la base de datos es estable" y, lamentablemente, esa hipótesis es falsa.

Problemas semejantes existen con la llamada programación orientada a objetos: cuando se trata de lógica, incluso muy compleja, pero que no necesita del acceso a la base de datos (como en el caso de diálogos sofisticados), todo funciona muy bien. Cuando es necesario el acceso a la base de datos, reaparecen los anteriores problemas.

GeneXus: un nuevo paradigma

El punto de partida

Se parte de los objetos de los usuarios, que son muy bien conocidos por estos. GeneXus captura el conocimiento existente en esos objetos y lo sistematiza en una base de conocimiento.

Lo que GeneXus hace, intrínsecamente, es una muy buena administración automática del conocimiento. El conocimiento es esencialmente incremental: de una forma incremental aprendemos y pensamos. La consecuencia más importante de esta muy buena administración automática del conocimiento es un comportamiento inteligente de GeneXus.

La forma de expresar el conocimiento es muy importante: si somos capaces de expresarlo en una forma pura, mantiene todas sus características, todo su valor (y, en particular, la representación de cada objeto depende de ese objeto pero no de la representación de los demás objetos que puedan interesar al sistema). Este esquema de representación es el ideal porque, por ejemplo, si necesitamos modificar un objeto, tan sólo debemos modificar su representación, con total prescindencia de los demás.

Si, en cambio (viejo paradigma), para representar un objeto necesitamos descender a elementos físicos (como los archivos) la representación pierde mucho poder expresivo con respecto al conocimiento puro y, en particular, todos los objetos pasan a ser dependientes de los archivos y, cuando esos archivos cambian, muchas especificaciones se tornan inválidas y el sistema es incapaz de modificarlas automáticamente para restaurarles la validez. Como consecuencia, las especificaciones deben estudiarse una a una, lo que determina los importantes costos de mantenimiento por todos conocidos.

En GeneXus cada objeto se especifica en forma autocontenida sin nunca referirse a archivos ni a cualquier otro elemento de bajo nivel. Una consecuencia de lo anterior, que será luego muy importante, es que la base de conocimiento es neutra con relación al ambiente: arquitectura, hardware, sistema operativo, sistema de gerencia de base de datos, etc.

El diseño

GeneXus diseña la base de datos y los programas de aplicación. El diseño de la base de datos es un proceso determinístico: dado un conjunto de objetos del usuario existe una única base de datos relacional mínima que lo satisface. Luego de muchos años/hombre de investigación Artech ha conseguido hacerlo automáticamente.

Las bases de datos que diseña GeneXus están en tercera forma normal y tienen los índices estrictamente necesarios. Sin embargo, a veces, es muy adecuado (por razones de performance) introducir ciertas redundancias de datos y ciertos índices adicionales. GeneXus da al analista indicaciones de que redundancias o índices puede ser adecuado definir y le permite hacerlo y, a partir de ese momento, se hace responsable por su mantenimiento.

La generación

Partiendo del conocimiento sistematizado GeneXus genera automáticamente la aplicación (base de datos y programas) para la plataforma que se haya escogido.

Desde un punto de vista lógico, lo que debe hacerse es independiente de la plataforma. Físicamente, sin embargo, no lo es: cada lenguaje, cada sistema de gerencia de base de datos, cada sistema operativo, cada arquitectura, tiene comportamientos diferentes. GeneXus resuelve este problema dividiendo la generación en dos: lógica (la más importante, la más sofisticada) que es común a todas las plataformas, y física, que se construye plataforma a plataforma de modo de optimizar los programas generados para cada una de ellas.

La prototipación

GeneXus genera fácil y rápidamente programas funcionalmente equivalentes para múltiples plataformas.

Muchas veces las plataformas de ejecución son complejas, costosas o no disponibles en el momento del análisis. Con GeneXus se genera de inmediato una aplicación funcionalmente equivalente a la que se está desarrollando que funciona en un PC o, incluso, en un

“notebook”. Ello permite al usuario probarla de inmediato (lo que probablemente no evita errores, pero permite detectarlos rápidamente, cuando su corrección es muy fácil).

En vez de escribir en papel copiosas especificaciones, como es habitual y, luego, hacerlas aprobar por el usuario (que difícilmente las entenderá), se trata de discutir con ese usuario y mostrarle funcionando el prototipo producto de esa discusión en su propia oficina y muy rápidamente.

Más allá de las razones técnicas que hacen muy importante la prototipación, este esquema de trabajo modifica mucho la actitud del usuario: ahora, en vez de sentarse en la vereda de enfrente a esperar (para, generalmente, criticar luego), es y se siente participante, ve evolucionar el sistema “su sistema” lo que crea una sinergia altamente positiva.

El mantenimiento

A nadie le gusta el mantenimiento, pero es totalmente necesario: los cambios son inevitables para mantenerse competitivos. En EE.UU. se manejan estadísticas que dicen que de los recursos teóricamente destinados al desarrollo, tan sólo el 20% se destinan a ese fin mientras que el restante 80% se dedican, realmente, al mantenimiento.

Siempre es necesario modificar sistemas para acompañar las necesidades de una organización, para permitirle a ésta mantenerse al día, dar buenos servicios, tomar buenas decisiones y, en general, ser competitiva.

¿En que consiste el mantenimiento con GeneXus? Consiste en determinar cuales son los objetos que, de acuerdo a las necesidades de la realidad (nunca del sistema) deben modificarse y modificarlos: GeneXus hace, automáticamente, el resto.

Los cambios pueden ser simples o complejos. Requerirán o no grandes modificaciones en la bases de datos. Todo es transparente porque GeneXus lo resuelve.

La documentación y las ayudas

Uno de los problemas clásicos, con el que muchas veces nos hemos topado al tratar de mantener sistemas o programas sin GeneXus, lo ha sido la falta de una documentación fiel. Siempre hemos querido una buena documentación. Ahora la tenemos.

La base de conocimiento de GeneXus mantiene, activamente, una documentación completa de la aplicación. Todo ello puede, en un momento dado, imprimirse o grabarse en disco, etc.

Tenemos los más diversos listados, completos "cross-references", etc. También tenemos diagramas E-R.

¿Cuál es la diferencia entre estos diagramas E-R y los que se utilizan en otros sistemas? Existe una enorme diferencia: en otros sistemas los diagramas E-R son un "input" esencial del sistema y, su propia rigidez, determina luego la rigidez del sistema. Con GeneXus, en cambio, los diagramas E-R son, simplemente, subproductos del sistema, una ayuda visual para comprender mejor la estructura de la bases de datos que GeneXus ha diseñado para nosotros.

¿Es bueno mantener una documentación escrita imprimiendo los diferentes elementos que nos da GeneXus? No: es completamente inútil. Toda documentación escrita se torna obsoleta al poco tiempo. La única función de las salidas impresas de la base de conocimiento es visualizar localmente algunos elementos en un momento dado y no tiene sentido alguno guardarlas. O sea: podemos ver lo que queramos en cualquier momento pero eso no es lo importante. Lo importante es que GeneXus utiliza el conocimiento sistematizado que posee para hacer el mantenimiento automático de nuestras aplicaciones.

El trabajo en grupo

GeneXus brinda completas opciones para el trabajo en grupo:

Uso "off-line"

- Una capacidad esencial de GeneXus es que permite distribuir y consolidar inteligentemente el conocimiento de manera que pueda trabajarse por separado.
- Existen múltiples aplicaciones para esto pero una fundamental es la de desarrollar por separado partes de un sistema e ir prototipándolo directamente con los usuarios involucrados (por ejemplo en un "notebook") y, cuando está aceptado, consolidarlo automáticamente con el resto.
- GeneXus, para viabilizarlo, suministra automáticamente un completo análisis de impacto previo a la consolidación y, una vez aceptado éste por el analista, consolida el conocimiento automáticamente.

O sea: la aplicación puede dividirse en tantas partes como se quiera y la posterior consolidación de las mismas es automática y con todas las garantías.

Uso "on-line"

En un mismo modelo GeneXus pueden trabajar simultáneamente varios analistas definiendo independientemente, por ejemplo, Procedimientos, Reportes, Work Panels, Web Panels, etc.

Aquí no habría que mencionar GXserver?

La reutilización del conocimiento (y su eventual comercialización)

Una característica importante de GeneXus (la cual es aprovechada cotidianamente en el trabajo en grupo) es la reutilización del conocimiento.

Con GeneXus podemos tomar objetos de una aplicación para utilizarlos (directamente o luego de una sencilla adaptación) en otra o bien licenciar conocimiento de terceros de manera de hacer más cortos nuestros ciclos de desarrollo.

La reutilización ha sido siempre una vieja aspiración. Generalmente la industria ha tratado de reutilizar código. Se han obtenido tradicionalmente buenos resultados cuando se trata de, por ejemplo, funciones matemáticas, estadísticas, etc. y, en los últimos años, cuando se trata de objetos (los utilizados, por ejemplo, en diálogos sofisticados) que no necesitan del acceso a la base de datos. El problema del acceso a la base de datos ha sido una barrera infranqueable para la reutilización de código.

En el caso de GeneXus, la reutilización se hace a un nivel mucho más alto, a nivel del conocimiento y no presenta problema alguno.

Resumen

GeneXus apoya al analista en todo el ciclo de vida de la aplicación:

- **Se utiliza desde el principio.** Evita las descripciones en papel: desde le principio, desde los primeros diálogos el conocimiento es guardado y sistematizado en la base de conocimiento GeneXus, que constituye una documentación activa y permanentemente actualizada.
- **Modelo Estratégico.** El estudio de una aplicación comienza por la construcción de un modelo estratégico, que contiene los objetos de interés general de la misma (o, en su caso, de la organización entera). Todo esto es administrado por GeneXus.
- **División en partes.** Luego se divide la aplicación en partes, se distribuye a cada una de esas partes el conocimiento general que pueda ser necesario y se la desarrolla independientemente en permanente contacto con el usuario, prototipando cada paso (debe quedar bien claro que esta división no es una imposición de GeneXus sino una posibilidad adicional que proporciona).
- **Consolidación Automática.** Cuando las partes van quedando probadas y aceptadas por los usuarios, se las consolida automáticamente.
- **Base de Datos Corporativa.** Esta capacidad de consolidación nos permite trabajar a nivel de la aplicación y, luego, a nivel corporativo lo que nos permite un óptimo muy difícil de alcanzar de otra manera: tener una Base de Datos Corporativa con la que interactúan todos los sistemas de la empresa. La existencia de esta Base de Datos Corporativa es una invaluable fuente de información para la toma de decisiones.
- **Multi Plataforma.** El sistema consolidado es generado para las o las plataformas soportadas por GeneXus que se desee. Debe quedar claro que GeneXus no utiliza lenguajes propietarios o interpretadores sino que genera programas en los lenguajes estándares del mercado (Microsoft .NET, JAVA, Visual Basic, RPG, COBOL, etc.falta SD), con las siguientes ventajas para el cliente:
 - Los programas son de disponibilidad absolutamente libre para uso del cliente, sin necesidad de pago de "run-times", etc.
 - Como no hay partes interpretadas, los programas son los más eficientes posibles.
 - Sólo es necesario GeneXus para el desarrollo y mantenimiento. El cliente no necesita contratar licencias para utilizar en tiempo de ejecución.
- **Mantenimiento.** Cuando debe modificarse algún objeto del usuario, se modifica directamente su descripción, se pide a GeneXus un análisis de impacto y, una vez aceptado este, se ordena a GeneXus propagar dicho impacto. Todo esto es automático.
- **Documentación activa, permanentemente actualizada.** En cualquier momento puede consultarse total o parcialmente la documentación que se desee incluyendo descripciones de los diversos elementos, "cross-references" entre ellos, diagramas E-R, etc.
- **Libertad.** Si el cliente quiere migrar de un sistema operativo a otro, de un sistema de gerencia de base de datos a otros, etc. puede hacerlo muy fácilmente.
- **Protección y rápido aprovechamiento de los cambios tecnológicos.** Si en algún momento se modifica la tecnología (por ejemplo el cliente utiliza un iSeries en arquitectura centralizada y quiere que total o parcialmente sus aplicaciones pasen a trabajar en arquitectura de múltiples capas utilizando el lenguaje JAVA, lo que implica dramáticas modificaciones de todo), podrá hacerlo muy fácilmente, reaprovechando integralmente las especificaciones anteriores.

GeneXus se comercializa en toda América Latina, USA, Canada, España, Italia, China y Japón. Hoy existen más de 6000 clientes que contrataron más de 70.000 licencias. La gran

mayoría de estos clientes, en todo tipo de actividades, utilizan GeneXus para desarrollar grandes aplicaciones de Misión Crítica sobre Bases de Datos Corporativas.

Artech trabaja mucho para que sus clientes le tengan la suficiente confianza para considerarla su agente de tecnología: para que ellos se puedan dedicar sin sobresaltos a su negocio y a los sistemas de su negocio, para que puedan ser más competitivos y exitosos en su actividad fin, dejando la tecnología en manos de GeneXus.