

Elección de tecnología para la capa de presentación de SOA

Huibert Aalbers
Certified Senior Software IT Architect

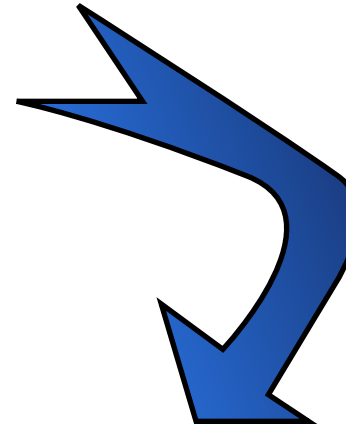


IT Insight podcast

- Este podcast pertenece a la serie IT Insight
- Pueden suscribirse al podcast a través de iTunes.
- El material adicional (presentación en formato PDF, white papers y otros) se obtienen directamente del sitio <http://www.huibert-aalbers.com> en la sección de IT Insight
- Me pueden enviar comentarios acerca del podcast o preguntas a mi correo personal, huibert_aalbers@mac.com

La capa de presentación de SOA

www.huibert-aalbers.com

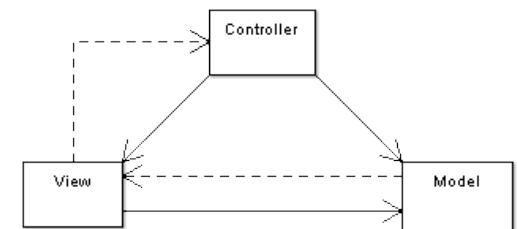


El problema

- HTML fue diseñado para presentar información de manera sencilla y eficiente
- HTML nunca fue pensado para capturar información de formas complejas ni para manipular estos datos localmente de manera flexible
 - HTML no resuelve el problema de la oficina sin papel, que tantas empresas quieren lograr
 - Muchas aplicaciones, en especial aquellas que requieren de interfaces de usuario complejas, son muy difíciles de desarrollar en un ambiente web
- Todo el mundo reconoce que la capa de presentación debería ser creada por diseñadores gráficos, sin embargo, la dificultad de crear interfaces complejas hace que muy a menudo esa tarea quede a cargo de programadores

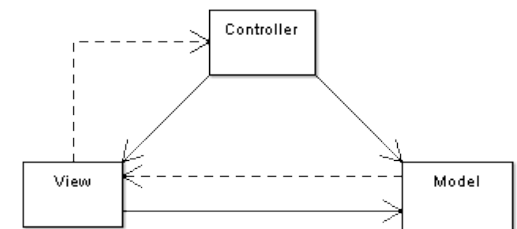
El patrón Model-View-Controller (MVC)

- Para separar la capa de presentación de la de lógica de negocios, se creó el patrón Model-View-Controller
- Se trata de un patrón de diseño en el que se manejan tres tipos de módulos
 - Vista - La capa de presentación, normalmente creada por los diseñadores gráficos, con la que interactúan los usuarios
 - Controlador - La capa de lógica de negocios que recibe datos de la Vista, los procesa y le devuelve los resultados para que sean presentados de manera atractiva a los usuarios
 - Modelo - La representación de los datos que manipula la aplicación



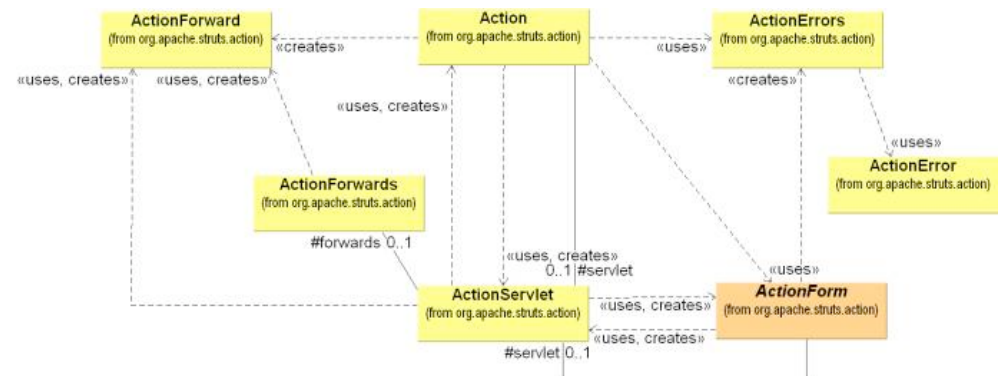
El patrón Model-View-Controller (MVC)

- El patrón MVC se desarrolló originalmente para aplicaciones cliente/servidor creadas en Smalltalk y hoy en día sigue siendo muy popular en frameworks como Cocoa (MacOS X), Swing o MFC.
- El mundo Java adoptó el patrón MVC para las aplicaciones Web
 - La capa de presentación (Vista) está formada por JSPs
 - El controlador suele implementarse como uno o varios Servlets
 - El modelo de datos puede ser implementado a través de componentes que representan objetos en la base de datos (EJBs o POJOs con Hibernate, etc.)



Struts

- Struts es la implementación open-source más popular del patrón MVC (Model 2) para Java, mantenida por la Fundación Apache
- La mayoría de las herramientas de desarrollo Java permiten crear aplicaciones basadas en Struts
- Sin embargo, a pesar de sus ventajas, Struts es un claramente un framework para desarrolladores que no simplifica realmente el trabajo de los diseñadores que tienen que crear páginas complejas



Struts



Java Server Faces (JSF)

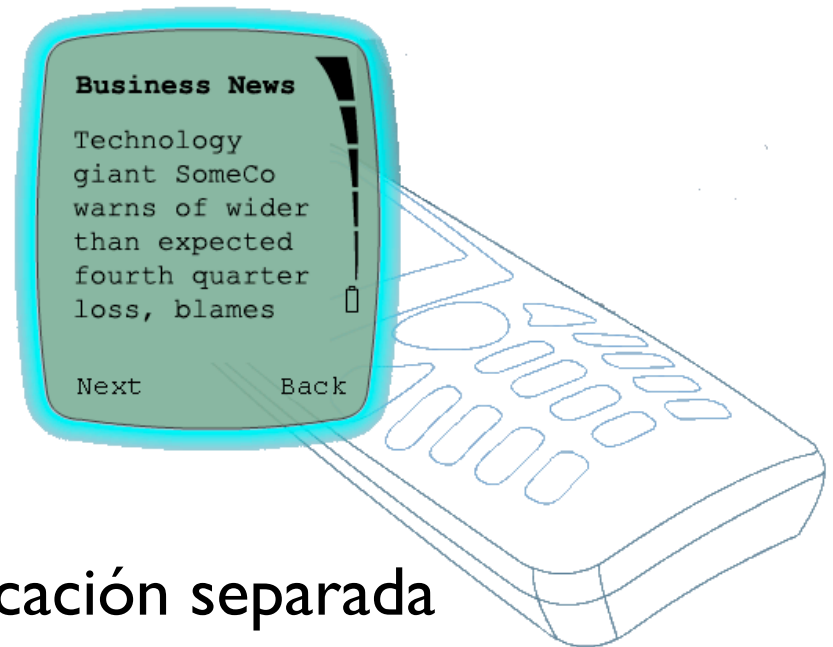
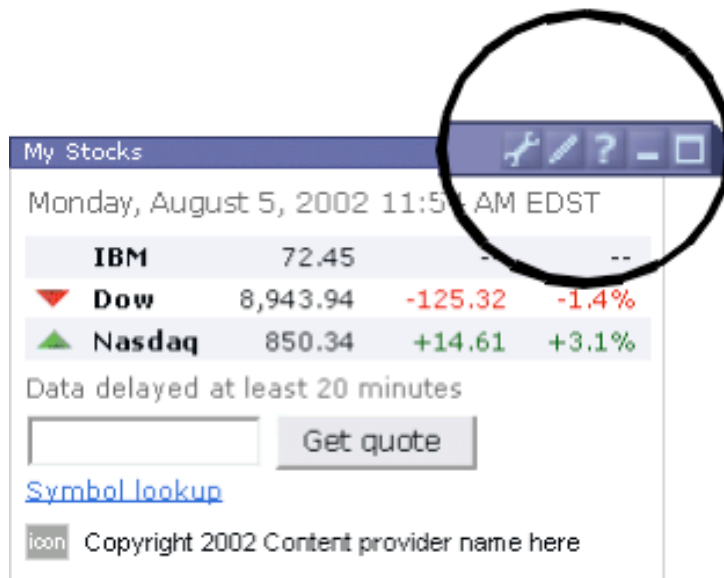
- Se trata de un estándar de Java (JSR-127)
- Es una solución que simplifica el desarrollo de páginas utilizando componentes reutilizables tales como tablas, calendarios, árboles, etc. y sustituye a Struts
- Tiene muchas ventajas para los desarrolladores J2EE
 - Está diseñado para integrarse de manera natural en herramientas de desarrollo Java
 - Se puede utilizar tanto con JSPs como Portlets (JSR-168)
 - Existe un mercado dinámico de componentes JSF
 - La ventaja es que los componentes JSF están basados en tecnologías estándar que se pueden visualizar en cualquier browser moderno

Portlets (JSR-168)

- Un portlet es conceptualmente similar a un servlet con la diferencia de que en lugar de generar una página completa, genera un fragmento de página
- Los portlets normalmente están ligados a un perfil de seguridad. Esto permite que solo los usuarios que tengan un determinado rol puedan usarlo.
- Simplifica la adopción de SSO (Single Sign-On) y la configuración de las páginas en base a las preferencias de los usuarios
- Soportado por algunos servidores de aplicaciones y la mayoría de servidores de portales
- Ejemplo: <http://www.usda.org>



Portlets (JSR-168)



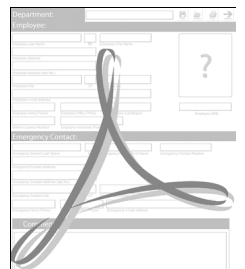
- Cada portlet es una aplicación separada
 - Se desarrolla de manera independiente
 - Se asocia a un perfil de seguridad y puede tener modos y estados
- Soportan distintos tipos de dispositivos
 - Celulares, PDAs, voz
 - Vistas únicas para cada tipo de dispositivo, aunque pueden compartir la lógica de negocios

XForms

- XForms es un estándar de la W3C que permite representar formas complejas en un archivo XML
- El estándar es muy completo pero eso hace difícil crear editores y visores de archivos XForms
 - Validaciones de campos usando web-services
 - Firmas digitales de segmentos de una forma
- Actualmente los browsers requieren de un plug-in para poder desplegar formas XForms
 - Firefox probablemente sea el primer browser en soportar XForms de manera nativa
 - No existen aún plug-ins para todos los browsers en todas las plataformas

PDF Forms

- Los archivos PDF son un estándar para documentos electrónicos que tienen que poder ser utilizados en cualquier plataforma
- Es natural que Adobe haya extendido el formato para soportar formas ya que esos documentos normalmente en algún momento tienen que imprimirse
- La gran ventaja de PDF es que los documentos se pueden ver en cualquier browser
- Adobe ofrece una solución que permite mapear los datos XML capturados a bases de datos



AJAX

- Utilizando HTML, JavaScript y CSS es posible crear atractivos clientes web de aplicaciones empresariales creadas en otros lenguajes (por ejemplo J2EE)
- La principal ventaja de AJAX es que está basado en estándares abiertos y funciona en la mayoría de los browsers modernos sin necesidad de plug-in
- Sin embargo, AJAX tiene una serie de inconvenientes
 - Solo se puede conectar al back-end utilizando HTTP
 - Las páginas HTML que utilizan AJAX pueden llegar a ser muy complejas y aún no existen buenos entornos de desarrollo para esta tecnología

Flash (Flex)

- Adobe Flex 2 es una herramienta de desarrollo de aplicaciones web que se visualizan utilizando el plug-in de Flash
- Herramienta de desarrollo basada en Eclipse
- Interactua de manera nativa con Servicios Web, XML y (algunas) bases de datos
- Ofrece una serie de ventajas interesantes
 - Interfaz muy atractiva visualmente, sin consumir casi ancho de banda
 - Se puede utilizar en JSPs y portlets
 - Buen entorno de desarrollo basado en Eclipse



Flash (Flex)

- Sin embargo, Flex no es una solución completa, ni perfecta
 - No se puede integrar fácilmente con colas de mensajes, bases de datos legadas y muchos ERPs
 - No existe un modelo de seguridad general y robusto
 - Requiere el aprendizaje de nuevas tecnologías
 - MXML
 - ActionScript
 - Tecnología propietaria
 - Requiere un plug-in (aunque es muy popular)
 - Es un problema, para dispositivos móviles

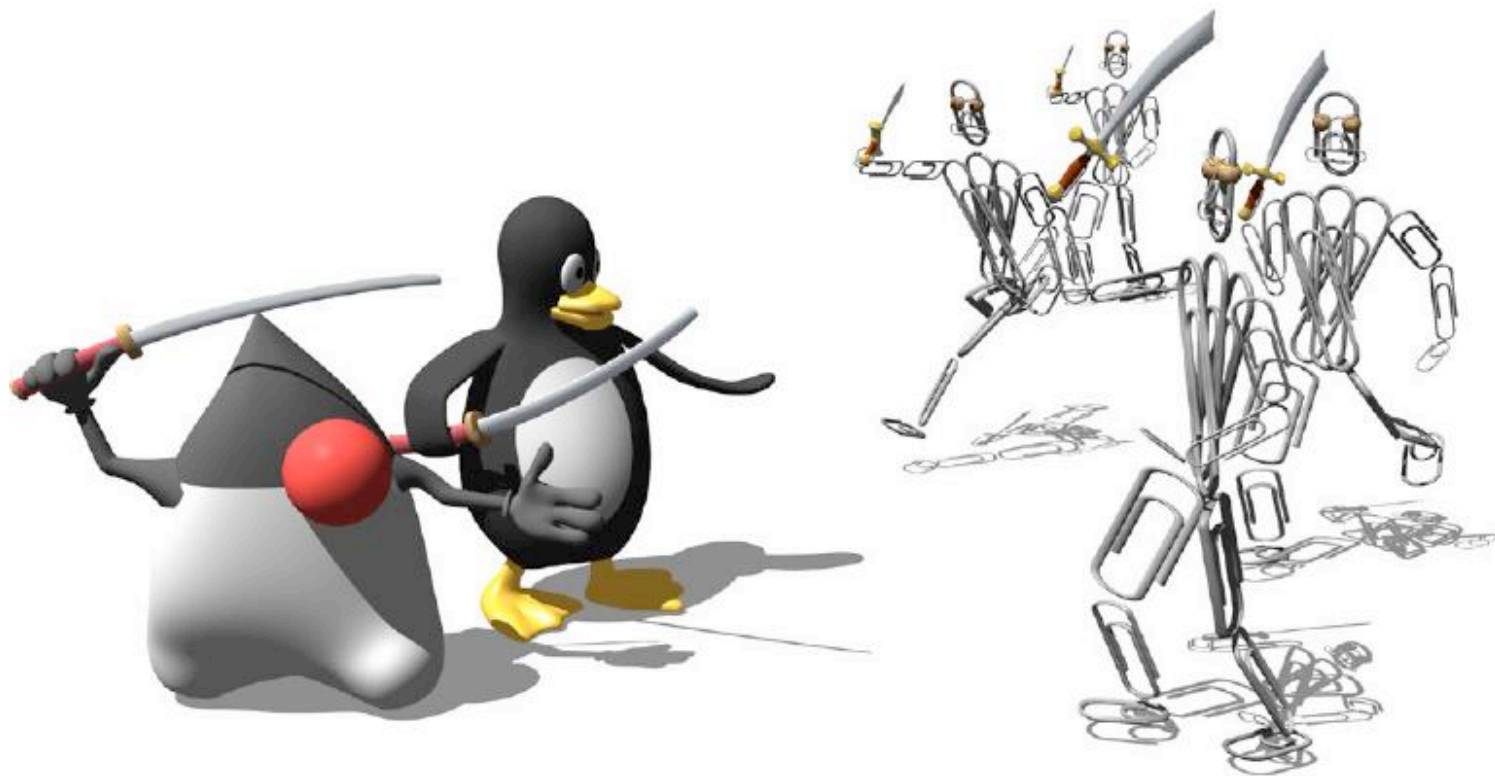


Conclusiones

- Además de las opciones que hemos discutido en esta presentación, existen otras muchas más que permiten desarrollar la capa de presentación de una arquitectura SOA
- Esto se debe a que ninguna resuelve de manera satisfactoria las necesidades de todos los usuarios
 - Algunas intentan resolver problemas de diseño de interfaces complejos
 - Otras intentan permitir que el browser pueda realizar tareas más complejas que las que actualmente permiten XHTML, JavaScript y CSS
 - Finalmente, otras intentan lograr una coherencia arquitectural que permita que todas las piezas encajen perfectamente
- Mi recomendación es buscar soluciones basadas en estándar no propietarias

Gracias

www.huibert-aalbers.com



Para mayor información, contácteme directamente
huibert_aalbers@mac.com