Historia del Proyecto

by Andrew C. Oliver, Agustín Martín Barbero

1. Breve Historia del Proyecto

El proyecto POI se gestó tiempo atrás, cerca de abril de 2001, cuando Andy Oliver obtuvo un contrato de corta duración para realizar informes Excel basados en Java. Ya había realizado este proyecto unas cuantas veces antes, y sabía exactamente dónde buscar las herramientas que necesitaría. Irónicamente, el API que solía utilizar se había disparado en precio desde unos $300 ($US) hasta unos $10K ($US). Calculó que a dos personas les llevaría unos seis meses el portar Excel así que le recomendó al cliente que pagase los $10K.

Cerca de junio de 2001, Andy empezó a pensar lo genial que sería tener una herramienta Java de código abierto para hacer esto y, mientras tuvo algo de tiempo libre, comenzó el proyecto y aprendió cosas sobre el Formato de Documento Compuesto OLE2. Tras chocarse con varios obstáculos insalvables, se dio cuenta de que necesitaría ayuda. Publicó un mensaje en su Grupo de Usuarios Java local (JUG) y preguntó si alguien estaba interesado. Tuvo mucha suerte y el programador Java de mayor talento que había conocido nunca, Marc Johnson, se unió al proyecto. A Marc le llevó unas pocas iteraciones el obtener algo con lo que estaban contentos.

Mientras Marc trabajaba en eso, Andy portó XLS a Java, basándose en la biblioteca de Marc. Varios usuarios escribieron peticiones para poder leer XLS (no sólo escribirlo como había sido planeado originalmente) y un usuario tenía peticiones especiales para un uso diferente de POIFS. Antes de que pasara mucho tiempo, el alcance del proyecto se había triplicado. POI 1.0 se distribuyó un mes más tarde de lo planeado, pero con muchas más características. Marc escribió rápidamente el marco del serializador y el Serializador HSSF en tiempo récord y Andy generó más documentación y trabajó en hacer que la gente conociera este proyecto.

Poco antes de la distribución, POI tuvo la fortuna de entrar en contacto con Nicola -Ken- Barrozzi quien proporcionó ejemplos para el Serializador HSSF y ayudó a descrubir sus desafortunados fallos (que fueron arreglados de inmediato). Recientemente, Ken portó la mayoría de la documentación del proyecto POI a XML partiendo de los documentos HTML cutres que Andy había escrito con Star Office.

Más o menos al mismo tiempo de la primera distribución, Glen Stampoultzis se unió al

proyecto. A Glen le molestaba la actitud impertinente de Andy en lo que añadir capacidades gráficas a HSSF se refería. Glen se molestó tanto que decidió coger un martillo y hacerlo él mismo. Glen ya se ha convertido en parte integral de la comunidad de desarrollo de POI; sus contribuciones a HSSF ya han comenzado a producir olas.

En algún momento decidimos finalmente remitir el proyecto a El Proyecto Cocoon de Apache, sólo para descubrir que el proyecto había crecido encajando perfectamente con Cocoon hacía tiempo. Lo que es más, Andy comenzó a ojear otros proyectos a los que le gustaría que se añadiera la funcionalidad de POI. Así que se decidió donar los Serializadores y Generadores a Cocoon, otros componentes de integración con POI a otros proyectos, y los APIs de POI pasarían a formar parte de Jakarta. Fue un camino con baches, ¡pero parece que todo salió bien puesto que ahora estás leyendo esto!

2. ¿Hacia dónde va POI?

Primero abordaremos esto desde el punto de vista del proyecto: Bueno, les hicimos la oferta a Microsoft y Actuate (de coña ... en su mayor parte) de que dejaríamos el proyecto y nos retiraríamos si simplemente nos firmaban a cada uno un cheque con muchos ceros. Todavía estoy esperando una llamada o correo electrónico, así que de momento asumo que no nos van a pagar para quitarnos de en medio.

Después, tenemos algo de trabajo que hacer aquí en Jakarta para terminar de integrar POI en la comunidad. Lo que es más, todavía estamos realizando la transición del Serializador a Cocoon.

HSSF, durante el ciclo 2.0, sufrirá varias optimizaciones. También añadiremos nuevas características como una implementación completa de Fórmulas y formatos de texto personalizados. Esperamos ser capaces de generar ficheros más pequeños añadiendo soporte de escritura para registros RK, MulRK y MulBlank. A día de hoy, la lectura en HSSF no es muy eficiente. Esto se debe sobre todo a que para escribir o modificar, uno necesita ser capaz de actualizar punteros del flujo de subida (upstream pointers) a datos del flujo de bajada. Para hacer esto hay que tener todo lo que haya en medio en memoria. En vez de eso, un Generador permitiría que se procesaran eventos SAX. (Esto se basará en las estructuras de bajo nivel). Una de las mejores cosas sobre esto es que así no sólo tendremos una manera más eficiente de leer el fichero, también tendremos una magnífica forma de utilizar hojas de cálculo como fuentes de datos XML.

El Serializador HSSF, se separará más aún en un marco genérico para la creación de serializadores para otras plataformas y en la implementación específica del serializador HSSF. (Esto ya es cierto en gran medida). También añadiremos soporte para características ya soportadas por HSSF (estilos, fuentes, formatos de texto). Esperamos añadir soporte para fórmulas durante este ciclo.

Estamos empezando a expandir nuestro alcance de nuevo. Si pudimos hacer todo esto para ficheros XLS, ¿qué hay de ficheros Doc o PPT? Pensamos que nuestro siguiente componente (HDF - Formato de Documento Horrible) debería seguir el mismo patrón. Esperamos que se nos una sangre nueva al equipo y que nos permita abordar esto con mayor celeridad (en parte porque POIFS ya está terminado). ¡Pero a lo mejor lo que más necesitamos es a ti!

Copyright (c) @year@ The Apache Software Foundation All rights reserved. $Revision: 1.2 $ $Date: 2003/04/24 00:53:31 $