

Formato de intercambio de información geográfica entre dispositivos embebidos

Oliveros Magaña José Humberto, Valencia Moreno José Manuel, Juan Pablo Torres
Maestría en Tecnologías de la Información y la Comunicación
Facultad de Ciencias Administrativas y Sociales, UABC
holiveros@gmail.com, jova@uabc.edu.mx, pablotorres@uabc.edu.mx

Abstract

En este trabajo se presenta un formato para intercambiar información geográfica entre dispositivos móviles de diversas plataformas y un sistema de información forestal. Tal formato está orientado a su facilidad de uso y atender requerimientos propios de la Confederación Nacional de Organizaciones de Silvicultores de México, en cuanto a desempeño y bajo costo.

1. Introducción

En los sistemas GIS de campo, una de las fases finales es la exportación de la información geográfica a bases de datos, a otros sistemas GIS o planos. Esto es necesario debido a que los dispositivos receptores utilizan formatos internos propietarios para trabajar con eficiencia.

Existen literalmente cientos de formatos en los que es posible transportar información geográfica, sin embargo tan solo unos cuantos son comunes. A pesar de ellos, aun entre los formatos más utilizados se presenta una gran variedad de problemas, como por ejemplo, el formato DXF utilizado en AutoCAD, es popular como formato de intercambio de datos vectoriales (la información geográfica es de tipo vectorial), sin embargo hay 2 principales variantes del mismo, una basada en datos binarios y otra en texto, cada una con variantes para cada versión de AutoCAD, finalmente existen detalles acerca de cómo representar la información en este formato, es decir, ¿en metros, pies, bloques, origen, sistema de coordenadas, atributos? Situaciones similares suceden con el resto de los formatos comunes para intercambio de datos geográficos.

Finalmente las capacidades de importación de datos de los sistemas GIS varían tremendamente, siendo capaces de leer solamente unos cuantos formatos y algunas de sus variantes.

Existe un formato que intenta resolver estos problemas, es el formato de dominio público llamado GPX, el cual está basado en XML textual. Éste formato permite almacenar datos geográficos y multitudes de atributos en una estructura jerárquica relativamente sencilla de leer por sistemas GIS.

Sin embargo existen algunas deficiencias importantes en este formato, como lo es su alto porcentaje de overhead, ineficiencia con altos volúmenes de datos, lentitud de lectura (exceptuando optimizaciones muy específicas que salen del estándar XML) y problemas con dispositivos empujados o no tan modernos como PDAs, receptores GPS, celulares y equipos de cómputo legados.

Existen otras soluciones desarrolladas por las grandes compañías de este giro, cuyos costos económicos y de personal son demasiado elevados, que deja fuera a la mayoría de las pequeñas y medianas empresas sin acceso a las mismas.

Tratar de llevar estas soluciones a precios razonables, hace que la creación y/o mejora de los sistemas SIG sea una tarea altamente compleja, lo que ha llevado entusiastas a la creación de formatos y soluciones alternativas que han tenido poca penetración en el mercado.

Algunos ejemplos de estas soluciones alternativas son GPSBabel, un convertidor estático multiformatos para receptores GPS; PostGIS, un conjunto de extensiones espaciales para el manejador de bases de datos relacionales PostgreSQL; MapServer, un servidor de mapas interactivos, solo por mencionar algunos.

2. La propuesta

Pocas de estas alternativas, han tenido un uso real en la industria de los sistemas SIG, principalmente debido a problemas organizacionales y falta de recursos. En este contexto, la *Confederación Nacional de Organizaciones de Silvicultores* (CONOSIL), debido a sus crecientes necesidades del uso de SIG's

como parte de su *Sistema de Información Forestal*, ha desarrollado un componente particular (de aquí en adelante llamado LIFOR), para el intercambio abierto de información geográfica, orientado a la simplicidad de uso y bajo costo, que puede ser almacenado y utilizado en dispositivos de cómputo de pocos recursos, receptores GPS, computadoras de campo, inclusive en equipo de cómputo obsoleto, y en una versión simplificada, en celulares estándar.

La información que integra LIFOR contempla entre otros aspectos, una plataforma para la comercialización de productos y servicios, indicadores de manejo, así como la posibilidad del registro de inventarios forestales.

LIFOR está implementado de forma tal, que permite su adaptación sin problemas a otras plataformas de cómputo en una multitud de lenguajes (C, Java, JavaScript, .NET, Python).

Proceso de operación de LIFOR



Figura 1. Proceso de operación de LIFOR

La forma en que LIFOR opera es la siguiente:

Recepción. Es el primer paso del funcionamiento de LIFOR. Aquí se recibe una secuencia de datos, la cual puede venir de varios medios, como un disco, un archivo, de una aplicación web, un programa, un receptor GPS, etc.

Decodificación de contenido. La secuencia de datos de entrada puede contener varios formatos de información, los cuales serán detectados de forma automática, eligiéndose el decodificador apropiado de un conjunto de decodificadores disponibles. De manera opcional se puede predeterminedir un sólo formato con el propósito de ahorrar recursos de memoria y espacio, sobre todo cuando se trabaja con dispositivos limitados, como receptores GPS o computadoras de campo.

Conversión. La información es convertida al micro formato interno YAGF al mismo tiempo que es recibida. El micro formato binario YAGF, es un formato simple secuencial diseñado especialmente para utilizarse en medios inestables o lentos como conexiones telefónicas a Internet), con compatibilidad con revisiones más nuevas del mismo.

Codificador. Los datos codificados en YAGF, opcionalmente se pueden convertir a otro formato.

Transmisión. Una vez que los datos se tienen en el formato deseado, se envían a otro destino, como un disco, un servicio Web, que se alimentan en otro programa, una computadora de campo, etc.

El formato YAGF

El formato YAGF es un formato binario simple basado en chunks (Figuras 2), diseñado para ser leído de forma progresiva (streamable), extensible y compatible con futuras versiones del mismo.



Figura 2. Estructura del stream YAGF

Un stream de datos YAGF siempre inicia con un encabezado (Figura 3) de 50 bytes de longitud, seguido de uno o más chunks de datos de tamaño variable, cada uno de los mismos conteniendo un código de secuencia, lo que permite cierta tolerancia a errores en caso de pérdida de segmentos.

El encabezado del stream YAGF

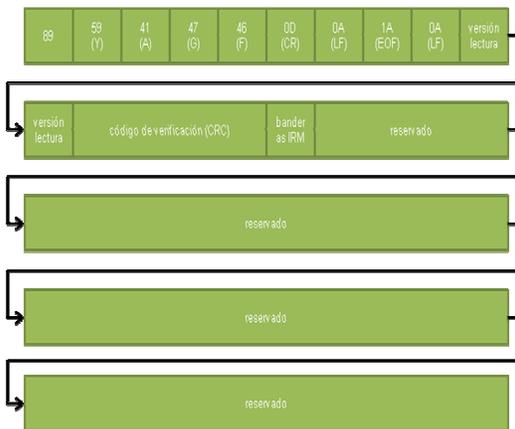


Figura 3. Encabezado del stream YAGF

El encabezado del stream de datos YAGF tiene un tamaño fijo de 50 bytes, con los bits más significativos al principio, es decir, big endian o network order. Éstos están organizados como se muestra en la Tabla 1, donde el byte en la primera posición es el 1:

Bytes	Significado
1-9	Secuencia fija que denota al stream o archivo con formato YAGF. El primer byte, 89 en hexadecimal, es representado en binario como 10001001, el cual al tener el primer bit activo (bit alto), permite detectar si los datos están pasando por un sistema de 7 y no 8 bits. Así mismo, al no ser simétrica su representación, permite detectar si se está usando el bit endian correcto. Los bits del 2 al 5 son la cadena YAGF en código ASCII. Los bits del 6 al 7 (CR, LF) indican un salto de línea de texto en UNIX / Windows, lo que permite detectar si los datos están pasando en forma de texto. El bit 8, EOF, indica fin de archivo en los sistemas de texto populares. El bit 9, LF, indica salto de línea de texto en los sistemas Macintosh.
10	Entero sin signo de 8 bits, indica la última versión de la librería que puede leer el stream sin problemas.
11	Entero sin signo de 8 bits, indica la última versión de

	la librería que puede escribir / modificar el stream sin problemas.
12	Entero sin signo de 32 bits, indica el CRC32 del stream de datos, excluyendo el encabezado.
13	Banderas de bit, indican de forma general el estado IRM (Manejo de Derechos de la Información) del stream de datos, de acuerdo a la Tabla 2.
14 - 50	Área reservada para futuras aplicaciones.

Tabla 1: Campos del encabezado del stream YAGF

Bit	Significado
1	Indica que el stream requiere una contraseña para poder ser leído.
2	Indica que el stream requiere una contraseña para poder ser modificado.
3	Indica que solo los datos de los chunks están encriptados.
4	Indica que el stream completo, excepto el encabezado, está encriptado.
5	Indica que el stream cuenta con listas de control de acceso.
6	Reservada.
7	Reservada.
8	Reservada.

Tabla 2: Banderas IRM

Chunks del stream YAGF

Después del encabezado del stream YAGF, le siguen uno a más chunks de información, es decir, elementos o artefactos de información. Estos chunks pueden ser de diferentes tipos según sea la información a requerir, esto es, pueden ser chunks de puntos (en 2D o 3D), rutas, tablas de datos, relaciones, mapas raster, mapas wavelet, mapas vectoriales, textos (Ascii, UTF 8, UTF 16), anexos binarios, metadatos e IRM.

La estructura que tienen los chunks es como se muestra en la Figura 4, es decir, cada chunk tiene su propio encabezado y los datos en sí. Cada chunk puede aparecer cero o más veces dentro del stream de datos, como sea requerido por la información manejada.



Figura 4. Estructura de un Chunk del stream YAGF.

El encabezado del Chunk

Como se puede observar en la Figura 5, el encabezado local de un chunk tiene cuatro elementos: Tamaño del encabezado, tipo, código de verificación y el tamaño de los datos.



Figura 5. Encabezado local de un chunk

Tamaño del encabezado. Primer byte, es un entero sin de 8 bits, que indica el tamaño del encabezado, teniendo como máximo 255 bytes.

Tipo. Segundo byte, es un entero sin signo de 8 bits, indica el tipo de chunk, véase Tabla 3.

Código de verificación. Byte 3 y es un Entero sin signo de 32 bits, indica el CRC32 de los datos del chunk.

Tamaño de datos. Del byte 4 en adelante, hasta 255. Entero sin signo de ancho variable¹, indica el tamaño de los datos del chunk.

Valor	Tipo
0	Puntos 2D
1	Puntos 3D
2	Rutas
3	Tablas de datos
4	Relaciones
5	Mapas raster
6	Mapas wavelet
7	Mapas vectoriales
8	Textos ASCII
9	Textos UTF 8
10	Texto UTF 16
11	IRM
12	Anexos binarios
12-255	Valores reservados

Tabla 3: Tipos de chunks.

Para los datos en sí, contenidos en los diferentes tipos de chunks se utiliza una estructura similar a XML binario, de acuerdo a las conclusiones del trabajo realizado por el XML Binary Characterization Working Group, adecuada a los requerimientos de este formato.

La mayoría de los chunks cuentan con un elemento link, el propósito principal del mismo, es ofrecer una forma de enlace entre elementos dispares, similar al funcionamiento de los identificadores únicos CID usados en el correo electrónico

3. Resultados

El Sistema Forestal para las Organizaciones de Silvicultores (SFOS) y la propia librería YAGF (Lifor), se encuentran en etapa de pruebas hasta el 20 de octubre de este año, en tres puntos del país:

- Perote, Veracruz (sitio piloto)
- Durango, Durango (instalación de prueba)
- Estado de México, en el Departamento de Manejo de Recursos Forestales, División de Ciencias Forestales, Universidad Autónoma Chapingo.

¹ Un entero de ancho variable consta de uno o más bytes, donde el bit más alto de cada uno indica que se utilizará el siguiente byte y los 7 bits restantes los datos del entero en sí.

Los resultados que se han obtenido se presentan en dos diferentes procesos:

En la captura de datos

Lifor está siendo utilizado para la captura de datos en algunos dispositivos y diferentes sistemas operativos y en otros casos no se han realizado pruebas. Las conclusiones hasta el momento son:

Usando cliente Windows Mobile. La captura de datos en el cliente Windows Mobile se ha realizado sin problemas, puesto que las operaciones de captura no requieren un proceso constante de datos. Debido a que las operaciones de captura son aisladas, no se apreció un cambio significativo en el *retraso* de transmisión ni en el tiempo de proceso de la información.

Usando cliente Windows. La captura de datos en el cliente Windows se ha realizado sin problemas, puesto que las operaciones de captura no requieren un proceso constante de datos. Debido a que las operaciones de captura son aisladas, no se apreció un cambio significativo en el *retraso* de transmisión ni en el tiempo de proceso de la información.

Usando cliente Java ME. El desarrollo para captura de información del cliente Java ME del SFOS fue cancelado debido a que la mayoría de los teléfonos celulares - principales dispositivos con Java ME - no cuentan con las características mínimas para uso práctico de dicho sistema; principalmente la falta de un teclado físico completo.

Usando cliente PalmOS, cliente Linux y cliente MacOS X. El desarrollo para estos sistemas operativos aún no se concluye, por lo que no se han realizado pruebas prácticas de la librería YAGF en dichas plataformas.

En el despliegue de información

Utilizando los mismos dispositivos y sistemas operativos para la captura de información, se han obtenido los siguientes resultados:

Usando cliente Windows Mobile. El uso normal del mismo se da en lugares donde las conexiones a Internet son nulas o a través de Wi-Fi. Al utilizar la librería y formato YAGF, se presentó una reducción en el *retraso* al transferir datos. Asimismo se presentaron menos errores de transmisión puesto que dicha librería permite el *resuming* de los datos; es decir, al haber una falla en la conexión no necesita transmitir de nuevo toda la información. Se encontró que en dicha plataforma el uso de conexiones Wi-Fi o a través de celular es inestable, las conexiones fallan cada cierto tiempo. Se apreció un cambio moderado en el uso de CPU requerido en procesar la información emitida por la librería. Se realizarán de nuevo pruebas

al optimizar la librería para sustituir el uso de funciones de punto flotante a funciones enteras.

Usando cliente Windows. El cliente para Windows de SFOS utiliza una conexión permanente a Internet, puesto que transfiere datos continuamente. El uso normal del mismo se da en lugares donde las conexiones a Internet son lentas o inestables. Al utilizar la librería y formato YAGF, se presentó una reducción en el *retraso* al transferir datos. Asimismo se presentaron menos errores de transmisión puesto que dicha librería permite el *resuming* de los datos; es decir, al haber una falla en la conexión no necesita transmitir de nuevo toda la información. No se apreció un cambio relevante en el uso de CPU requerido en procesar la información emitida por la librería.

Usando cliente Java ME. No se han realizado pruebas ya que el desarrollo del cliente Java ME del SFOS fue cancelado debido a que la mayoría de los teléfonos celulares - principales dispositivos con Java ME - no cuentan con las características mínimas para un uso práctico de dicho sistema; principalmente la falta de un teclado físico completo y la limitada conectividad a otros dispositivos sin utilizar la conexión a la red celular.

Usando cliente PalmOS, cliente Linux y cliente MacOS X. El desarrollo para estos sistemas operativos aún no se concluye, por lo que no se han realizado pruebas prácticas de la librería YAGF en dichas plataformas.

4. Conclusiones

La librería YAGF ha sido implementada en algunas plataformas y se espera que se implemente en otras. A pesar de que aún no concluye el periodo de pruebas de la librería YAGF en el campo, ya ha demostrado ser una herramienta eficaz y eficiente para el intercambio abierto de información geográfica, siendo simple de utilizar y cuyo costo es mucho más bajo que una herramienta adquirida.

Como todo proyecto de TI, siempre hay cosas por mejorar y nuevas funcionalidades por implementar y Lifor no es la excepción. Entre los retos que están pendientes por vencer, está el desarrollo para las plataformas de PalmOs, Linux, MacOS y J2ME, además de incluir algunas funcionalidades más.

5. Bibliografía

American Calcar, Inc. *Positional camera and GPS data interchange device*. Michael L. Obradovich. Estados Unidos de América, patente de invención. 6525768. 2003-02-25

Bundesamt für Kartographie und Geodäsie (Federal Agency for Cartography and Geodesy) (2007). *Networked Transport of RTCM via Internet Protocol (NTRIP)*. Disponible en <http://figs.bkg.bund.de/>

Chuck Gilbert. (1997, enero). GPS Consumer Series: Translating GPS Data to a GIS, CAD System or Database Format, Part III. *Earth Observation Magazine*, 6.

Dominio Público. *GPX: the GPS Exchange Format*. Disponible en <http://www.topografix.com/gpx.asp>

Eric Steven Raymond (2004). *Towards a Better GPS Protocol*. Disponible en <http://gpsd.berlios.de/replacing-nmea.html>

Eric Steven Raymond (2005). *Where's the Latency? Performance analysis of GPSes and GPSD*. Disponible en <http://gpsd.berlios.de/performance.html>

National Marine Electronics Association (2007). *Approved 0183 Manufacturer's Mnemonic Codes*. Disponible en <http://www.nmea.org/pub/2000/NMEA0183MFGCodes.pdf>

National Marine Electronics Association (2002). *NMEA 0183 Standard*. Disponible en <http://www.nmea.org/pub/0183/>

SQLite (2008). *SQLite Database File Format*. Disponible en <http://www.sqlite.org/fileformat.html>

XML Binary Characterization Working Group (2005). *XML Binary Characterization. W3C Working Group Note 31 March 2005*. Disponible en <http://www.w3.org/TR/xbc-characterization/>