

Clientes IDE 3D: Visualización de edificios 3D y SIG móvil

Alberto Romeu¹, Daniel Gastón², Amelia del Rey³

^{1, 2, 3} PRODEVELOP, S.L

aromeu@prodevelop.es; dgaston@prodevelop.es; adelrey@prodevelop.es

Resumen

En los últimos años se ha producido un gran avance en tecnologías y herramientas que permiten generar, visualizar y analizar contenido geográfico.

Estas nuevas tecnologías, formatos y globos virtuales 3D están dando un gran impulso a la investigación en áreas como la planificación urbanística, y van abriendo el camino para un nuevo entorno de estandarización que permita establecer un contexto real de interoperabilidad.

Algunos proyectos de investigación, como SOSTRE y Glob3 Mobile, son claros ejemplos de aplicación de todo este conglomerado de tecnologías, y dan pistas del camino a seguir a la hora de consumir contenido 3D, tanto en web, como en entornos móviles multiplataforma.

Palabras clave: IDE, 3D, WebGL, open source, SIG 3D, SIG 3D móvil

1 Introducción

El uso generalizado de Internet en nuestra vida cotidiana, no solo está influenciando la forma en que las actuales tecnologías espaciales están siendo desarrolladas, sino que además está propiciando la aparición de nuevas aplicaciones. Muchas de esas aplicaciones aportan un valor añadido tan notable, que hacen que rápidamente dichas tecnologías geoespaciales se estén convirtiendo en esenciales, integrándose en las actividades diarias tanto de profesionales como de particulares. Posiblemente uno de los ejemplos más claros, es el caso del manejo de los datos procedentes de las Infraestructuras de Datos Espaciales (IDE) a través de sus Geoportales.

Los Geoportales al permitir la visualización, consulta, utilización del nomenclátor y localización de datos geoespaciales han propiciado que los mapas actuales se hayan convertido en un concepto interactivo en donde, tanto usuarios genéricos como profesionales, pueden explorarlos dinámicamente pudiendo resultar especialmente útiles en la toma de decisiones.

Es por ello, que hay una clara convicción de continuar con el desarrollo de soluciones similares que además de aportar el valor 2D de la información geospacial, aporten el plus añadido de la tercera dimensión, necesaria tanto en geoanálisis 3D como en la representación de los resultados de los mismos.

La capacidad de importar contenido 3D y representarlo gráficamente de forma nativa en aplicaciones Web en general y Geoportales en el caso de las IDEs, es probablemente, el cambio más radical que la Web va a adoptar en mucho tiempo [1].

2 Estado del arte de formatos, globos virtuales 3D y estándares OGC 3D

2.1 Formatos 3D y estándares para Clientes Web

El hecho de poder representar contenido 3D en un entorno Web no es algo novedoso, puesto que ya era posible hacerlo en el pasado [2] con APIs y formatos tales como:

- Flash
- O3D
- VRML

La característica común de todos estos formatos, es la necesidad de instalación de *plugins* de terceros con la consabida problemática de incompatibilidad entre aplicaciones, navegadores y dichos *plugins*. Sin embargo, la aparición y paulatina integración de la API WebGL en los navegadores actuales, ha simplificado y mejorado en todos los aspectos la visualización de contenido 3D hasta un punto inimaginable hace unos años.

WebGL es una API multiplataforma para JavaScript que permite representar gráficos 3D acelerados por hardware (GPU), en tiempo real, en los navegadores actuales sin la necesidad de *plugin* alguno. Internamente se basa en la implementación nativa de OpenGL ES 2.0 con ligeros cambios en un pequeño conjunto de instrucciones, quedando vinculada al elemento *canvas* presente en el DOM de HTML5. De este modo no solo se consigue una total compatibilidad entre plataformas y aplicaciones, sino que además se mejora el rendimiento gráfico de las mismas.

Adicionalmente, cabe destacar otros formatos 3D y estándares más en uso hoy en día como son:

- CityGML
- KML/COLLADA
- X3D

CityGML es un formato definido en XML, concebido para almacenar, representar e intercambiar modelos virtuales 3D de entornos urbanos. Proporciona una forma de

describir objetos considerando su geometría, topología, semántica y apariencia, todo ello interrelacionado con distintos niveles de detalle.

KML es un formato también definido en XML, concebido para almacenar y representar datos geográficos en 3D. Definido por y para la Web, da soporte a la representación de imágenes, regiones, polígonos o POIs entre otras cosas; y en conjunción con COLLADA, también modelados 3D como edificios.

A pesar de que ambos formatos (CityGML y KML/COLLADA) son ampliamente usados y aceptados, X3D es el que requiere una mención especial en relación al propósito de este artículo

X3D ha evolucionado del mencionado formato VRML hasta convertirse en un formato más definido y usado. X3D, ratificado como estándar ISO, proporciona un sistema para el almacenamiento, la recuperación y posterior reproducción en tiempo real, de gráficos 3D en una amplia gama de aplicaciones.

Definido en XML, cuenta con cuatro perfiles troncales y varios adicionales, en donde se definen las características que todo sistema compatible con X3D debe apoyar. Todos ellos en conjunto no solo dan soporte a la necesaria componente geoespacial sino también a la definición de las correspondientes primitivas geométricas y sus niveles de detalle asociados entre otras cosas.

X3D define además un DOM particular, una serie de eventos predefinidos, capacidad de *scripting*, soporte audio, video y sensores externos, *shaders*, animaciones o cálculos de físicas. Por estos motivos, es uno de los formatos a tener más en cuenta en la construcción de mundos virtuales de los existentes a día de hoy [3].

Formato	Lenguaje	Estándar (entidad) Uso	En uso	Enfoque Web	Soporte Geo
Flash	Propio	NO	-	X	-
O3D	Propio	NO	-	X	-
VRML	Propio	NO	-	X	-
CityGML	XML	SI (OGC) – representación objetos urbanos 3D	X	-	X
KML/COLLADA	XML/XML	SI (OGC) – visualización geográfica / SI (Khronos) – definición entidades 3D	X	X / -	X
X3D	XML	SI (ISO) – definición entidades 3D en tiempo real	X	X	X

Tabla 1. Comparativa de formatos de datos geoespaciales 3D

2.2 Clientes Web 3D: Globos virtuales en WebGL

La gran mayoría de los proyectos de globos virtuales Web que están siendo desarrollados se han percatado del potencial que WebGL brinda tanto a desarrolladores como a usuarios finales en términos no sólo de representación 2D, sino también 3D.

En este punto cabe dividir las soluciones existentes en dos grupos bien diferenciados: SDK para representación simple de datos geográficos (ej. temperaturas); y SDK para representación compleja no solo de datos sino de cualquier tipo de geometría 3D (ej. edificios).

Empezando por los SDK de representación compleja, desde el punto de vista de código abierto nos encontramos globos virtuales tales como:

- Cesium
- OpenWebGlobe
- ReadyMap
- WebGLEarth

Desde la de código fuente propietario destacan:

- Nokia Maps 3D
- Google Maps WebGL

Todos estos globos virtuales podrían resultar extremadamente útiles al considerarse una posible plataforma a desarrollar en la que poder mostrar diversa información geográfica en 3D.

En lo que respecta a los SDK para representación simple de datos, cabe destacar a:

- PhiloGL
- Chrome Experiments WebGL Globe

Ambos dan funcionalidad a la representación de tuplas de datos (coordenadas – valor) sobre la superficie de un globo virtual 3D en WebGL. Ésta tipología de globo queda especialmente indicada para el caso de visualización de variables o eventos puntuales georreferenciados, así pues, se pueden implementar visualizaciones de distinta índole como cambios de temperatura en la Tierra, población por países, resultado de elecciones por regiones o rutas de vuelos internacionales entre otros casos.

Globo virtual 3D	Tipo Representación	Open source	Licencia OS	Soporte espacial	Iniciativa de estandarización /formato
Cesium	Compleja	X	Apache 2.0	X	Si / CZML
OpenWebGlobe	Compleja	X	MIT	X	No
ReadyMap	Compleja	X	GNU LGPL v3	X	No
WebGLEarth	Compleja	X	GNU GPL v3	X	No
Nokia Maps 3D	Compleja	-	Nokia ToS	X	No
G. Maps WebGL	Compleja	-	Google ToS	X	No
Philo GL	Simple	X	MIT	X	No
G. Exprim. globe	Simple	X	Google ToS	X	No

Tabla 2. Comparativa de globos virtuales 3D

2.3 Estándares OGC 3D

En lo relativo a formatos y estándares OGC, las aplicaciones geoespaciales basadas en clientes 2D ya han resuelto mayormente los problemas de interoperabilidad. A este respecto, en la actualidad los estándares OGC WMS y TMS ya son ampliamente aceptados para el intercambio de información geográfica en formato ráster, mientras que otros estándares OGC, como WFS, permiten el intercambio de información vectorial.

No obstante, en el caso de las soluciones basadas en clientes geoespaciales 3D, se observa la falta de consenso al adoptar estándares que faciliten la interoperabilidad entre las distintas aplicaciones.

Mientras prosiguen algunos intentos de resolver ese vacío, como es el estándar OGC W3DS (aún en borrador) [4], el hecho es que los proveedores de aplicaciones Web ya se están decantando por múltiples formatos no estandarizados para describir la topología y semántica asociada a sus escenas 3D.

Por ejemplo, en el caso del proyecto Cesium, se ha diseñado un formato CZML personalizado con la idea de proponerlo como estándar ante alguna entidad. Por otra parte, el resto de globos virtuales mencionados proponen sus correspondientes formatos basados en el formato JSON.

3 Proyectos de Investigación relacionados

Durante los últimos cuatro años PRODEVELOP ha abierto una línea de investigación en proyectos para la mejora en la visualización y gestión de datos geoespaciales 3D a través de su participación en los siguientes proyectos de I+D+i:

3.1 SOSTRE: TRatamiento SOSTenible en la Edificación

SOSTRE tuvo como objetivo fundamental la investigación en modelos precisos de cálculo de, primeramente, las necesidades de demanda energética de edificios; y seguidamente, del cálculo del impacto paisajístico derivado de su localización. Se ha basado tanto en tecnologías de modelización y diseño de edificios, como de análisis 3D del emplazamiento de edificios en el terreno [5].

Todo ello permite disponer de consumos estimados de energía y emisiones a la atmósfera de gases con efecto invernadero, con objeto de minimizar el consumo energético desde un punto de vista pasivo; así como obtener con precisión la visibilidad relativa del edificio para poder obtener estudios de impacto paisajísticos.

En lo relativo a la arquitectura del cliente IDE, se ha optado por un sistema de *cliente pesado* donde la mayor carga de cómputo está desplazada hacia la computadora que ejecuta la aplicación Web.

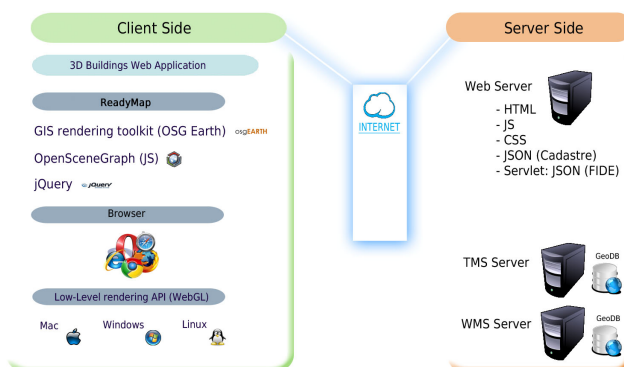


Figura 1. Arquitectura de SOSTRE

Empezando en el lado del cliente, y centrándonos en la representación de edificios, se ha optado por ReadyMap como solución multiplataforma basada en WebGL. ReadyMap es una SDK que nos da la funcionalidad de visualizar un globo virtual WebGL sobre un navegador genérico. Es la base perfecta sobre la cual desarrollar nuestra representación de edificios con sus correspondientes cálculos.

En el lado del servidor, podemos diferenciar tres tipos de servidores. El servidor Web, nos provee con los ficheros necesarios que en última instancia se ejecutarán en el cliente como son ficheros HTML, JavaScript, hojas de estilos y los ficheros JSON en donde se encuentran definidas las geometrías de los edificios como veremos más adelante.

Adicionalmente se han configurado tanto servicios WMS como TMS para acceder a los correspondientes servidores y traerse las teselas de mapa necesarias para cubrir la superficie del globo.

Centrándonos en los datos necesarios para la representación de los edificios, se han tenido en cuenta dos fuentes:

- Ficheros SHP de catastro (capas MASA y CONSTRU)
- Ficheros FIDE

Empezando por los datos de catastro, se han usado dos ficheros SHP distintos para la ciudad de Valencia. Ambos ficheros en conjunto nos proporcionan el perfil de la base del edificio y un atributo de altura, elementos necesarios para poder extruir el edificio en la aplicación Web.

Los archivos SHP se exportan a continuación a GeoJSON y posteriormente se cargan en una aplicación Java diseñada específicamente para eliminar redundancia de puntos en las geometrías. Posteriormente, estas geometrías y las semánticas asociadas a cada edificio se guardan en formato JSON de forma ligera y entendible para nuestra aplicación Web, consiguiendo por tanto una mayor velocidad [6]

de procesamiento respecto con el formato XML, ampliamente usado en Internet también.

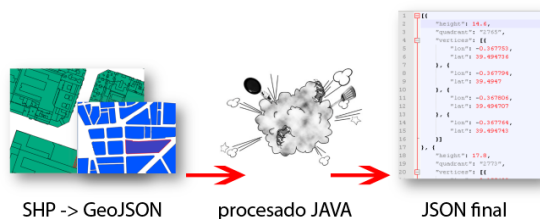


Figura 2. Pre-procesado de ficheros SHP de Catastro para consumir en SOSTRE

FIDE es un formato de intercambio de datos en la edificación definido en XML y usado ampliamente en la Comunidad Valenciana. El objetivo a alcanzar con los ficheros FIDE es el mismo que con el caso de los edificios del catastro: conseguir describir las geometrías del edificio en un fichero JSON de forma ligera y comprensible; sin embargo, el procesamiento interno en Java varía notablemente. En vez de eliminar puntos redundantes, lo que se ha necesitado es describir la geometría del edificio con referencia a un único sistema de referencia local, para luego pasarlo a uno geográfico y poder representarlo así en la superficie del globo.

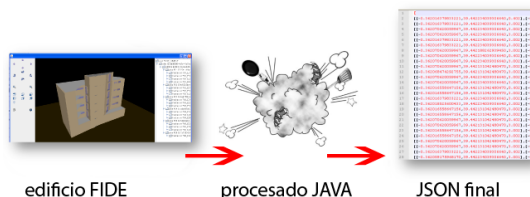


Figura 3. Pre-procesado de ficheros FIDE para consumir en SOSTRE

Tanto en el caso de los edificios procedentes del catastro, como el caso de los edificios FIDE, los ficheros JSON finales que se cargarán en la aplicación Web, se han definido y descrito siguiendo estrictamente las necesidades de la aplicación ante la inexistencia de ningún estándar en la materia.

De este modo, y para el caso de datos procedente de catastro, transferimos las coordenadas geográficas de los vértices que componen la base 2D del edificio, y un atributo altura común a todos ellos. Ya una vez dentro de la aplicación, extruimos el edificio usando ese atributo de altura, obteniendo el edificio en 3D. Por último, los datos del edificio extruido se procesan y se renderizan en la aplicación, de este modo se obtiene gráficamente la geoposición de los edificios en la superficie del globo.

Para los datos procedentes de ficheros FIDE, el método a seguir es similar al caso del catastro, con la salvedad de que en este caso, podemos extraer las coordenadas 3D del edificio. Así pues, no se necesita extruir el edificio y se puede pasar a la fase de renderización directamente.

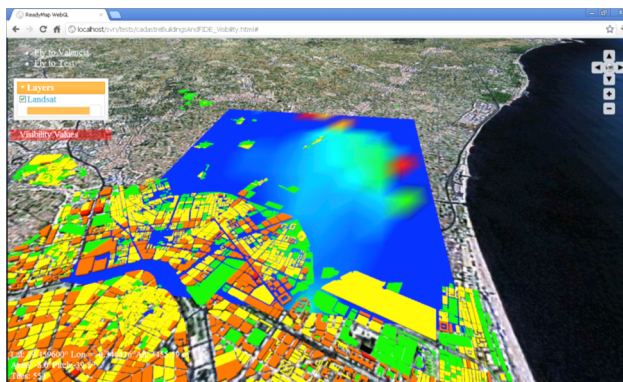


Figura 4 SOSTRE. Visualización de cuenca de visibilidad

3.2 Glob3 Mobile: Globo Virtual 3D para dispositivos móviles basado en un interfaz universal

La rápida evolución en las especificaciones de dispositivos móviles ha dado un extraordinario impulso a las aplicaciones y servicios que sobre ellos se desarrollan. El desarrollo de aplicaciones SIG para dispositivos móviles abre además, nuevos caminos a la explotación de las IDEs, de manera que se pueden explorar nuevas vías de visualización de información geográfica, como puede ser el 3D o la realidad aumentada, así como nuevos mecanismos de consumo de esta información por parte del usuario final, proporcionándole información útil a su localización, hábitos, contexto, etc.

El mercado de los dispositivos móviles se presenta como una plataforma idónea para la investigación en SIG 3D y así es como nace Glob3 Mobile [7].

El objetivo del proyecto es el desarrollo de un innovador sistema de visualización de escenas 3D a través de globos virtuales, mediante la generación de un sistema abierto *Open Source*, implementando el acceso a servicios de información 2D y 3D a través de estándares, y asegurando la accesibilidad desde las distintas plataformas de dispositivos móviles existentes.

Este último constituye especialmente su elemento diferenciador, el funcionamiento en tres plataformas iOS, Android y Web. Además de servir como plataforma para el desarrollo de aplicaciones SIG 3D sectoriales, proporcionando una API sencilla y fácil de extender.

El innovador ciclo de desarrollo de Glob3 Mobile se resume en el siguiente diagrama:

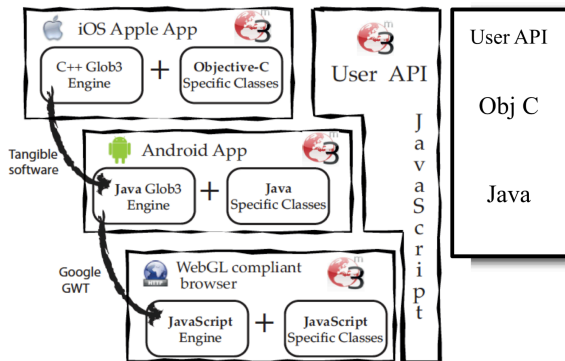


Figura 5. Ciclo de desarrollo de Globe3 mobile.

La solución elegida ha sido la siguiente: se ha desarrollado inicialmente todo el motor en C++ estándar. Este motor usará en todo momento clases virtuales para las funciones básicas (acceso al disco, acceso a la red, captura de eventos, trabajo con la librería gráfica), de tal forma que luego, para cada proyecto específico, se realizarán implementaciones de todas estas clases en el lenguaje correspondiente (Objective-C para las clases específicas en iOS, Java para Android y JavaScript para web).

De esta forma, combinando el motor C++ con las clases específicas en Objective-C obtenemos la aplicación nativa para dispositivos iOS.

A continuación, con este motor escrito íntegramente en C++, se utiliza una herramienta software de conversión, que es capaz de convertir código escrito en C++ a código escrito en Java. Con este motor convertido, junto con las clases específicas en Java, se obtiene la aplicación nativa para dispositivos Android.

Finalmente, con el motor escrito en Java, se usa la tecnología GWT de Google para convertir a código JavaScript. Este código, junto con las clases específicas en JavaScript permite obtener la aplicación para navegadores.

Quedando la arquitectura de capas de Glob3 Mobile definida con el siguiente diagrama:

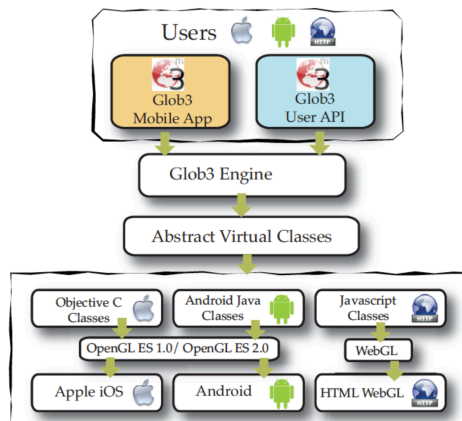


Figura 6. Arquitectura de Glob3 Mobile.

La arquitectura de capas muestra claramente un motor que corresponde al globo virtual 3D escrito en C++. Este depende de clases abstractas con sus respectivas implementaciones para cada una de las plataformas a las que se da soporte. Por último la capa superior se corresponde con el API de usuario que le permite a un desarrollador crear aplicaciones verticales para la plataforma que elija.

Dentro de este contexto, se están realizando avances en el soporte de diferentes orígenes de datos tanto 2D como especialmente 3D en el núcleo de Globe3. Esto implica, escribir código que sea compatible con la arquitectura planteada teniendo especial cuidado a la hora de añadir dependencias con librerías de terceros que puedan “romper” el ciclo de desarrollo.

Para el soporte de formatos 2D se han barajado hasta el momento dos posibilidades:

1. Dar soporte a varios formatos:

La tendencia habitual en muchos clientes GIS es dar soporte al mayor número posible de formatos, para facilitar la vida al usuario. En general, se puede conseguir dando soporte a alguna librería de acceso a datos como GDAL/OGR en C++ o GeoTools en Java.

Se han realizado pruebas para dar soporte a GDAL/OGR en Glob3 Mobile llegando a las siguientes conclusiones:

- a. No se ajusta al ciclo de desarrollo. El código de OGR no es compatible con el conversor de código que se utilizar para derivar el código Java.
- b. Esta dificultad se puede superar ya que Android permite utilizar código en C++ a través de su NDK. La manera de conseguirlo es obteniendo *bindings* Java que conectan con el código en C de GDAL/OGR a través de alguna herramienta como SWIG. De esta manera, se puede acceder a las librerías de GDAL/OGR compiladas para Android a través de clases Java y conseguiríamos tener código compatible con la versión para iOS y Android, pero quedaría fuera la versión para web.

2. Especializarse en un único formato:

A las dificultades técnicas de dar soporte a diversos formatos, podemos sumar algunas ventajas que supone especializarse en un único formato:

- a. En el mundo de desarrollo para dispositivos móviles hay una palabra clave: *optimizar*. En muchas ocasiones, la mejor opción es contar con el mejor soporte a un tipo determinado de datos que favorezca la eficiencia en la carga y optimización de recursos del dispositivo (batería, capacidad de procesamiento, memoria...)
- b. Dar soporte a un sólo formato puede simplificar el proceso de generación de código.

Ahora bien, a la hora de decidir un formato adecuado, se presentan de nuevo, alternativas. Por una parte, existen librerías de terceros como libKML que dan soporte completo al formato KML que puede resultar de especial interés. Tras los

tests realizados, se llega de nuevo a la conclusión de que la librería no es compatible con el ciclo de desarrollo.

Por tanto, parece razonable que la mejor opción sea dar soporte de manera nativa a algún formato de uso común, en este caso, GeoJSON, al que la mayoría de servidores de mapas dan soporte y está ampliamente utilizado en entornos web. Además, su complejidad es relativamente baja comparada con otros formatos y en todo caso, existe la posibilidad de extender la especificación para ajustarla a necesidades futuras.

La siguiente tabla resume el análisis de soporte de formatos 2D para Globe3 Mobile

Formatos	Librería	Plataformas	Complejidad	Modelos 3D
Todos	GDAL/OGR	iOS + Android NDK	Alta	No
KML	libKML	iOS + Android NDK	Alta	Collada
GeoJSON	Nativa	Todas	Baja	Extensión

Tabla 3. Análisis de soporte de formatos 2D para Glob3 Mobile.

Este análisis aclara el camino a seguir a la hora de plantear el soporte de orígenes de datos 2D en un cliente SIG 3D multiplataforma y da pistas del camino a seguir a la hora de dar soporte a orígenes de datos 3D. Las claves son elegir formatos sencillos y ligeros, ampliamente utilizados y con capacidad para ser extendidos en función de las necesidades.

4 Conclusiones

En los últimos años se ha producido un gran avance en tecnologías y herramientas que permiten generar, visualizar y analizar contenido geográfico 3D y se están dando los primeros pasos para crear estándares que permitan el consumo de información y la interoperabilidad.

En este artículo hemos tratado de dar una visión general al estado del arte actual en lo que se refiere a tecnologías 3D y formatos, orientados principalmente a la Web. Además, hemos mostrado dos líneas de investigación en lo que se refiere a clientes SIG 3D.

Por una parte SOSTRE, un cliente Web orientado a la planificación urbanística capaz de consumir datos de Catastro y servicios WMS.

Por otra parte, Glob3 Mobile, un globo virtual 3D multiplataforma que pretende servir de plataforma para el desarrollo de aplicaciones SIG sectoriales.

Se ha explicado parte de la problemática actual en el desarrollo de clientes SIG 3D. En entorno Web, la industria está evolucionando en el sentido de crear globos virtuales 3D utilizando WebGL y formatos de intercambio de datos que se ajustan a necesidades concretas. En entorno móvil se está investigando en la creación de entornos multiplataforma, siendo las dos plataformas móviles iOS y Android las mejor posicionadas. En estos entornos la clave consiste en optimizar el consumo de recursos de los dispositivos.

Dicho esto, todavía queda un largo camino que recorrer en lo que se refiere a servicios y protocolos de intercambio de información geográfica 3D. Todavía no existen estándares, como sí los hay en 2D (WMS, TMS, WFS...) que faciliten la publicación de datos 3D y la interoperabilidad, características fundamentales en cualquier IDE.

Recientemente el *Open Geospatial Consortium* ha publicado un *paper* [3] que puede aclarar el futuro cercano de interoperabilidad en SIG 3D. El *paper* muestra los resultados de un experimento en lo que se refiere a interoperabilidad de sus actuales candidatos a estándar 3D: W3DS (Web 3D Service), WVS (Web View Service) y 3D SE (3D Symbology Encoding).

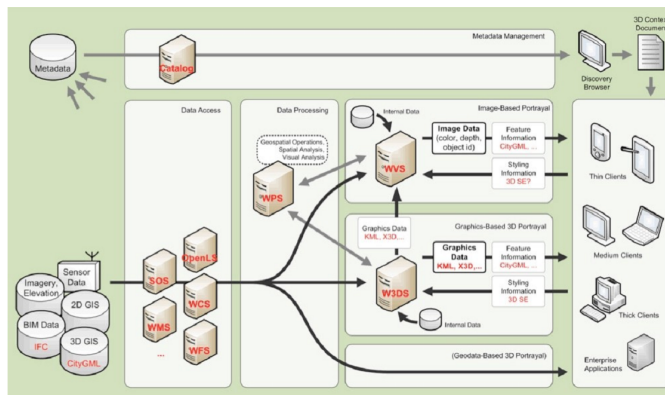


Figura 7. Arquitectura de experimento de representación de contenido 3D e interoperabilidad realizado por OGC.

El experimento ha sido diseñado para probar las distintas implementaciones de referencia de estándares 3D, centrándose en la carga de datos, su publicación y representación por parte de clientes Web y móvil. Del experimento se desprenden las siguientes conclusiones:

1. Respecto a formatos, se desprende que la opción mejor soportada actualmente, tanto por editores GIS como editores 3D es la combinación KML/Collada. El formato X3D que cuenta con una extensión geoespacial para definir contenido geográfico no está soportado por la mayoría de editores.
2. Respecto al estándar WVS que se basa en la renderización de escenarios 3D del lado de servidor para servir imágenes al cliente. Se ha observado la necesidad de optimización en el renderizado de modelos de ciudades grandes. Para ello será necesario utilizar técnicas de cacheado, renderización paralela o utilización masiva de hardware (RAM, discos SSD, memoria en GPU, etc.)
3. Respecto al estándar W3DS (que equivaldría a WFS en 2D) se recomienda el uso de peticiones teseladas.

En el futuro cercano se prevé la consolidación de estándares 3D y por consiguiente, parte de la industria evolucionará creando implementaciones que

favorecerán el desarrollo de IDEs 3D. Pero para llegar a este punto, todavía queda camino por recorrer.

5 Referencias bibliográficas

- [1] Bochicchio, M., Longo, A., & Vaira, L. (2011). Extending Web Applications with 3D Features. 13th IEEE International Symposium on Web Systems Evolution (WSE), (pp. 93-96).
- [2] Ortiz, S. (2010). Is 3D Finally Ready for the Web? Computer , 43, 14-16.
- [3] OGC (2012), OGC completes 3D Portrayal Interoperability Experiment. <http://www.opengeospatial.org/pressroom/pressreleases/1693> (último acceso 11 de Octubre de 2012)
- [4] OGC (2010), Draft for Candidate OpenGIS Web 3D Service Interface standard. v0.4.0
- [5] Prodevelop (2011) SOSTRE project. <https://confluence.prodevelop.es/pages/viewpage.action?pageId=16318495>, (último acceso 11 de Octubre de 2012)
- [6] Wang, G. (2011). Improving Data Transmission in Web Applications via the Translation between XML and JSON. Third International Conference on Communications and Mobile Computing, (pp. 1-4).
- [7] Prodevelop (2011) GIS 3D Móvil. <https://confluence.prodevelop.es/display/GIS3DMOV/Home>, (último acceso 11 de Octubre de 2012)