

# Un Modelo de Generación de Código en Flash para InTml

Christian O. Benavides, Pablo Figueroa

Grupo IMAGINE, Universidad de los Andes, Bogota, Colombia  
{co.benavides86,pfiguero}@uniandes.edu.co

**Abstract.** InTml is a language for Mixed Reality Applications, independent of particular implementation technologies. This paper adds another method of execution for InTml Applications, in this case on top of Flash. We describe a way to translate InTml applications into a Flash execution environment, with code generation in ActionScript 3 and a basic framework based on native 3D engines and a basic layer of execution. We present implementation details and a comparison of latency between this implementation and a previous solution in Java for a simple interactive application.

**Key Word :** InTml, Actionscript, 3D Nativo, RIA.

## 1 Introducción

Con el auge y demanda de información 3D para la Web, han aparecido tecnologías capaces de manipularla y entenderla, una de ellas y la mas populariza es Flash [1] tecnología actualmente escrita y distribuida por Adobe Systems [2] que a través de su visualizador Flash Player 10 casi tan común en cualquier plataforma lo hacen ideal para estas tareas.

Es tanto así que para esta tecnología ya se encuentra motores de desarrollo 3D entre los que se pueden nombrar: PaperVision 3D [3], Away3D [3], Alternativa 3D [3] y entre muchos otros que han recorrido el mundo de la Internet hace algunos años y actualmente han sido base de proyectos industriales y de alternativas visualización.

Todos estos tipos de motores se caracterizan por su gran abstracción del lenguaje nativo ActionScript 3 [1] para lograr presentar información multimedia 3D, campo que ya programas comerciales de diseño 3D como Swift.3D [4] implementan recreando sus propios generadores de código y un modelo de interacción basado en estos motores.

Pero este tipo de desarrollos son aun muy débiles al momento de interactuar con ellos cuando ya se habla de dispositivos diferentes a un mouse, un teclado o una cámara Web.

En consecuencia a estas desventajas han aparecido servidores de intercambio de datos como WiiFlash [5] una plataforma servidor escrita en C++ limitada a la transferencia de información con el dispositivo Wiimote, condición poco atractiva para otros desarrollos sin olvidar su forma poco convencional de programación.

Esto nos lleva a requerir de un modelo que basado en el lenguaje genérico de Flash [1] permita desarrollar aplicaciones multimedia 3D con soporte en intercambio de información entre dispositivos y de una manera fácil de programar.

Por lo tanto basados en las características del modelo planteado es que nos aproximamos al concepto de InTml [6] ya que representa un framework de generación de código para Java y C++ que permite definir modularmente la forma de interacción con información 3D permitiendo a diferentes participantes aprovechar de múltiples dispositivos, de tal forma que lo hace útil para este fin.

De esta forma y gracias al concepto de InTml [6] es que se presenta alrededor de este artículo un esquema de implementación para Flash basado en este tipo particular de tecnología y se mostraran los diferentes estudios de latencia con respecto a la plataforma base InTml [6].

## **2 Arquitectura de Desarrollo InTml para Flash**

La manera de implementar el objetivo planteado se lo puede ver en la figura 1, donde se observa un servidor diseñado en Java que encapsula funcionalidades capaces de interactuar con dispositivos externos y proveen de datos a aplicaciones tipo FlashPlayer generadas bajo InTml.

Consecuentemente las aplicaciones ya generadas bajo InTml son mostradas para ser ejecutadas en un navegador Web compatible con FlashPlayer.

### **2.1 Servidor VRPN-XMLSocket**

Una de las maneras de interactuar información con dispositivos como un Gamepad, Wii, Flymouse, etc., es a través de un servidor VRPN [7] desarrollado para generar una interfaz de red transparente entre una aplicación y un conjunto de dispositivos. Actualmente esta implementado para Java y C++.

Este tipo de implementaciones nos plantean combinar los servicios VRPN [7] con un forma de comunicación cliente-servidor a baja latencia para transportar los datos de interacción hacia flash.

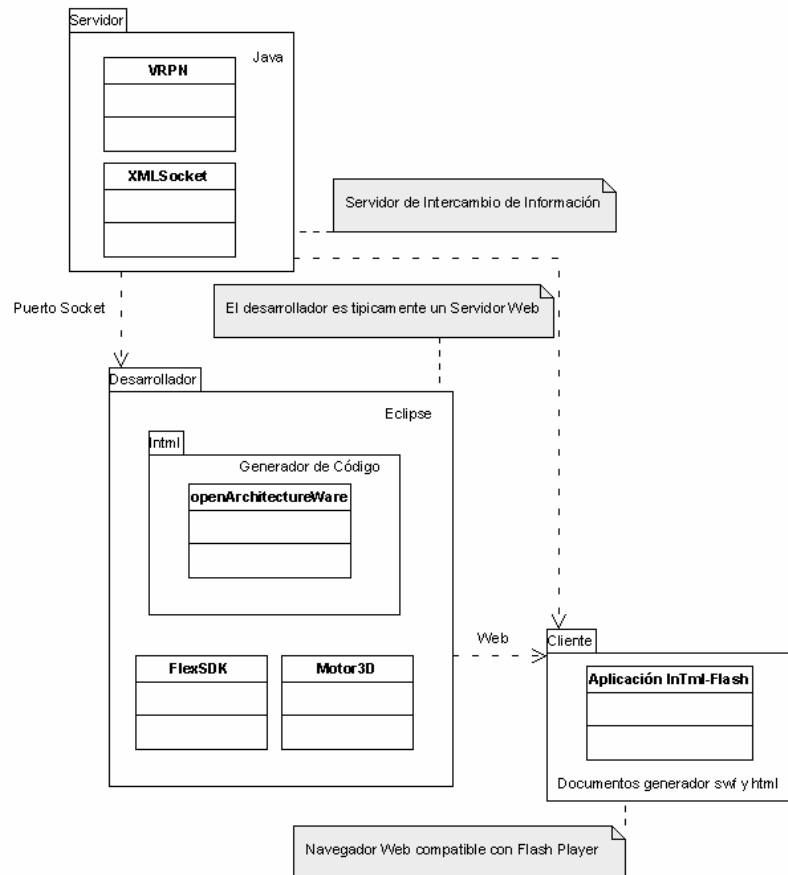
En consecuencia se ha optado por utilizar XMLSocket [8] un tipo de objeto cliente para la tecnología Flash por el cual se envía y se recibe datos en formato XML con el servidor en cualquier momento y dirección.

De esta manera el servidor planteado y teniendo en cuenta las características de comunicación se desarrollo en Java un XMLSocket Server [8] que envuelve los datos provenientes de VRPN [7] y los encapsula en un XML para ser leídos por el cliente Flash que los convertirá en un evento de dispositivo.

Una de las primeras ventajas que aparece es la capacidad de no interactuar con un tipo determinado de dispositivo dando flexibilidad para diferentes tipos de implementaciones.

## 2.2 InTml

El framework de InTml [6] está basado en un plug-in para Eclipse sobre la plataforma del desarrollador, permitiendo definir modularmente la forma de interacción con la información 3D, por ejemplo como se observa en la figura 2



**Fig. 1.** Arquitectura Cliente-Servidor para una aplicación InTml-Flash

Cada módulo generado representa un comportamiento, un objeto o un dispositivo, haciendo de su programación altamente desacoplada y muy bien definida.

El paso de información producida en un tipo particular de módulo se almacena en un puerto de salida que está conectado con un puerto de entrada en otro módulo.

Todo esto nos genera una nueva ventaja ya que se cuenta con entidades independientes fáciles de entender y de las cuales nos preocupa únicamente los datos a devolver.

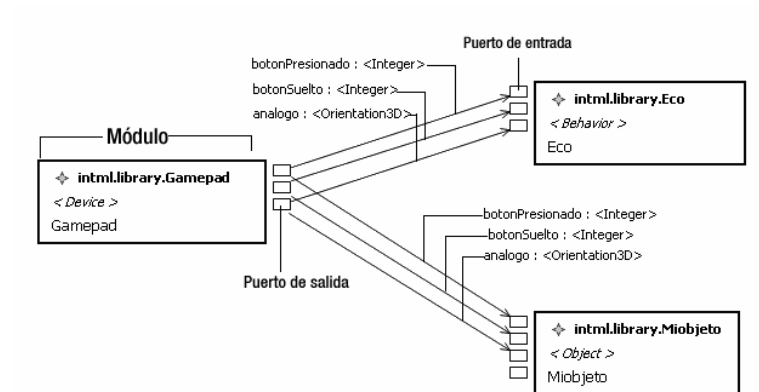


Fig. 2. Definición modular de objetos en InTml [6]

Al tener definido gráficamente el modelo de interacción, se lleva a cabo la generación de código producida en InTml por openArchitectureWare [9] el cual contiene lenguajes de transformación de modelos que a través de plantillas generadoras producen la definición de código, de tal forma que se añadió adicionalmente a InTml [6] a través de openArchitectureWare las plantillas de generación de código al lenguaje ActionScript 3 propio de Flash [1].

Las características fundamentales de la traducción de código a ActionScript 3 esta basada en la utilización de Sprites para cada clase independiente de tal manera que logra embeberse en el marco de ejecución de Flash Player, además se tiene en cuenta los eventos Event.ENTER\_FRAME de los Sprites que producen los escuchas constantes sobre los puertos, básicos del entendimiento de InTml.

### 2.3 Motores 3D y Flex SDK

Ya con una generación de código InTml para Actionscript 3 se hace necesario un motor 3D para la generación de aplicaciones multimedia ricas en este tipo de contenido.

De tal forma que para motivos de este proyecto se utilizo PaperVision 3D Great White [10] debido a su versatilidad, fácil manejo con interacciones y su fácil integración con objetos de diseño grafico dando un formato de visualización mucho mas atractivo para los usuarios en la Web.

La última fase del desarrollo es tomada por el compilador de Flash, que para nuestro caso particular fue Flex SDK 3.0.3 [2] debido a su potencial de generación a partir de una clase ActionScript 3 y adicionalmente con librerías particulares se produce los archivos base .swf y .html de ejecución bajo el cliente.

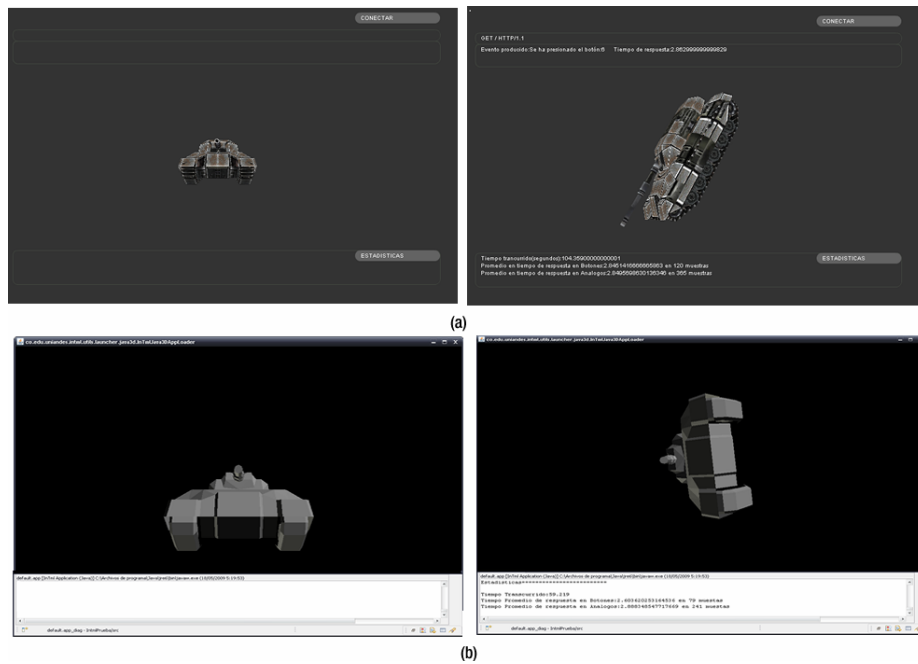
### 3 Pruebas

Una de las preocupaciones generadas alrededor de este proyecto es verificar la latencia presente entre el intercambio de datos de un dispositivo y la aplicación final del cliente debido a que hay más conexiones que en una aplicación InTml y además estas aplicaciones están muy relacionadas con los recursos limitados del navegador Web.

Para esto se tomaron 4 muestras bajo diferentes equipos de hardware y software con compatibilidad con Flash Player 10 y Java 1.6.

La prueba consistió en ejecutar una aplicación InTml, la cual se muestra en la figura 3 en la que se puede observar la carga de un objeto 3D, que para el caso de la aplicación InTml-Flash (figura 3a) es un archivo tipo Collada3D y para la aplicación InTml-Java (figura 3b) es un archivo tipo WRL.

Consecuentemente ya con las aplicaciones compiladas son cargadas bajo el navegador Web para InTml-Flash (figura 3a) y bajo el compilador de eclipse para InTml-Java (figura 3b), todo esto durante un tiempo determinado, de tal manera que se pueda observar el promedio en tiempo de intercambio de datos entre un dispositivo y la aplicación final al mover un objeto 3D en el espacio.



**Fig. 3.** (a) aplicación multimedia InTml para Flash bajo un navegador compatible, (b) Aplicación InTml para Java ejecutada bajo eclipse.

Las pruebas se realizaron con un dispositivo Gamepad de 9 botones y 3 ejes obteniendo los siguientes resultados para la interfaz Web a través de Flash y la aplicación ejecutada por eclipse.

Tabla 1. Prueba realizada con una aplicación Intml-Flash un navegador Web compatible y un Gamepad

Hardware	Software	Tiempo de prueba (Segundos)	Tiempo de respuesta en Botones (Segundos)	Tiempo de respuesta en Análogos (Segundos)
Pentium IV 2,8 GHz con 1Gb DDR2 en RAM	Windows XP SP3 Internet Explorer 7	120.36 s	0.8456 s en 123 eventos	0.8374 s en 350 eventos
MSI Dual Core xon 2Gb DDR2 en RAM	Windows 7 Internet Explorer 8	125.46 s	0.7505 s en 85 eventos	0.7405 s en 217 eventos
Intel Dual –Core 1Gb DDR2 en RAM	Linux-Fedora Mozilla-FireFox	117.59 s	0.7887 s en 141 eventos	0.7725 s en 308 eventos
Equipo portatil Intel Dual-Core de 2Gb DDR2 en RAM	Windows Vista Opera	122.74 s	0.8269 s en 102 eventos	0.8368 s en 258 eventos
<b>Promedio</b>			<b>0.802925 s</b>	<b>0.7718 s</b>

Tabla 2. Prueba realizada con una aplicación Intml para Java y un Gamepad ejecutada desde eclipse

Hardware	Software	Tiempo de prueba (Segundos)	Tiempo de respuesta en Botones (Segundos)	Tiempo de respuesta en Análogos (Segundos)
Pentium IV 2,8 GHz con 1Gb DDR2 en RAM	Windows XP SP3	126.43 s	0.7056s en 125 eventos	0.6574 s en 384 eventos
MSI Dual Core xon 2Gb DDR2 en RAM	Windows 7	123.74 s	0.7205s en 95 eventos	0.5905 s en 257 eventos
Intel Dual –Core 1Gb DDR2 en RAM	Linux-Fedora	125.36 s	0.6866 s en 115 eventos	0.6727 s en 314 eventos
Equipo portatil Intel Dual-Core de 2Gb DDR2 en RAM	Windows Vista	118.54 s	0.7023s en 187 eventos	0.6268 s en 369 eventos
<b>Promedio</b>			<b>0.7023 s</b>	<b>0.63685 s</b>

Las tablas presentadas muestran bajo un tiempo determinado el promedio de respuesta desde el dispositivo hasta la salida grafica contando con un número de eventos producidos en el mismo tiempo, caso que se aplica para los botones y los análogos.

Las pruebas anteriores nos llevan a concluir que el cambio de latencia de una aplicación InTml para Flash es muy baja, aproximadamente de 0.2 s con respecto a una aplicación InTml para Java, diferencia que nos hace pensar que la implementación InTml para Flash es una buena propuesta para desarrollar aplicaciones enriquecidas para Internet (RIA) que hagan uso de dispositivos, como es

el caso de simulaciones académicas, o simplemente juegos de entretenimiento para la comunidad Internet.

Otra de las características a observar en este estudio es como el navegador influye al momento de la interacción, facialmente visto con la plataforma Windows 7 ya que el navegador Internet Explorer 8 esta optimizado para aplicaciones Flash de tal manera que los tiempos de acceso y visualización son mucho mas eficientes.

#### 4 Conclusiones y Trabajo Futuro

En este documento se propuso la utilización del framework de desarrollo InTml para aplicarlo a proyectos Web multimedia Flash de lo cual se puede rescatar:

- Una aplicación InTml Flash es un buen medio para presentar aplicaciones 3D multimedia para la Web aprovechando el concepto de intercambio de información con dispositivos.
- Cabe resaltar que el tipo de aplicaciones InTml Flash que se producen en muchos casos son de carácter mas estético y por ende de mas cogida por lo usuarios.
- Aunque se pretende en la actualidad un lenguaje nativo mas especializado 3D para Flash aun queda la necesidad de acoplarse a un motor eficiente como se hizo para generar este proyecto , así que basados en esta necesidad se propone para futuras versiones el desarrollo de un lenguaje de interfaces que encapsule y aproveche los motores 3D preexistentes.
- También es importante recalcar que el uso del servidor XMLSocket aunque eficiente requiere de la intervención del usuario para ponerlo en funcionamiento ya sea para interactuar en Internet o bajo una aplicación local, algo que termina siendo fastidioso par algunos usuarios de tal manera que en trabajos futuros se aprovechara esta desventaja para probar un modelo mas autónomo.

#### Referencias

1. Adobe Labs, <http://labs.adobe.com/technologies/flashplayer10>
2. Adobe Systems, <http://www.adobe.com/es>
3. Flash Player 10 y demás motores 3D, <http://www.nomeva.com/2008/10/15/flash-player-10-3d-nativo-vs-papervision3d/>
4. Swift 3D, <http://www.eraim.com>
5. Anton, Eliens.: Wiimote In Rich Internet Applications , Timen Olthof, Amsterdam (2008)
6. Figueroa, Pablo. Green, Mark, Hoover, H. James.: InTml: a description lenguaje for VR applications, ACM Proceeding of the seventh international conference on 3D Web technology, pp. 53-58, ACM, New York (2002)
7. Rusell M. Taylo II, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, Aron T. Helsler.: VRPN: A Device-Independent, Network-Transparent VR Peripheral System, Proceeding of the ACM Symposium on Virtual Reality Software & Technology 2001, VRST 2001. Banff Centre, Canada, November 15-17, (2001)
8. The Development of Multi-User Flash Game Application, [http://cis.k.hosei.ac.jp/~speng/files/lab\\_thesis/2006/Kondo/abstract.pdf](http://cis.k.hosei.ac.jp/~speng/files/lab_thesis/2006/Kondo/abstract.pdf)

9. Markus, Völter.: Introduction to openArchitectureWare: a flexible Open Source platform for model-driven software development. Germany, (2006)
10. PaperVision3D, <http://www.papervision3d.org/>