2012

Networked virtual environments with Javascript, WebSockets and WebGL

McGregor, Don

http://hdl.handle.net/10945/47800
Simulation App Deployment

- How hard is it to deploy a new application in NMCI or a similar managed environment?
- Very! Security review, extensive hoop-jumping
- Modern commercial environments are moving to web-deployed apps
- But 3D and complex simulations are (or were) challenging to create inside a web browser
New Technology Convergence

- WebGL: accelerated 3D graphics in the web page
- Websockets: TCP sockets directly to the web browser page, without AJAX
- Javascript: general purpose programming language that runs in the web page
- HTML5
WebGL

- Accelerated 3D graphics for Javascript in the web browser, from Khronos group (OpenGL)
WebGL

- Supported by recent releases of Chrome, Firefox, Safari, some mobile devices; standard created by Khronos, group that controls the OpenGL standard
- Not supported by Microsoft—they use the competing, equivalent Silverlight product.
- Effectively this is OpenGL ES for Javascript.
WebGL

- There are packages that layer on top of WebGL and provide scene graphs, model loading, and other functionality, such as X3DOM, Three.js, GLGE, Copperlict, etc.
- Before long I expect to see (more) commercial and open source game engines based on WebGL.
WebGL

- No plugins. Getting plugin approval in NMCI is difficult; WebGL is built into the web browser, and the Javascript WebGL frameworks are usually only several hundred KB and downloaded as part of the web page.
- NMCI uses very old browsers, and is MS based, but it’s somewhat plausible to expect a modern browser to be approved by the time the technology is mature.
WebSockets

- WebSockets are a joint standard between IETF (the protocol) and W3C (the API).
- Allow bidirectional full duplex TCP connections from a server to Javascript code inside a web browser window.
- This is a major advance over AJAX techniques, which typically revolve around client polling of the server, and therefore have very high latency.
- Can transit proxies.
WebSockets

Web Socket connection, negotiated with (near) http conversation

Web server with additional WebSocket capability on traditional port 80/443
Javascript

- General purpose scripting language, present in almost all browsers
- Performance has dramatically increased in the last few years in response to demands from AJAX web applications; generally speaking it is now “fast enough”
HTML5

• New <canvas> and <video> tags allow much more flexibility
• Embed WebGL scene in a <canvas> element
Architecture

Server: 3D content, Javascript files, communication with web browsers

DIS in JSON Format

Native DIS Traffic

Web Browser Window

Web Browser Window
Architecture

• Native DIS traffic on UDP on the server side—this can be any DIS application or an HLA application with a DIS gateway
• WebSocket connections from the server to each browser window pass Javascript JSON-format DIS
• Server acts as central hub and can relay DIS to the native network
• Not necessarily virtual; can have plan views, data collection & sim management web apps (these are lower risk initial applications)
Javascript

• Can use WebWorker standard for multiple threads in one page (with limitations)
• Javascript packages for physics are available
• Potential for server-side area of interest management/DDM
• Clustering on data center side for scalability?
• Cloud deployment?
• Content Distribution Network for static files
Performance

• Several hundred JSON format DIS PDUs per second to browser windows!
• Acceptable latency on LANs (with expected TCP vs. UDP pathologies on bad connections)
• Good enough for small-to-medium Networked Virtual Environments
• All the inherent problems of TCP. Worst case is dropped packet/timeout-resend
SISO WebLVC

- “WebLVC is an interoperability protocol that enables web-based applications (typically JavaScript applications running in a web browser) to interoperate in Modeling and Simulation (M&S) federations.”
- Intended to work across multiple standards (DIS, HLA, TENA) though focused on RPR-FOM/DIS style information
- Focused on network protocol, not graphics
- Working group just starting; siw-sg-lvc@discussions.sisostds.org
- Potential to be disruptive technology
Demo

- [http://oam.nps.edu](http://oam.nps.edu) has demo; wsad keys to navigate
- Source code available on sourceforge open-dis project