

Simulation Interface and Visualization with Existing Internet Technologies

Jason Schutz, Christopher Markuck
SAIC
Orlando, FL
Jason.D.Schutz@saic.com, Christopher.D.Markuck@saic.com

ABSTRACT

With the introduction of terrain imagery applications downloadable over the Internet, there is a desire to leverage these technologies in the world of simulation. Google Inc.'s Google Earth™ is a freely available application that gives the user a view of satellite imagery, 3D terrain, and Geographic Information System (GIS) data from a database over the Internet. The user is able to import customized placemarks, shapes, and images into Google Earth and overlay them on the terrain using Keyhole Markup Language (KML). Combined with the ability to refresh such data over the network when connected to an HTTP server, the user can retrieve the visual data that represents the simulated entities and events. The entities can be represented using any of the objects available in KML, such as icons or "COLLABorative Design Activity" (COLLADA) models. Finally, Google Earth permits the display of feature data such as roads, rivers, and lakes, which can be exported from existing terrain databases to KML.

This paper presents the results of a prototype development effort which implemented a simulation adapter for the U.S. Army's One Semi-Automated Forces (OneSAF) system. The adapter serves as the Web server and bridge to the simulation for Google Earth to retrieve KML data representing the simulation. In addition, the adapter provides access to the simulation information through a Web browser. The prototype consists of two components. The first serves as a bridge from Distributed Interactive Simulation (DIS) to KML by running as a separate process. The second is embedded directly into OneSAF as a OneSAF Component, where the amount of data served is limited only by the amount of information that OneSAF can provide to the Component.

ABOUT THE AUTHORS

Jason Schutz is a Software Engineer for Science Applications International Corporation (SAIC) and works on the Environment Runtime Component (ERC) of OneSAF Objective System (OOS). Mr. Schutz has six years experience in the simulation industry and has a Masters of Computer Science from the University of Central Florida.

Christopher Markuck is a Software Engineer for Science Applications International Corporation (SAIC). Christopher received a B.S. in Computer Science and Computer Engineering from the University of South Florida. He has worked on system performance for the OneSAF Objective System (OOS).

Simulation Interface and Visualization with Existing Internet Technologies

Jason Schutz, Christopher Markuck

SAIC

Orlando, FL

Jason.D.Schutz@saic.com, Christopher.D.Markuck@saic.com

INTRODUCTION

As we transition from geotypical to geospecific simulation events, our ability to provide the warfighter with high quality environments is increasingly falling short of expectations. To this end, we have been investigating the means of providing a richer virtual presence. The limitation on the database development technologies and graphics engines has discounted the use of polygonal representations for worlds of any scale. This has led us back to the image drape technology. However, the cost of acquiring, rectifying, and managing the geospatial imagery is not a long-term viable option. This, in turn, led us to investigate the use of commercial off-the-shelf (COTS) products for this type of application. It is this path that leads us to the Google Inc.'s Google Earth system. Realizing the inherent limitations of an image draped system, we are focusing the use of Google Earth as a visualization and after-action platform rather than a virtual scene generator. This paper discusses the underlying technology and approach we have used in this endeavor.

PROPOSED SOLUTION: GOOGLE EARTH

The proposed visualization system uses Google Earth COTS application and Web technologies. It will let the instructor of a training exercise view the simulation in a 3D virtual world with the aid of a typical Web browser to acquire ground truth knowledge and operational picture. The tools are built for ease of use and ease of setup. It is designed to allow multiple viewers into the simulated world. The solution also is extendable by design, allowing proprietary, non-disclosed terrain data to be used in place of the default provided with Google Earth.

The Google Earth Adapter will take information from the simulation and convert it into data that can be represented in Google Earth. The adapter will provide the user of Google Earth with an overview of the current simulation (entity status, recent weapon fires, current simulation status, etc.) on the Google Earth map for enhanced situational awareness. Another feature

supported by the adapter is a Web interface to the simulation which will provide detailed information on the simulation and give limited control to authorized users.

OVERVIEW OF GOOGLE EARTH TECHNOLOGY

Google Earth is a terrain imagery application that gives the user a view of satellite imagery, 3D terrain, and Geographic Information System (GIS) data such as roads and political boundaries from a central database. Using drawing tools in the application, the user has the capability to create customized placemarks, shapes, images and overlays. This information can also be imported or sent over a NetworkLink to Google Earth by utilizing the KML (Keyhole Markup Language) data format.

The terrain imagery and elevation data is downloaded off the Internet on demand. If working in off-line mode, Google Earth will use data that has been cached since the last use of the application while connected to the Internet.

Google Earth Specifications

Google Earth utilizes the World Geodetic System from 1984 (WGS84) with mostly 15-meter resolution images. It has easy access to geography across the globe with fast download, fast rendering of the terrain, and highly optimized level-of-detail (LOD) techniques. It is intuitive to use, thus allowing the user to focus on the simulation. The currently supported platforms are Microsoft Windows®, Mac OS® X, and Linux®.

Google Earth Programming Interface

Google Earth registers a Microsoft Component Object Model (COM) object with Microsoft Windows which exposes a small set of objects. An example of a function call is commanding Google Earth to fly to a location.. From a Web page, JavaScript™ can obtain a reference to the Google Earth COM object using the

ActiveX® application programming interface (API). With this object it can send a command to the COM object to fly to a location on Google Earth.

Google Earth Versions

Google Earth is available as a free version, Plus, Pro, and Enterprise. The Plus version adds GPS waypoint imports, polygon drawing. The Pro version is for commercial use, with optional modules for “Movie Making” and GIS importing. The Enterprise version allows seamless integration of GIS terrain data and buildings into Google Earth by using the Google Earth Enterprise Server. Work for this prototype was done initially with Google Earth Free version when work was done on this as a hobby, and then we switched to Google Professional edition when we decided to pursue more research with it.

Keyhole Markup Language

KML is a language for describing data inside of Google Earth. KML is based on XML, and follows XML syntax rules. Figure 1 shows an example KML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml
xmlns="http://earth.google.com/kml/2.0">
<Placemark>
  <name>Orlando</name>
  <LookAt>
    <longitude>-81.37944</longitude>
    <latitude>28.53805</latitude>
    <range>1007.50645</range>
    <tilt>0</tilt>
    <heading>0</heading>
  </LookAt>
  <Point>
    <coordinates>
      -81.37944, 28.53805, 0
    </coordinates>
  </Point>
</Placemark>
</kml>
```

Figure 1. KML Example

With KML, placemark icons with labels can be created at specific geodetic locations. Image overlays can be put on the terrain, useful for custom maps. The data can be organized into folders in the KML document. Introduced in KML 2.1, “COLLABorative Design Activity” (COLLADA) models can also be referenced from the KML Model tag, with the 3D transformation performed by Google Earth.

KML data can be dynamically downloaded from a Web server over HTTP or HTTPS by using KML NetworkLinks in Google Earth. The described prototype takes advantage of this type of dynamic KML generation. On Microsoft Windows, Google Earth uses the Windows Internet (WinINET) API, which is the same Internet API used by Microsoft Internet Explorer®. Internet access configuration on Internet Explorer propagates to Internet access on Google Earth. This enables Google Earth to communicate with the Web server over a secure connection.

PROTOTYPE

The prototype we are introducing uses Google Earth and Web-based technologies to provide the user a view into the simulation. The functionality provided through KML NetworkLinks is used to dynamically display entity movement, and simulation events in real time. The entities are represented as icons, with the movement of the entities animated by a KML NetworkLink which periodically requests KML from the adapter. Figure 2 shows how Google Earth will display the unit hierarchy of simulated entities.

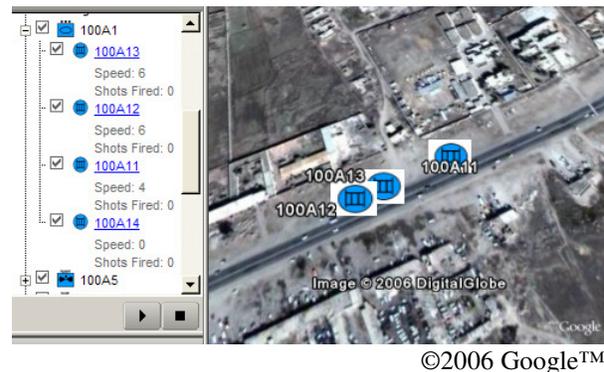


Figure 2. View of the Unit Hierarchy.

Entity movement is also visualized through line trails (3D snail trails), representing the previous positions of the entities, as shown in Figure 3. The trails are extruded to the ground to display the elevation of the entity.

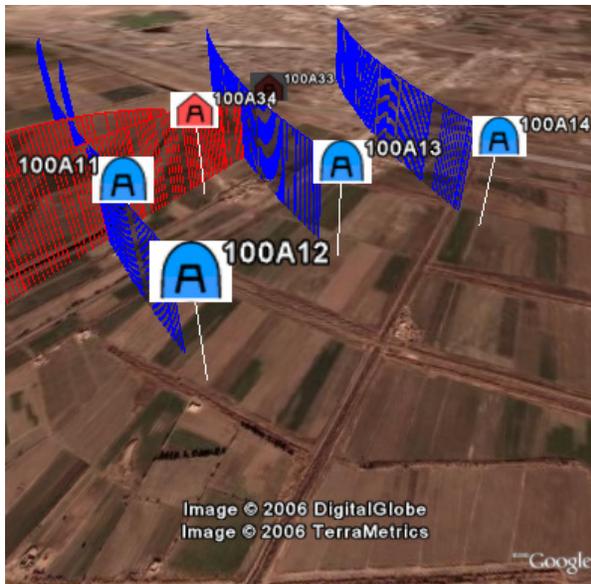


Figure 3. Snail trails of the entities.

Icons are also created for fires and detonations. The fire and detonation events can be matched to create KML lines that are extruded to the terrain to show each line of fire. As shown in Figure 4, the fire lines are converging on the opposing tank platoon.

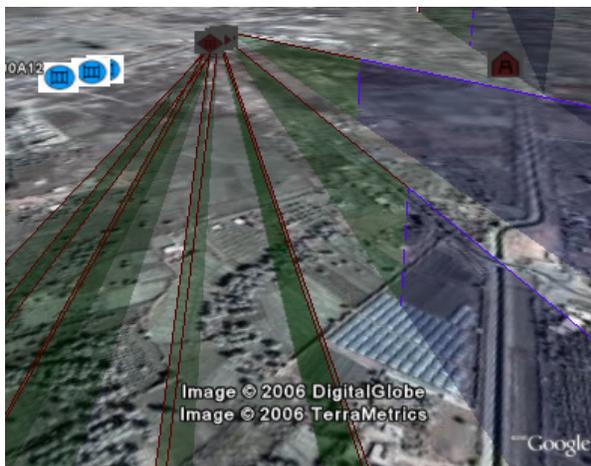


Figure 4. Fire lines.

The simulated terrain feature information can also be shown on Google Earth. Shown in Figure 5 are buildings loaded from the OneSAF terrain format in real time. The view in Google Earth will determine which features are downloaded from the simulation. This provides correlation between the 2D image and the 3D simulated world.



Figure 5. Buildings dynamically downloaded from simulated terrain.

The Web interface that is a part of the OneSAF adapter provides real-time access to all simulation information such as the status of all the nodes in the distributed simulation. An example of possible information in the status page is simulation state, object load, capability, and memory usage. Entity information is enhanced by being able to provide weapon status, sensor/weapon range and supply status of the selected entity. Reports generated by OneSAF, such as Observation reports, can also be viewed through the Web interface. Also, all users who access the OneSAF adapter interface will be required to have a username and password specified by the OneSAF system. Properly authorized users will have limited control over the simulation.

OneSAF Google Earth Adapter Simulation 5

Simulation Status

Simulation Engine Node Monitor

Node	State	Simulating Model	Number of Objects
onesafab01	Loaded	Yes	39
onesafab02	Loaded	Yes	39
onesafab03	Loaded	No	0
onesafab04	Loaded	No	0

Figure 6. OneSAF Web Interface – Simulation Status

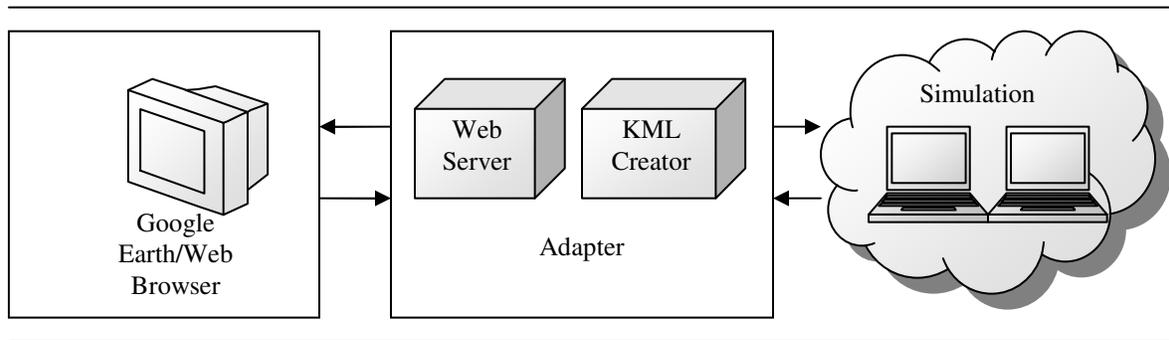


Figure 7. Overview of System Components

SYSTEM COMPONENTS

The components of the system are diagrammed in Figure 7. Google Earth and the Web Browser both communicate with the adapter, which creates data in the format usable by them. The adapter will retrieve the data from the simulation by listening to the simulation network and by querying the simulation objects directly.

KML NetworkLink

Google Earth allows the KML data to get updated dynamically by using a special KML tag called a "NetworkLink". There are a few different types of NetworkLinks that can be applied. For instance, the simulated entities will have a NetworkLink which has a periodic update. Also, multiple NetworkLinks can then be used to update different entity platform types at different intervals. A NetworkLink to update static information such as features can use either the one-time update, or the region-based update. The region-based update sends the information about the camera location and orientation to the HTTP server as part of the request. The server can then send a new KML document for the new view.

However, KML 2.0 NetworkLinks have a side effect. If a placemark is created from a KML document downloaded from a NetworkLink, it must be deleted by Google Earth to be refreshed with the latest placemark. This is evident when a placemark pop-up description balloon is being viewed. The description balloon is created when the placemark shown on the map is clicked by the user. When the placemark is removed for the new placemark, the balloon disappears.

KML 2.1 alleviates the refresh issue from KML 2.0 by allowing a second NetworkLink to introduce updates to the KML data loaded from the first NetworkLink. This will allow new placemarks and geometries to be created

in the original KML data or changes to existing KML data. Finally, it is possible to delete KML data from the original KML data.

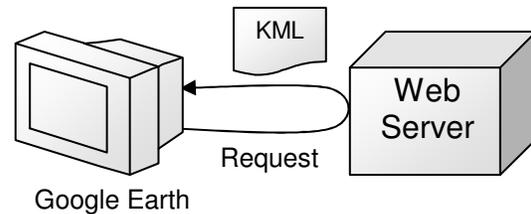


Figure 8. Google Earth Refreshes KML through the NetworkLink

A more advanced technique for refreshing KML data based on view has been introduced in KML version 2.1. Level-of-detail (LOD) support in KML 2.1 will allow multiple levels of NetworkLinks for specific regions of the terrain. For example, a 1 degree by 1 degree image overlay can be partitioned into four equal sized boxes. Each of the four boxes will have a NetworkLink to download the KML data of an image overlay for itself when the viewer is close enough to the box. The downloaded KML data for the box will also have four more NetworkLinks for an additional four smaller boxes

Web Server

The Web server uses the Hypertext Transfer Protocol (HTTP) to serve requested files and uses the Web page generator combined with the KML creator to create dynamic responses. The request coming from the KML NetworkLink expects a KML document or a compressed KML (KMZ) attachment as a response. This kind of request to the Web server will contain specific information in the HTTP request string regarding the type of KML data required.

Web Page Generator

In the Microsoft Windows version, Google Earth has an embedded Web browser that can be used to view HTML pages. The adapter utilizes this functionality to server HTML pages with the simulation status, simulated entity status, an interface to control to simulation. The adapter utilizes this functionality to serve HTML pages with the current status of the simulation, the status of simulated entities (weapon status, damage, speed, location, orientation), and to provide an interface to control the adapter. The Web page generator utilizes AJAX (Asynchronous JavaScript and XML) to provide periodic updating of simulation data. Simulation data is updated behind the scenes so updates seem transparent to the user.

Also, controls are available to control Google Earth. For instance, while viewing entity information you can click on the “Fly to Entity Location Button” as seen in Figure 9 to tell Google Earth to change its current view to show the selected entity. The Web page generator utilizes ActiveX to send commands to Google Earth.

General	
Entity Name	1/MIA1Platoon:1
Composition	entity/mr/COMBAT/ARMOR/Tank_MIA1_Armor_US
Type	MIA1
Activity	Stationary
Location	GOC: -298.29 -5,445.62 3,299.42
Orientation	90.03
Formation	Line
Speed	20 Km/Hr
Damage	Healthy
Weapon Max Range	4400.0 m
Sensor Max Range	7000.0 m
Weapon Control Status	HOLD

Figure 9. Web Interface showing entity status

KML Creator

The KML creator is the interface between the Web server and simulation subsystem, such as Distributed Integrated Simulation (DIS), OneSAF Object Database (ODB), and OneSAF Environment Runtime Component (ERC). When a request for data is received by the Web server the request is forwarded to the KML creator to retrieve the appropriate information from the simulation and to create a KML file with the requested data. The KML document is then served by the Web server to Google Earth for processing.

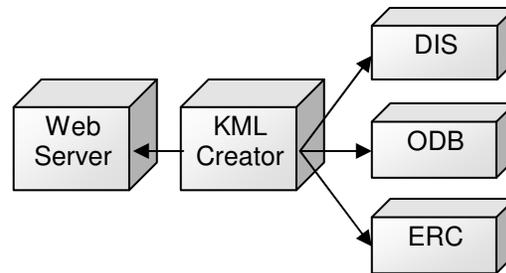


Figure 10. KML Creator can access DIS, ERC, or ODB

Custom icons for placemarks can be used to represent specific entity types, fire events, or detonation events. Each KML placemark references a certain icon style. Each icon style then references an icon file. The icon file can be accessed either from the local disk drive, from inside a KMZ file, or it is accessed from the Web server directly using a Uniform Resource Locator (URL) as the location.

OneSAF Object Database

OneSAF applications are created by using interdependent software “components”, which are formed into a “composition”. An implementation of the adapter was created as a OneSAF component that can be included in any composition utilized by OneSAF. The adapter will interface with the OneSAF Object Database (ODB) in order to pull information on simulation status, entity status, and reports. The ODB is a database distributed over the simulation network. It stores all the objects relevant to the simulation. Since the adapter is part of the OneSAF framework it is able to get much more information as compared to the information from DIS. Also, being part of the OneSAF framework allows the adapter to have limited control during the simulation.

OneSAF Environment Features

The KML creator can also produce KML data representing features in the user’s current view. A bounding box KML NetworkLink is used to accomplish this task. When the user of Google Earth repositions the camera view, the NetworkLink will send the HTTP request for the features in the new view. The KML creator will then produce the KML data of the features for the current view. The OneSAF Environment Runtime Component (ERC) provides the access to the feature information for the Objective Terrain Format (OTF) database.

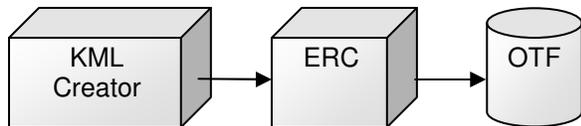


Figure 11. KML Creator may use OneSAF ERC

ANALYSIS

The initial goal of the prototype was to display DIS entities on Google Earth. It expanded into DIS events and migrated into OneSAF as a OneSAF component. Since it resides in OneSAF, there was effort on creating units, controls measures, and tasks for the units from Google Earth. The limited application programming interface into Google Earth proved to be a hindrance to creating objects for the simulation. There are techniques that make it possible to create certain objects in the simulation, but this made the user interface less friendly.

While the use of COTS products enables significant cost and development savings, it has forced some architectural compromises that may limit us down the road. Having said that, our overall development experience was quite pleasant, and we were rapidly able to develop a robust system.

The earlier version of the KML specification used with Google Earth gave limited ability to animate the movement of the icons without deleting the current list of entities and redrawing them. The problem was fixed with the new functionality of later versions of KML.

Other possible applications which can be used for visualization are two-dimensional Microsoft Virtual Earth™, two-dimensional Google Maps, three-dimensional NASA World Wind, and three-dimensional Environmental Systems Research Institute, Inc. (ESRI)™ ArcGIS™ Explorer. Microsoft Virtual Earth and Google Maps will require access to internet to download terrain imagery; however they will provide a rich programming environment for direct manipulation of object on the map. NASA World Wind is open source, allowing developers to take full advantage of the functionality currently available. As with Google Earth, ESRI ArcGIS Explorer can use KML data, along with data from ArcGIS Server. As of this writing, ArcGIS Explorer has not been released. In addition to ArcGIS Explorer, Arc2Earth is an extension to ArcGIS which allows the user to export

GIS shapefile, coverages, and other GIS formats into KML format.

Integrating into OneSAF

We chose to integrate the Google Earth adapter into OneSAF due to its component architecture and role in most next generation simulation systems. The component architecture allows us to create a Google Earth component that can be included in any compositions as long as the required component dependencies are met. The component allows us to add the Google Earth Adapter functionality to OneSAF without modifying source code. When the component is included in a composition the adapter will have access to all the information contained in the Object Database.

FUTURE ENHANCEMENTS

This prototype was purely for proof of concept. There are many possible enhancements and variations for whatever type of training and situational awareness is required.

OneSAF Adapter

Currently the OneSAF adapter web interface provides minimal control over the simulation. Future implementations of the OneSAF adapter will provide more control over the simulation. This new functionality will be dependent on the availability of a SDK from Google Earth. We are still in the process of extending the prototype to add increased functionality and to conduct scalability testing. On the scalability side, the goal is to support a OneSAF Objective System brigade-level exercise.

Integrating into other Simulations

There is the future possibility of integrating a variation of this adapter into other simulation protocols such as High Level Architecture (HLA). The same techniques can be done as in the DIS case. However, in the HLA version, the adapter will need to take a more proactive role and subscribe to the objects and interactions of interest. However, a more embedded approach can be taken, and did in the OneSAF Component case. The component “plugs in” to the simulator code, gaining more insight, and feeding the KML data with more information.

ACKNOWLEDGEMENTS

Special thanks David Reese, Dr. David Pratt, Benito Graniela, Robert Burch, and Eric Root.

REFERENCES

Arc2Earth

<http://www.Arc2Earth.com>

COLLADA – Digital Asset Exchange Schema for Interactive 3D. <http://www.khronos.org/collada>.

ESRI ArcGIS Explorer

<http://www.esri.com/software/arcgis/explorer/index.html>

IEEE Standard for Distributed Interactive Simulation – Application Protocols. IEEE STD 1278.1a-1998.

Google Earth.

<http://earth.google.com/>

Google Earth KML (2006)

http://earth.google.com/kml/kml_toc.html

Google Earth – Wikipedia, (2006)

http://en.wikipedia.org/wiki/Google_Earth

Microsoft Virtual Earth

<http://local.live.com>

NASA World Wind

<http://worldwind.arc.nasa.gov>

Microsoft, Windows, Internet Explorer, and ActiveX are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Mac OS is a registered trademark of Apple Computer, Inc. in the United States and /or other countries.

Linux is a registered trademark of Linus Torvalds in the United States and/or other countries.

JavaScript is a trademark of Sun Microsystems, Inc. in the United States and/or other countries.

Google and Google Earth are registered trademarks and/or trademarks of Google, Inc. in the United States and other countries.

ESRI and ArcGIS are trademarks of Environmental Systems Research Institute, Inc.