Open and Decentralized Platform for Visualizing Web Mash-ups in Augmented and Mirror Worlds

Vlad Stirbu
vlad.stirbu@nokia.com

David Murphy
david.j.murphy@nokia.com

Yu You
yu.you@nokia.com

Multimedia Technologies Laboratory
Nokia Research Center
Visiokatu 3, Tampere 33720, Finland

ABSTRACT

Augmented reality applications are gaining popularity due to increased capabilities of modern mobile devices. However, existing applications are tightly integrated with backend services that expose content using proprietary interfaces. We demonstrate an architecture that allows visualization of web content in augmented and mirror world applications, based on open web protocols and formats. We describe two clients, one for creating virtual artifacts, web resources that bind together web content with location and a 3D model, and one that visualizes the virtual artifacts in the mirror world.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: On-line Information Services - Web-based services

General Terms

Design, Languages, Experimentation

Keywords

augmented reality, mash-ups, feeds, www, authoring

1. INTRODUCTION

Augmented reality and mirror world applications belong to the mixed reality section of Milgram's Virtuality Continuum [3]. While augmented reality applications are overlaying digital artifacts over a see-through display for in-situ exploration, the mirror world applications aim to create a realistic virtual replica of the world that enables remote exploration. The data displayed in both application domains is geo-tagged and is typically fetched from specialized geo-index servers that are tightly coupled with the viewers.

Our approach on decoupling the augmented reality and mirror world clients from the infrastructure services leverages standard web protocols and formats for representing geo-tagged data and 3D models. Similar to KHARMA [2], we use open standards and technologies as the foundation for bring web content into augmented reality and mirror world applications.

In this paper we demonstrate an application platform that enables the visualization of web mash-ups in mobile augmented and mixed reality application scenarios. The web mash-up role is played by a Twitter stream that is relevant to a specific geographic location, then we attach the specific geographic coordinates where the data will be visualized, and a 3D model to the stream. The demonstration includes a part in which we use a web tool to create a virtual artifact that bundles the data from the web mashup together with geo location and 3D visualization, then we explore the artifact in the mirror world application, while new tweets that are posted to the stream are visualized in real time in the mirror world.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

2. INFRASTRUCTURE SERVICES

Many augmented and mixed reality services are either vertically integrated, or are built on concepts that are specific to one service. Our work focuses on having a common generic augmented and mixed reality backend service [4], which provides a web interface that allows clients to perform operations on user-generated content, or commercial geo-content such as street-view panoramas and building models. The service maintains a spatial index that allows client applications to easily retrieve content collections that match specific application needs, using geo or reverse-geo queries.

This infrastructure forms the backbone that allows us to expose arbitrary web content in augmented and mirror worlds. The web content is augmented with a physical location and a 3D model and stored by the service as a *virtual artifact*. The virtual artifact is a fully fledged web resource, having its own URI, a KML representation [6], and a web interface that allows clients to interact with it. The KML representation contains links that convey the 3D model (e.g. a link to a Collada document [1]), and the content to be displayed (e.g. a link to the site that provides the web content).

Along with the spatial content management, the server provides a text-to-texture conversion function, enabling a lightweight content placement and rendering mechanism suitable for mobile devices, which does not require embedding a web browser engine or any client-side rich-text processing into the viewer for each web content in a scene (e.g. the KHARMA approach). The text-to-texture feature notifies the clients whenever the converted web content changes [5]. A web client can decide if the text-to-texture or embedded-browser method suits better and informs the server via content negotiation. Since the virtual artifacts are accessible over open web interfaces, third party applications can create new web mashups using a browser (Section 3), or consume the created content in a mirror world or augmented reality application (Section 4).

3. THE VIRTUAL ARTIFACT BUILDER

The artifact builder is a lightweight web application. The application allows a user to bind, using a map interface, web data provided by the origin server or a mashup service, with a geographic location and the appearance in the physical world. The binding

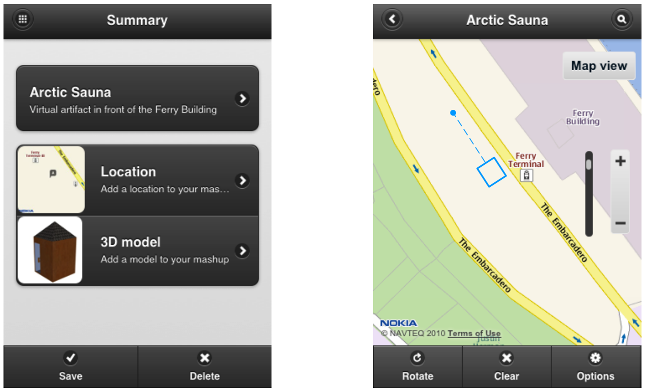

Figure 1: The virtual artifact builder: summary view (left), and location editor view (right)

Figure 2: City Scene mirror world viewer

consists in attaching to the data a physical position and a 3D model for presentation.

The artifact builder is a pure client-side browser application, written in JavaScript, having two main functional components, an artifact management interface and an artifact editor. The artifact management interface, practically a feed reader, allows the user to load existing artifacts from the artifact server, each artifact corresponding to an entry in the feed. The artifact editor allows the users to visualize existing artifacts or create new ones using a map-based interface.

When the artifact builder is loaded (see Figure 1), it reads a list of subscribed feeds containing the artifacts belonging to the current user. After the user clicks on an artifact, the artifact builder fetches the artifact and creates a summary view that presents a thumbnail of the location of the artifact on a map, a thumbnail of the 3D model and a description of the web mash-up used. The summary page allows the user to select the web mash-up from the user's available Yahoo Pipes that corresponds to the Twittr stream, the 3D model and the artifact's location. The user has to select a 3D model provided by the server, or by providing a URI of an external model. Once the 3D model is selected, the user can place the object at a physical location using a map-based interface. The 3D model is presented on the map using an overlay that contains the polygon shape of the model's footprint. The user can move and rotate the polygon to the desired position. The artifact builder application automatically adjusts the geo-coordinates of the 3D model.

If the artifact server has read/write capabilities, the user can create new artifacts, or edit and delete existing artifacts using the HTTP methods `POST`, `PUT` or `DELETE` respectively.

4. THE MIRROR WORLD VIEWER

CityScene[1] is a mirror world viewer application, running on mobile devices, that provides a realistic visualization of the world using street-level panorama images augmented with 3D building models. The application enables users to visualize their surroundings, if the they decide to use the local positioning sensor, or to explore remote locations.

The application finds relevant content for the specific locations that are visualized from the mixed reality backend service using geo-tagged queries. Responding to a query, the service provides a list of objects that are in the proximity of the given panorama image coordinates. An object (typically KML) contains the geo coordinates and the associated 3D Collada model. The application, renders the objects into the 3D scene, taking into account the visibility and possible occlusions against the terrain mesh and other objects.

For example, in Figure 2, the mirror world viewer application visualizes the virtual artifact created earlier into the corresponding scene in front of the Ferry Terminal building in San Francisco. The rendered 3D building model that represents the Twitter stream gets updated, without user intervention, each time a new tweet is posted. Because our viewer application is fully aware of the surrounding 3D environment, it can operate in an augmented reality mode, to render the virtual artifacts onto images obtained from the camera viewfinder, without changing the data formats or the interaction pattern with the web-based backend.

[1] http://betalabs.nokia.com/apps/nokia-city-scene

5. REFERENCES

[1] M. Barnes and E. L. Finch. *COLLADA - 3D Asset Exchange Schema, Release 1.5.0*. Khronos Group, 2008.

[2] A. Hill, B. MacIntyre, M. Gandy, B. Davidson, and H. Rouzati. Kharma: An open kml/html architecture for mobile augmented reality applications. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 233 –234, oct. 2010.

[3] P. Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems*, vol. E77-D(no12):1321–1329, 1994.

[4] P. Selonen, P. Belimpasakis, Y. You, T. Pylvänäinen, and S. Uusitalo. Mixed reality web service platform. *Multimedia Systems*, pages 1–16, nov. 2011. DOI: 10.1007/s00530-011-0254-9.

[5] V. Stirbu and J. Savolainen. Hypermedia-driven framework for scalable and adaptive application sharing. In *REST: From Research to Practice*, pages 199–219, 2011.

[6] T. Wilson. *OGC Keyhole Markup Language, 2.2.0*. Open GIS Consortium, 2008.