

# A component-based approach for mobile AR applications

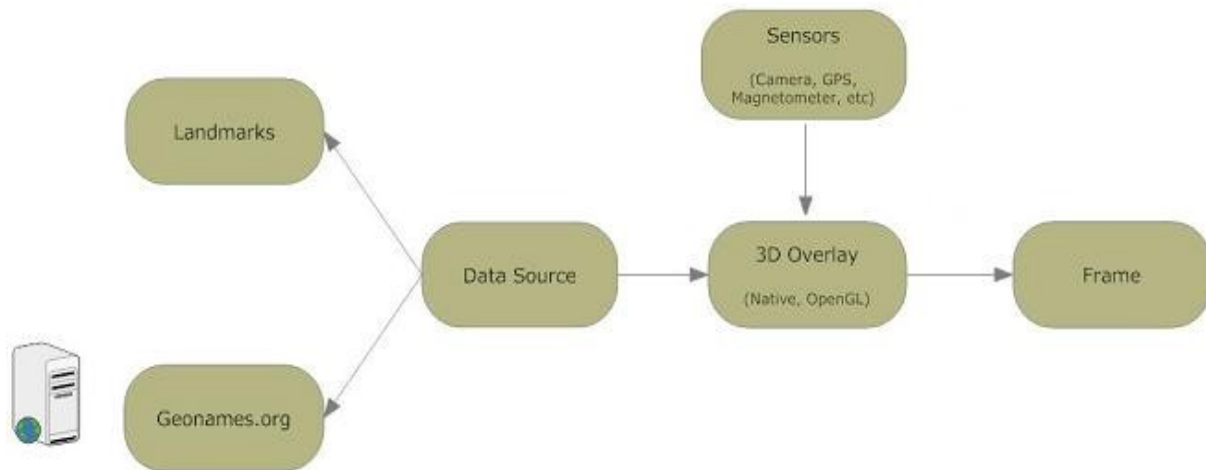
David Caabeiro – Sequence Point Software S.L.

Contact: David.Caabeiro@seqpoint.com

Presented for the Mobile AR Summit – February 17<sup>th</sup> 2010

This short article describes the design of a simple framework for mobile augmented reality applications.

By using a multi-tier plug-in design, we allow seamless use of different and non-related geolocated data formats. Also by following the practice of using established standards (like LMX, KML, etc) this approach enables easy interoperability and gives freedom to the user by allowing him to create their own data repositories in an easy way, without forcing him to use proprietary tools or services.



What follows is a short description of the different components in this framework, followed by some simple use cases:

The *overlay* component retrieves data from a group of sensors, and requests a group of data sources for related information in order to render it in some way.

Within a data source, we can describe two different components. First we have a *provider*, which retrieves information either locally or remotely, and optionally uses a *renderer* which is in charge of displaying the information in some way.

In the case of a simple AR browser, the *overlay* would make use of orientation sensors (magnetometer + accelerometer) and a position sensor (GPS or WLAN) in order to issue the request the data *provider*. Once associated data (POIs) is retrieved, the overlay would make use of the *renderer* in order to create the scene. This scene will be blended with the camera frame.

Note that several overlays could be created, each one using specific data sources. As an example, we could have a “noise” overlay which would show to the user information about the noise level in his surroundings. This overlay would use an audio sensor (microphone), perform some basic analysis and display the information in some way to the user. This overlay could be blended with other existing overlays, using different levels of opacity.