

A Proposal for Reference AR System Architecture Based on Standardized Content Model

Gerard J. Kim,¹ Youngsun Kim²
Korea University

and

Christine Perey³
PEREY Research & Consulting

ABSTRACT

In this paper, we consolidate basic requirements for a standard AR content model and browser architecture. The core aspect of any AR content lies in the manner and abstraction level for representing the physical object, i.e. the augmentation target, and its relation to the virtual counterpart. We then survey several major mobile augmented reality content representations and associated browsers, and assess them with respect to their amenability to standardization and for interoperability. We conclude that the abstraction level must be specified consistently and in a fashion such that the representation is independent of a specific implementation for sensing or acquiring the user's contextual information. Such a provision would decouple the browser architecture into two parts: one interpreting the content and transforming it into an abstract scene model and the other mapping the digitally rendered components components into the scene.

KEYWORDS: Mobile Augmented Reality, Browser Architecture, Standards, Augmented Reality Content.

1 INTRODUCTION

With high performance computational, multimedia and sensing capabilities in mobile devices, augmented reality (AR) applications have emerged as compelling ways to receive, view and interact with digital data using the mobile devices. As such, many commercial and non-commercial AR applications have appeared. In order to scale across multiple AR content sources and multiple content viewing platforms, platform providers and content publishers must agree upon a common architecture to share such information or augmentations across a multitude of devices OSs, use cases and hardware. Content publishers and AR platform providers have yet to agree on a common architecture. This can be attributed to differences in the (1) platforms or devices, (2) types of AR (e.g. location-based, vision-based), (3) numerous tracking and sensing technologies, and (4) a variety of content presentation/methods. In short, the providers lack a “generally applicable and comprehensive” AR content model.

In this paper, we consolidate basic requirements for a general and standard AR content model and browser architecture. The core attribute of any AR content is the manner and abstraction level for representing the physical object, i.e. the augmentation target (e.g. a marker, a landmark, points of interests (POI's)), and its relation to the virtual counterpart (e.g. text, 3D digital object, image, sound). In order to promote standards and content sharing, we assert that the content model must be abstract enough to be independent from particular technologies for “playing” or “delivering” the experiences to the user e.g. through the browser or underlying AR application.

We focus on this point in the subsequent survey and analysis of different commercial and non-commercial AR content representations (file formats) from which we are able to portray the structure of the associated browser implementations. In particular, we limit the survey to those which fit one of two representative or typical forms of AR content: “vision-based” and “location-based” AR contents, both of which have to sense and recognize a physical trigger in the real world and associate it to a virtual object for rendering. Before commencing the survey, we first lay out the main aspects of a content model.

¹ gjkim@korea.ac.kr

² hide989@korea.ac.kr

³ cperey@perey.com

2 INITIAL PROJECTED REQUIREMENTS

The very first necessary condition for standardization seems to be the (perhaps obvious) decoupling of the content and the browser/application. For example, viewing of the Web pages is realized by representing their contents using a standard mark-up language (like HTML [1]) then allowing any browser implementation perform the actual visualization of the page. The user is not subjected to the idiosyncratic details of the “technical” realization of the intended content through the abstractions provided by the mark-up language.

Given this analogy, the next requirement might be for the AR mark-up language or file format, to be “reasonably” comprehensive in the types of expressible AR contents. We put forth the following required components to be considered for minimum comprehensiveness. In addition, we assert that the encodings of the information for these components should use and leverage on the existing standards as much as possible and be extensible as well (to accommodate new future AR technologies).

2.1 World Model

The world model refers to a rich scene graph-like data structure that is able to represent many different types of objects such as not only 2D/3D models but also various types of multimedia data (e.g., structured document, text, videos, images, panorama, sounds, etc.). These objects may serve as augmentation or proxies/placeholders for physical objects. In addition the world model also organizes the objects into a “scene”, that is, encodes the spatial relationship among the objects.

Distinctions may be made between the purely virtual or those that have physical correspondences, and furthermore, there needs to be a construct for specifying the augmentation itself (i.e. what is augmented by what?), a connection between an object representing the physical (sensed from the environment) and the virtual. Note that in actual implementation, the MR/AR scene is “modelled” and rendered as a virtual world (using the aforementioned world model). Fortunately, we already have several standards for representing the 3D virtual worlds which we should be able to extend for AR/MR purposes [2][3].

2.2 External Sensing of the Physical Object

In addition to the basic structure for representing the whole scene, for AR/MR, new constructs are necessary for representing the “real world objects” (e.g. markers, 3D objects and points of interests (or POIs)). As will be seen in the survey, many vendors and organizations have proposed such constructs (e.g. KML, ARML) at different abstraction levels.

These real world object constructs will probably require specifications of “features” necessary for sensing (e.g. visual description of a marker), optionally with the characteristics of the sensor itself (e.g. camera and its parameters). Note that, for instance, theoretically, a marker may be sensed and recognized by means other than using a camera. It is arguable, thus, whether to include the sensing details or not. Abstracting them away will leave it to the browser or application to generate the necessary information for the target physical object without specifically stating which sensor should be used.

2.3 Communication with external entities

AR/MR systems will often have needs to communicate with external entities to retrieve and upload the various content elements. These may include content servers, database servers, e-mail and mash-up application servers. In fact, even object tracking and recognition may occur on remote sites. Standards for internet communication and remote procedure calls are mostly applicable [4][5].

2.4 Interaction and Behaviors

For sophisticated forms of dynamic and interactive content beyond just simple and static augmentation, the basic world model needs to be extended with a power to express dynamic behavior and allow user interaction. Invocation of scripts, remote mash-up application protocols, and GUI object abstraction may be means to achieve such a requirement. HTML standards have this capability through mark-ups for UI objects, and Java scripts and bindings [1].

2.5 Rendering, Display, Presentation Style

Content which makes use of multimedia objects as much as an HTML document does, should have the same extent for expressing style. While HTML provides ways to encode style for 2D oriented content, only minimal attention has been given

Reference AR System Architecture

to those for 3D oriented content that subsumes and uses 2D object presentations. In addition, for purely 3D augmented objects to be more visually realistic and look indistinguishable from the physical objects, the specification of various lighting and shading effects may be important [6]. With the multimodal interaction catching on, non-visual media may be displayed in various styles too (e.g. “stereo” sound, vibration pattern).

3 SURVEY

What follows is a short survey of various AR content representation and it is the authors’ opinion that most solutions are not sufficiently comprehensive (e.g. especially lacking capabilities in vision/sensor-based AR or rich and dynamic augmentation representation), lack extensibility due to strong ties to the vendor’s own technologies or specificity to location/GPS-based AR, and often recreate new constructs (hampering interoperability) when existing standards could have been extended. While the survey is not exhaustive, we posit that most approaches probably fall within the assessment categories as described below.

3.1 KML/ARML/KARML/Junaio [7][8][9][10][11][16]

KML (Keyhole Mark-up Language) offers simple XML-based constructs for representing a physical GPS (2D) location and associating text descriptions or 3D model files to it. KML has no further sensor-related information, and thus the event of location detection (whichever way it is found by the application) is automatically tied to the corresponding content specification. KML is structurally difficult to be extended for vision-based AR (which requires a 3D scene graph-like structure) and more sophisticated augmentation can be added only in an ad-hoc way.

ARML (AR Mark-up Language) is an extension to KML and allows for richer types of augmentation for location-based AR services. KARML goes a bit further by adding even more decorative presentation styles (e.g. balloons, panoramic images), but more importantly, it proposes a method of relative spatial specification of the augmented information for their exact registration. These KML-based approaches use OGC standards [7] for representing the GPS landmarks, but for the rest, mixture of non-standard constructs, albeit being somewhat extensible (perhaps in an ad-hoc way and driven mostly by specific vendor needs), for augmentation (e.g. vs. HTML or X3D [1]).

Junaio [11] uses a custom XML file format, similar to ARML in its structure. In addition to the basic location and static augmentation, simple interactive and behavioral augmentation is possible as well (e.g. making phone calls, e-mails, and thumb-nailed viewing). Vision-based AR is also possible, but its description as a real world physical object is proprietary. We can only guess that the visual target description probably contains certain features particular to their technology.

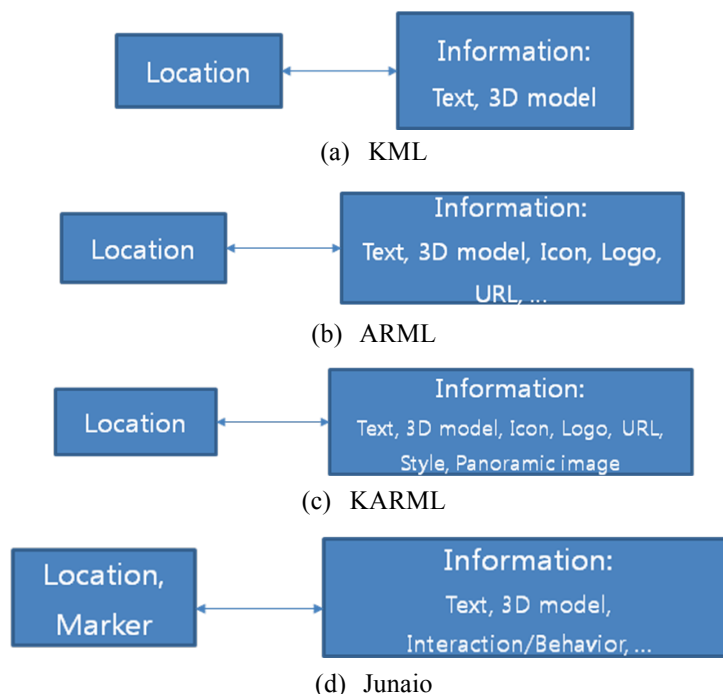


Figure 1: The basic content models of KML and its evolutions into ARML, KARML and Junaio [7][8][9][11].

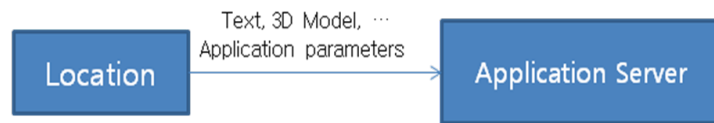


Figure 2: The content model of Layar [17].

3.2 Layar [17]

Layar [17] seems to have a similar internal content model, that is, a simple and direct association of a POI to augmentation (services) provided by networked servers (using a JSON protocol). However, the exact model structure is hidden from the user. To create a “layer” (like a content snippet), the user simply uses a web-based tool to create a POI and specify the web server supplying the augmentation service, stored in a central server to be processed by the Layar browser.

3.3 InstantReality [12]

InstantReality Suite extends their X3D-based virtual reality (VR) model to AR/MR. X3D already has a sophisticated scene graph structure and representation power for a comprehensive list of media objects [2]. The AR/MR functionality is realized by defining new “nodes” and processing the AR-enabled X3D file by their own AR browser. Among the new nodes, it is the “IOSensor” node that is the construct that generates 3D pose for the augmentation relative to the physical location of the augmentation target. One typical IOSensor might be of the type “VisionLib” (camera-based marker recognizer). However, the actual augmentation target object is described separately in the “configuration file” along with details of the particular sensor used. This means the description of the target object is separate from the main content file, plus some idiosyncratic knowledge of the sensor or recognition algorithm is needed on the part of the user.

Note that as for GPS locations, Web3D (the organization handling X3D standards) has a recent proposal for “GeoLocation” nodes for representing GPS locations and associated sensing capabilities (e.g. GeoProximitySensor), which can be easily used for location or GPS-based AR [2].

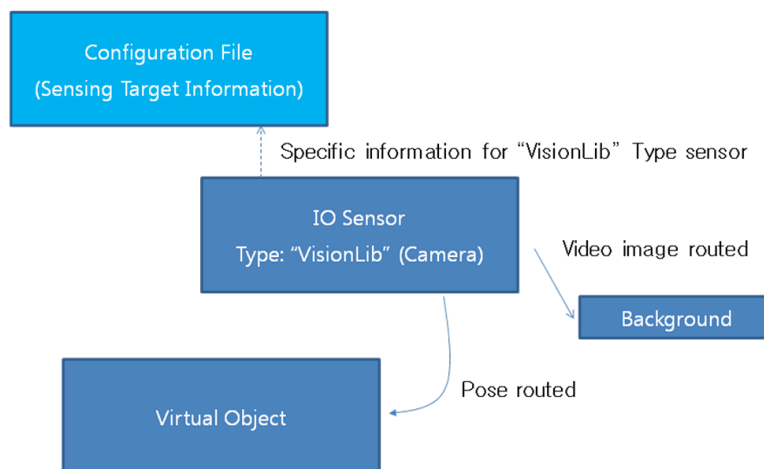


Figure 3: InstantReality’s association of a physical object to a virtual through the IOSensor specification.

3.4 K-Mart [13]

K-Mart is also a X3D-based content model in which the sensor and physical object representation are merged together. This is based on the view that a physical object can be abstracted as sensing its own pose in the real world. In addition, the sensing details is abstracted. Only the type of sensing is specified (e.g. proximity, visibility, collision, etc.) extending the already existing sensors in X3D. The physical object description is contained within the sensor description and grouped together with the associated augmentation information.

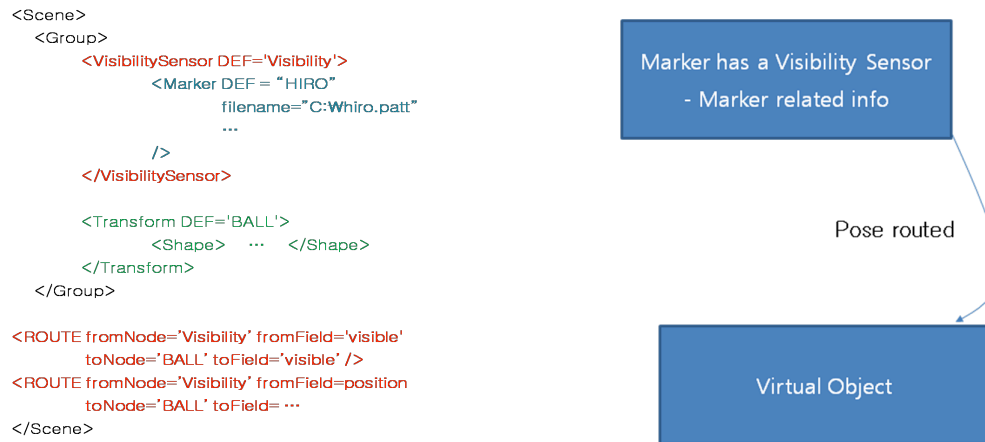


Figure 4: The K-Mart abstracts physical objects as sensor objects with a particular type (e.g. visibility sensor).

3.5 MPEG-4 [14]

MPEG-4 offers a way to specify how different media such as video, audio and synthetic graphical objects can be mixed and synchronized together. However it lacks representations for the real world objects and registration/sensing methods in the 3D space. It also does not have constructs for interaction or dynamic behavior.

3.6 RML / RWW [15]

RML (Reality Mark-up Language) used in the Microsoft “Read Write World” project offers schema-less representation of geospatial contents using the JSON notation. As seen in Figure 5, it uses text to freely add new attributes and let the browser figure out or choose what to do with them, thus not amenable to standardization. It uses Web-based protocols and the Bing search engine API [15] to retrieve and post objects using global identifiers (shown below). The keys/ids/objects returned by the protocol may be referenced and exchanged for further cloud processing. It seems the same mechanism can be used for explicitly mapping sensing/tracking events to the AR/MR contents in order to make AR content model more independent from a particular browser technology.

```

[GET] http://www.bing.com/api/rml/author/(name)
[POST] http://bing.com/api/rml/id/(#id)?key=(#key)

```

The content presentation is also not directly specified in the content itself but rather is handled implicitly by the browser or RWW application.

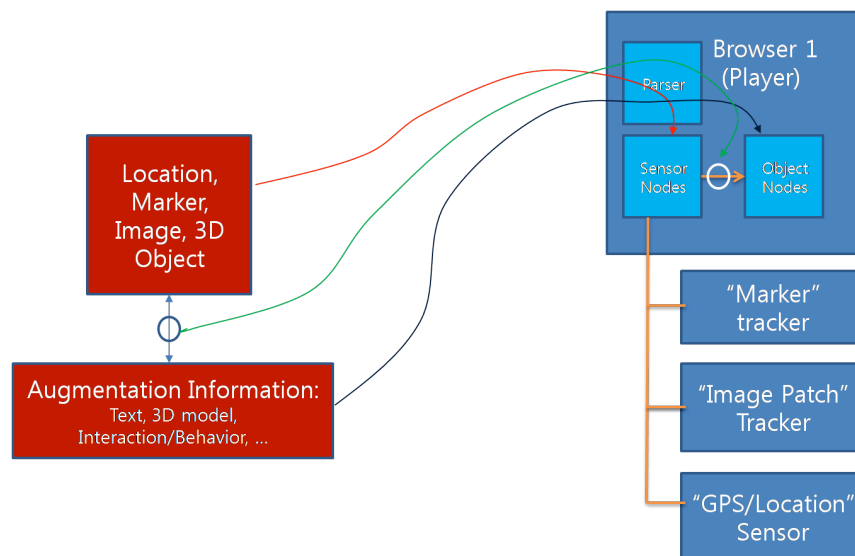
```

{
  "type": "Entity",
  "id": "12342342",
  "transform": { "worldCoord": { "lat": -122.3121, "lon": 47.234 } },
  "info": { "author": "avi", "license": 3 },
  "sources": [
    {
      "type": "Image",
      "url": "http://www.example.com/avi.jpg"
    }
  ]
}

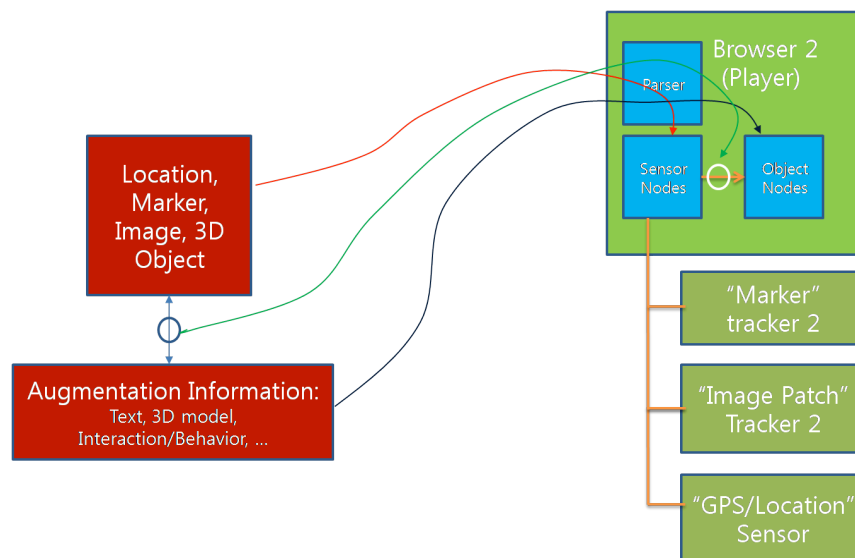
```

Figure 5: RML specification of a geospatial tag. The quoted attributes are to be guessed and interpreted by the browser without any standard semantic definition [15].

Reference AR System Architecture



A content interpreted and rendered on Browser 1



Same content interpreted and rendered on Browser 2

Figure 6: The proposed AR content model and browser architecture implemented as two different browsers with different sensing technologies.

4 DISCUSSION

Our perspective on the current status of AR content models is that they are mostly not comprehensive in their capabilities, e.g. in types of augmentation supported, interactive behavior, types of AR platforms (location based only or few vision based) and lack extensibility for future AR platforms (e.g. natural feature based, depth map based). X3D-based approach seems promising at least for providing a sophisticated world model (scene graph structure) and many media objects for augmentation. As for representation of geospatial locations, while more types of POI's need to be represented (not just single location but also paths, hierarchical POIs, area, etc.), there is no good reason not to use the mainstream standards such as those by OGC, also already adopted in the KML-like approaches. Therefore, we propose a universal AR content model as an extension of a virtual world with provisions for representing the physically-sensed objects. The provisions refer to ways to

Reference AR System Architecture

specify the physical augmentation “targets” without specific sensor information and ways to (intuitively) tie or associate them to their virtual counterparts (e.g. as shown in Figure 2, or in the left parts of Figure 6). This will result in vendor independence, use convenience and support future extensibility. Note that while the sensing details might have to be abstracted away, certain protocols will be needed to map the outputs of the sensors to the events in the AR content. The protocols adopted by RML (GET and POST) can be used for this purpose.

Then, such a provision would in turn decouple the browser architecture into two parts: one interpreting the content and transforming it into an internal abstract scene model and the other mapping the scene components to particular methods of graphic rendering or physical sensing. The resulting content model and ensuing browser architecture may be depicted as that shown in Figure 6, in which the core interpretation part is common while many different tracking and sensing technologies may be utilized for detecting the real world.

This document is a work in progress submitted at an interim point for feedback at any and all levels. The authors seek to cover, in the final version of this survey, all relevant examples available in July 2011 and would appreciate the engineers of various relevant projects providing a comparable level of information about their architectures.

ACKNOWLEDGEMENT

The research in this paper was supported in part by the Korean Agency for Technology and Standards (KATS), Korea Evaluation Institute for Industrial Technology (KEIT Strategic Technology Lab Program), and Korea Institute of Science and Technology (KIST-Kocca Project on AR-based Mobile Tour System).

REFERENCES

- [1] W3C, HTML 4.01 Specification, www.w3.org/TR/html401
- [2] Web3D, What is X3D?, <http://www.web3d.org/about/overview>
- [3] Kronos Group, Collada-3D Asset Exchange Schema, <https://www.kronos.org/collada>
- [4] Berners-Lee, Masinter and McCahill, Uniform Resource Locators (URL), <http://www.ietf.org/rfc/rfc1738.txt>
- [5] Sun Microsystems, RPC: Remote Procedure Call, Protocol Specification Version 2 (RFC 5531), <http://tools.ietf.org/html/rfc1057>
- [6] Jung, Franke, Dahne, and Behr, Enhancing X3D for advanced MR appliances. Proc. of Web3D Conference, 2007
- [7] OGC, KML/OGC, www.opengeospatial.org/standards/kml
- [8] Mobilizy, ARML Specification for Wikitude 4, <http://www.openarmr.org/wikitude4.html>
- [9] Hill, MacIntyre, Gandy, Davidson and Rouzati, Khamra-KML/HTML Augmented Reality Mobile Architecture, <https://research.cc.gatech.edu/polaris>
- [10] Butchard, Augmented Reality for Smartphones, Proc. of AR Standardization Forum, Barcelona, 2011
- [11] Junaio, your mobile companion, <http://www.junaio.com/>
- [12] InstantReality, <http://www.instantreality.org>
- [13] Choi, Y, Kim, Lee, G. Kim, Nam and Kwon, K-mart: An Authoring tool for Mixed and Augmented Reality (poster paper), Proc. of ISMAR, 2010
- [14] MPEG Industry Forum, What is MPEG-4?, <http://www.mpegif.org/mpeg4>
- [15] Bing, Read Write World, <http://www.readwriteworld.net/>
- [16] Visser, A Survey of XML Languages for Augmented Reality Content, Proc. of AR Standardization Forum, Barcelona, 2011
- [17] Layar, <http://www.layar.com>