A Proposed Step-By-Step Guide to an AR Standard

Martin Lechner

Wikitude GmbH.

Abstract

Many discussions have emerged recently from the Augmented Reality industry about creating an Augmented Reality Standard. In the rather new and thus still fairly unstructured AR industry, various institutions and groups got together and analyzed the AR landscape from the ground (the *analytical approach* to a standard), while others started from the other end and already created proposals for an AR standard (the *hands-on approach* to a standard, based on the lessons learnt from their own applications). This paper proposes a step-by-step guide to an AR standard by structuring the gathered information and combine both approaches to finally be able to define the scope of an AR standard, which eventually leads the way to define an AR standard. The step-by-step guide will be used as the starting point for the ARML 2.0 Standards Working Group[1] within the OGC[2], which will be officially established in September 2011.

1 Introduction

Ever since Mobile Augmented Reality Browsers and Applications emerged to the mass market in late 2008, many stakeholders working in the AR industry started to think about defining an Augmented Reality standard to allow distribution of AR content on multiple AR ecosystems, platforms and applications. Various working groups were established to analyze the AR environment (for instance the W3C AR Community Group [6] or the International AR Standards Working Group [1]) and initial AR standard formats were proposed (e.g. ARML 1.0 [8], proposed by Wikitude, or KARML [2], proposed by Georgia Tech) to give a first glimpse on how an AR standard might look like. Very recently, the OGC has announced the formation of the ARML 2.0 Standards Working Group, with the group's mission statement to define an officially accepted AR Standard within the OGC being an official SDO[3]. While this is a necessary activity for an industry to grow, there needs to be some kind of a step-by-step guide or roadmap while defining the standard, to ensure that the standard is not isolated within the defining SDO, but the interest of all AR stakeholders are met and the standard is actually useable within the AR industry. In the following sections, the paper will outline a proposed step-by-step guide for defining an AR Standard, with the steps headlined as follows:

1. **Stakeholder Analysis:** Identify the Stakeholders in the AR ecosystem to ensure their interests are captured.

2. **Architecture Analysis:** Analyze AR architectures from a technical point of view and extract common patterns to ensure the standard can be implemented at all and can be used by already existing AR applications.

3. **Scope Definition:** Based on the architecture and stakeholder analysis, identify the highlevel architectural parts which are in the scope, and which are out of scope.

4. **Usecase Definition:** Identify usecases which are already implemented and served in AR applications, and develop usecases which are likely to be implemented in the future (if possible). This ensures that a "measure of feasibility" is available for already existing standards to determine if they already provide required functionality.

5. **Breaking it down:** Based on the scope and usecase definition, break the scope of the standard down into smaller, more concrete parts ("Working Items") to ease further analysis.

6. **Analyze existing standards and SDOs:** For each of the identified working items, analyze already existing standards covering or influencing the working item and consult SDOs working on these standards to define the working items which are already covered by other accepted standards. Measure the feasibility of adopting the standard by utilizing the usecases.

7. **Fill the gaps:** For the items not (fully) covered by existing standards, or for those items where reusing existing standards does not make sense, fill the gaps by proposing new functionalities.

8. **Feasibility Check:** For the final outcome, check if the usecases are tackled and the stakeholders are able to implement and/or use the standard.

In this paper, steps 1, 2 and 3 are covered in more detail, with the scope of an AR standard being the outcome of the analysis in this paper. Steps 4 to 8 will be covered in the ARML 2.0 SWG and published as results arrive.

2 Stakeholder Analysis

The first step when defining an AR standard is to analyze who is part of an AR ecosystem and why, which stakeholder is influenced by a standards discussion, and how the stakeholders could benefit from a new standard. The outcome is a diagram of the involved stakeholders, together with their goals and tasks. The author's proposal of a stakeholder analysis is shown in figure 1. Starting from

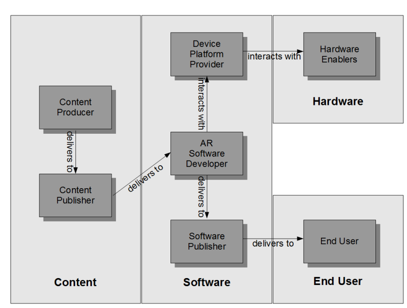

Figure 1: Stakeholders in an AR Ecosystem

the top left corner of the diagram, the stakeholders are briefly described below, a full discussion of each stakeholder is out of scope of this paper.

[1]The author of this paper is also the convener of the ARML 2.0 SWG
[2]Open Geospatial Consortium
[3]Standards Development Organization

Content Producer: The Content Producer creates content for multiple occasions, not just for delivering the content to an AR application. Real life examples include travel book writers or tourist bureaus.

Content Publisher: The Content Publisher receives content from the Content Producer and specifically publishes the content to an AR application. Examples include publishing agencies specialized in mobile publishing.

AR Software Developer: AR Software Developers provide the AR software. Examples include AR Browser vendors such as Wikitude [7] or Layar [3], as well as application developers creating single-usecase AR apps. They receive content from the Content Publisher to fill their apps with content.

Device Platform Provider: Device Platform Providers provide the platform the AR Software Developer is using to build the AR application, typically by creating operating systems and development platforms. They also provide the interface between the software and hardware. Examples include Google (Android), Apple (iOS), Nokia (Symbian) and the like.

Hardware Enablers: Hardware Enablers manufacture hardware necessary to build AR applications, including chipsets, sensors, cameras, network, battery etc. Essentially, the Device Platform Provider builds the platform on top of the hardware. Examples include Qualcomm, Nokia, HTC etc., an entire list could fill quite a number of pages.

Software Publisher: Software Publishers finally make the AR Software available to the market. The AR Software Developer delivers the software to the Software Publisher. In a mobile environment, this stakeholder is typically represented by the various application stores available for each platform (Android Market, App Store, Ovi Store, App World etc.), but it could also be a webpage advertising the AR software and making it available for download.

End User: The end user finally downloads the application from the Software Publisher.

3 Architecture Analysis

In [4] and [5], the authors defined a generic software architecture for AR applications, consisting of six subsystems which form the entire architecture (*Application, Tracking, Interaction, Presentation, Context and World Model*). While this is very valuable to describe AR architectures in general, it is beneficial to slightly modify the architecture when specifically analyzing standards for mobile AR. This holds true especially for the *Application* subsystem, which is too much of a general term for our purposes. The author's proposal for a mobile AR architecture is outlined in figure 2 and described briefly below:

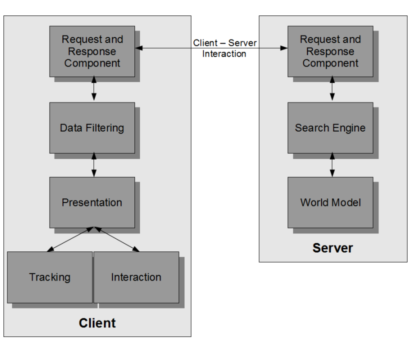

Figure 2: Proposed AR Architecture

Tracking: Tracking is the task of analyzing the real world, typically done via sensor evaluation. Sensors observe one or multiple properties in the real world and deliver an approximation of the current value of the property back to the system. Typically, tracking is a multi-step process. In a first step, the hardware is accessed and the raw values are determined, followed by (multiple) sensor value post-processing and aggregation steps. Finally, the values are made accessible to the application.

Interaction: The interaction subsystem takes care of user interaction.

Presentation: The presentation subsystem deals with the rendering part of an AR application, may it be visual, audio, haptic etc.

World Model: The world model describes the world around the user and stores information about the virtual objects and their presentations.

Request/Response Component: A component on both the server and the client side to request and deliver data which is considered for presentation.

Data Filtering: The client needs to filter data based on realtime knowledge gathered constantly on the client (location, camera video stream etc.), before they are presented in the application.

Search Engine: The server's search engine, which queries data based on certain query parameters sent by the client.

4 Scope Definition:

The scope definition can now be derived from the architectural analysis, taking the stakeholders' point of view into account as well. The proposed scope definition is outlined below:

Tracking: The final tracking step where the AR tracking result is available for the application should be in scope of the discussion, as this highly affects how the applications can interact with the underlying sensor hardware, which is crucial for AR applications. However, the underlying steps are not in scope of an AR standards discussion, as the sensor landscape is inherently diverse and influences not only AR, but many other fields as well.

Interaction: When looking at the standards the web is offering in terms of user interaction, we see that standardization thereof breaks into two parts:

- How does the user interact with the application (the triggers of events)?
- What can be executed on a certain interaction (the code being executed on occurrence of an event)?

In a HTML file, the events are defined by keywords which can be included in the file, such as onMouseOver, onClick, onKeyDown etc. The code executed is typically (but not exclusively) defined in JavaScript, which has access to any element in the HTML DOM and can modify the exposed properties of the DOM elements (for example the *innerHTML* property of a *div* tag). This functionality has proofed to be very successful and flexible (yet sometimes vulnerable to security leaks, such as cross scripting etc.) and widely used by web developers. What matters to the AR standards discussion is obviously: is it necessary for the AR community to standardize certain events and/or the code being executed on occurrence of the events? When looking at the ecosystem, various stakeholders are influenced by this decision. The Content Publisher relies almost entirely on standardizing these functionalities, otherwise, the content published by him will appear totally different, depending on the AR application the content is published on. Also the AR Software Developer is influenced by the standardization of user interaction, as he needs to comply with the standard and expose the standardized properties to the content publisher. The AR Platform Provider is influenced to a certain extent as well, as he needs to provide the platform-capabilities for the software developer to be able to comply with the standard. User Interaction in AR is – besides the advise to reuse existing concepts from the web – exclusive to AR applications and effects multiple stakeholders, thus defining standards for user interaction should be part of an AR standards discussion. Additionally, there might be parts which are also relevant to map-based applications, and it is worth considering potential reuses from or in map standards as well, where applicable.

Presentation: When comparing the AR world with the web world, presentation in the web is basically represented by the elements in the HTML DOM, as well as their specification of appearance in web browsers. The user interaction and tracking components will typically tightly interact with the presentation component. Thus, when stating that user interaction should be part of the standardization process, it immediately follows that the presentation of the virtual objects must be part of the discussion as well. Otherwise, the user interaction itself would be standardized, but it does not have any standardized components to interact with, which does not make sense. Additionally, almost every stakeholder is again influenced by how the presentation of the content is defined – essentially in the same way as they are influenced by the user interaction standardization.

World Model: The World Model describes the virtual world around the user – basically, it is a virtual representation of the physical world. It very much relates to the question which data format will be used to represent the virtual world. Basically, this is a key question to any AR application, and again, the same stakeholders as in user interaction and presentation are influenced by this question. It is obvious that discussions around how the world model can be described are essential for an AR standard.

Data Request, Data Response, Filtering and Search Engine: Requesting and delivering data basically translates to two questions:

- How should data be requested and queried, and should the process be part of the standard?
- Does AR require a special transmission format for the data response, or is it sufficient to reuse the world model data format?

Data queries are highly dependent on the used data storage, and typically, these storages already have standardized query languages (SQL for databases) or at least some proprietary query language (like Lucene's query language) which needs to be reused. There is no sense in trying to define a meta-query language matching all data storages that could be used. Data requests can also come in numerous shapes, but a typical usecase is to interact with a RESTful interface, sending parameters to the server. After all, at least crucial request parameters (such as location in geo-based AR applications) should be specified. In terms of the data response, it needs some more investigation on the world model to define whether a separate transmission format is required or not. We will leave the discussion to the point when a world model is defined, for now, we state that a separate transmission format will be kept in mind for the discussion. Finally, when looking at the filtering component, this is essentially part of the application logic. The AR software must be able to filter the virtual data (or virtual objects) in a manner that it can be presented to the user in a meaningful way. This is application-specific and does not require standardization.

5 CONCLUSION

We are now in a position to state what should be defined within an AR standard. The outcome of the scope-discussion is reflected in the following table:

Component	**Consider?**	**Why (not)?**
Tracking	YES	the interface between the tracking component and the application logic
User Interaction	YES	Many Stakeholders involved Most parts rather AR specific Potential reuses in map-based applications
Presentation	YES	required for User Interaction
World Model	YES	most basic AR standard discussion topic
Data Request	YES	certain basic standardized parameters required
Data Response	YES	depends on the World Model
Data Filtering	NO	too application and use case specific
Search Engine	NO	Well established standards already exist Too diverse

To summarize, we will consider the following areas for standardization:

- The interface between **Tracking** and the application logic to allow access to the tracking values in a standardized way.
- The **World Model** representing the underlying data model for virtual objects.
- The **Presentation** of the World Model in the virtual world.
- **User Interaction** and the control of the application workflow by the user.
- The (remote) **Data Request** and **Response**.

Finally, it is necessary to state that AR is still evolving, and it will be important to set a strong focus on the extensibility of the standard once it is finally defined. There will be more usecases coming with better devices and additional interaction capabilities, thus there must still be room for improvement within the standard as well.

6 NEXT STEPS

According to our step-by-step guide, the next step is to define AR usecases which will serve as a measure how well our new standard can support AR specific problems. The usecases shall require all the subsystems we have defined to be in the scope. In the author's opinion, there should be at least usecases specifically targeting

- Geo-based AR
- Vision-based AR and Natural Feature Recognition
- User Interaction with virtual objects
- Non-visual AR (audio, haptics etc.)

Finally, we will be able to break the scope down into small sub-parts of the standard, such as *a standard for geolocation*, *a standard for 3D models*, *a standard allowing user interaction* etc. After that, the actual work on standard definition will start with analyzing the standards ecosystem and testing existing standards against the usecases (such as KML, ARML, KARML, GeoJSON, GML etc. for geolocation, Collada, X3D, etc. for 3D models, JavaScript for user interaction). Major parts of the discussion and work will be done within the ARML 2.0 Standards Working Group, which will officially be established during the OGC meeting in Boulder, CO in September 2011.

REFERENCES

[1] Christine Perey. Homepage of the International AR Standards Group. In *http://www.perey.com/ARStandards/*.

[2] George Tech. Specification of KARML. In *https://research.cc.gatech.edu/kharma/content/karml-reference*.

[3] Layar B.V. Homepage of Layar. In *http://www.layar.com*.

[4] T. Reicher, A. MacWilliams, and B. Bruegge. Towards a system of patterns for augmented reality systems. In *Proc. STARS 2003*, pages 6–11, Tokyo, Japan, October 7 2003.

[5] T. Reicher, A. MacWilliams, B. Bruegge, and G. Klinker. Results of a study on software architectures for augmented reality systems. In *Proc. ISMAR 2003*, Tokyo, Japan, October 7–10 2003. IEEE.

[6] W3C. Homepage of the W3C AR Community Group. In *http://www.w3.org/community/ar/*.

[7] Wikitude GmbH. Homepage of Wikitude. In *http://www.wikitude.com*.

[8] Wikitude GmbH. Specification of ARML 1.0. In *http://www.openarml.org/*.