[POSTER] Authoring Tools in Augmented Reality: An Analysis and Classification of Content Design Tools

Roberta Cabral Mota*
University of Calgary

Rafael Alves Roberto†
Voxar Labs
Informatics Center of UFPE

Veronica Teichrieb‡
Voxar Labs
Informatics Center of UFPE

Abstract

Augmented Reality Authoring Tools are important instruments that can help a widespread use of AR. They can be classified as programming or content design tools in which the latter completely removes the necessity of programming skills to develop an AR solution. Several solutions have been developed in the past years, however there are few works aiming to identify patterns and general models for such tools. This work aims to perform a trend analysis on content design tools in order to identify their functionalities regarding AR, authoring paradigms, deployment strategies and general dataflow models. This work is aimed to assist developers willing to create authoring tools, therefore, it focus on the last three aspects. Thus, 19 tools were analyzed and through this evaluation it were identified two authoring paradigms and two deployment strategies. Moreover, from their combination it was possible to elaborate four generic dataflow models in which every tool could be fit into.

Keywords: Augmented reality, authoring tools, content design tools.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; D.2.2 [Software Engineering]: Design Tools and Techniques—Software libraries

1 Introduction

Recently, Augmented Reality (AR) technology started to be widely used in various application domains, such as advertising, medicine, education, robotics, entertainment, tourism, and others. However, the time and technical expertise needed to create AR applications is one of the reasons that has prevented widespread use. In this sense, authoring tools have become a largely used solution to boost mainstream use of AR since they facilitate the development of AR experiences [20].

AR authoring tools can be classified according to their programming and content design characteristics from low to high-level. Therefore, distinctive authoring approaches have different abstraction concepts and interface complexity, and hence address audiences that do not necessarily have the same technical expertise [8].

In this sense, development tools for building AR applications can be broadly organized into two different approaches: AR authoring for programmers and non-programmers. In the former case, tools are typically code libraries that require programming knowledge to author the application. In this work, this approach is called programming tools. In the latter case, abstraction is added and low level programming capability is removed or hidden. Thus, tools for non-programmers are content driven and commonly include graphical user interfaces for building applications without writing any lines of code. Here, it is addressed as content design tools.

These two generic categories can be further organized into low-level and high-level. Low-level programming tools require low-level coding while high-level ones use high-level libraries. Furthermore, low-level content design tools demand scripting skills and high-level tools use visual authoring techniques. All of these authoring approaches are built upon each other. Abstraction is gradually added and low-level functionalities and concepts are removed or hidden. Also, different abstraction levels address different target audiences with different technical expertise.

Among the approaches of AR authoring tools, it is important to note the relevance of the content design tools. They are particularly relevant because they leverage the widespread adoption of AR, since they highly simplify the authoring process and allow the development of applications and content by ordinary users, which do not need to have programming knowledge. Therefore, content design tools in AR have greater relevance when we take into account the potential amount of users that can use AR solutions in the future.

From the first solutions [9] to the most recent ones [3], it is possible to see that several content design tools have been developed. However, there are a few works aiming to provide a classification or to identify tendencies and patterns for such tools. To the best of the authors knowledge, there is only one work that proposes a taxonomy for AR authoring tools. In turn, this classification is according to the application interface and concept abstraction [8], in which the most abstract application interfaces are named content design frameworks and the least abstract are called programming frameworks.

Due to the relevance of content design tools, this paper aims at conducting a trend analysis in order to understand the current tendencies of such tools. It was also performed a deeper inspection regarding the functionalities they provide, such as the type of trackables these tools support or the kind of resource it is possible to use as augmented content. Since this work is aimed at developers who want to create an authoring tool it is focused on development aspects. Thus, this investigation attempts to identify the current tendencies regarding the authoring paradigms and deployment strategies of AR experiences that have been used in both commercial and academic realms. Furthermore, these strategies were combined to elaborate generic dataflow models in which all of the content design tools could fit into. Finally, based on these findings, it was introduced a taxonomy representing the different authoring and deployment trends, as well as each of the general models. That classification may guide researchers and companies to develop solutions aiming at their needs.

This work is organized as follows: Section 2 describes the methodology used to perform the trend analysis. Section 3 presents the results obtained from the analysis and the conclusions of this work are drawn in Section 4.

2 Methodology

Three steps were taken to explore the trends regarding the AR authoring tools. The first was the selection of content design tools available in the marketplace and literature. Then, as a second step,

*e-mail: roberta.cabralmota@ucalgary.ca
†e-mail: rar3@cin.ufpe.br
‡e-mail: vt@cin.ufpe.br

978-1-4673-7660-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMAR.2015.47

the analysis relied on observing the dataflow of development and access to the AR content of each of the selected tools. Finally, the third step consisted in discover the authoring paradigms and deployment strategies used in the content design tools. Their combination was used to elaborate general dataflow models.

2.1 Selection

Initially, the keywords and expressions *("authoring tool" AND "augmented reality")* were searched in IEEE Xplore Digital Library and ACM Digital Library in order to find relevant papers concerning authoring tools in AR. That allowed for an investigation over important publications from 2001 to 2014. During this examination, only authoring tools classified through content design tools were selected for a deeper analysis.

2.2 Analysis

Following this, a deeper analysis was performed on each one of the selected authoring tools in order to understand the dataflow for development and access to the AR content. On a high level, this dataflow describes the end-to-end scenario that outlines the authoring and deployment of AR experiences, from the creation of AR semantic through authoring tools to its visualization by end-users.

2.3 Categorization

The deeper analysis performed on each of the selected content design tools made possible the observation of trends regarding authoring and distribution strategies of AR experiences that have been used. Furthermore, this observation also tried to understand (a) how the different authoring paradigms may support AR content development, and (b) how the deployment strategies seek to reach a larger number of end-users. Finally, the identification of AR authoring and deployment trends allowed the translation of the project-specific dataflow, observed in the selected content design tools, into the creation of general dataflow models. In this sense, a minimum number of combinations of trends was performed in order to elaborate generic models, in which all of the content design tools could fit into.

3 Results

The search on the scientific libraries returned 134 papers and 13 works about content design tools were selected for analysis. Moreover, 6 commercial tools that are well consolidated or relevant in the market were chosen. Thus, taking into account both academic and commercial realms, there were culled 19 content design tools. Thereafter, a dataflow analysis was performed in each of the selected tools.

3.1 AR Authoring Paradigms

Once individual analyzes were performed in each of the previous selected content design tools, it was observed that two authoring paradigms have been used to create AR solutions: stand-alone and AR plug-in approaches.

3.1.1 Stand-Alone

Stand-alone augmented reality authoring tools are software with all the necessary components for the development of complete AR experiences, as can be seen in Fig. 1 (a). In turn, these components may include a graphical user interface, a series of importers, sensor interfaces, tracking and rendering engines, among others. In this sense, each stand-alone content design tool is a new software that allows designers to create their custom AR projects with more or less ease.

As an example, the Layar Creator [13] provides a complete set of features along the entire creation workflow, such as graphical user interface including drag and drop to ease the scenario creation.

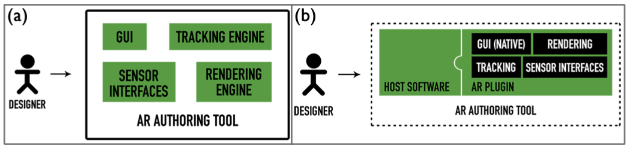

Figure 1: (a) Stand-alone AR authoring tools enable building entire AR experiences. In order to provide AR capabilities, these tools integrate components such as sensor interfaces, tracking and rendering engines; (b) AR plug-ins provide AR functionalities for non-AR authoring environments. The designer interacts directly with the hosting software in order to create AR experiences.

3.1.2 AR Plug-in

Similar to conventional digital plug-ins, AR plug-ins are third-party software components installed on host applications in order to enable additional features, as illustrated in Fig. 1 (b). In this sense, these authoring tools provide AR capabilities to software that natively does not support it, such as tracking techniques, access to physical sensors, three-dimensional rendering engine, among others.

It is relevant to note that, from the practical point of view, an AR plug-in instance will appear in the target software as a set of GUI elements, such as one or more menu items and toolbar buttons. Therefore, the whole AR authoring process occurs within the hosting environment, as the designer completely configures the desired AR experience by means of those elements along with the ones already provided by the target software.

As an example, the DART [5] system is built as a collection of extensions to the Adobe Director [2], a widely used environment for multimedia content creation, to support the development of a variety of AR applications.

3.2 AR Deployment Strategies

It was noticed that two deployment strategies have been applied to make these AR experiences available for end-users: platform-specific and platform independent methods.

3.2.1 Platform-Specific

In the platform-specific (PS) approach, AR projects built through authoring tools are exported to archive files to be independently distributed. Some common archive file formats include .exe in Windows, .dmg or .app in Mac OS, .apk in Android, and .ipa for iOS operating systems. Note that these archive files are software packages used to distribute and install native application software. A native app, in turn, is considered a stand-alone program itself since it is a self-contained program that does not require any auxiliary software on which must be executed, as can be observed in Fig. 2 (a). Native apps are usually available through application distribution platforms, such as App Store, Google Play, and Windows Phone Store. However, they must be downloaded from the platform to the end-user devices, such as iPhone, Android, Windows phones, or even laptops or desktop computers.

As an example, the Wikitude Studio [22] supports deployment options to mobile applications for iOS/Android platforms, and to executable programs for Windows/Mac OS computers.

3.2.2 Platform-Independent

The platform-independent (PI) approach delivers the AR solutions as data files read and executed on a software platform (SP) running on the end-user device. Also, it is worth pointing out that, after the authoring process, the generated content requires a platform on which it must be executed. Therefore, the content cannot be considered a stand-alone program. Rather, it comprehends data files

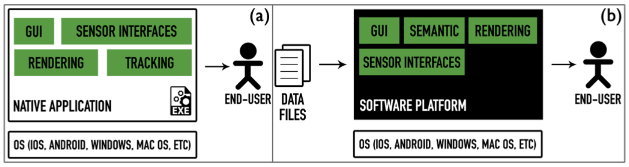

Figure 2: (a) A native app includes all required elements to execute AR experiences, thus can be considered a stand-alone software itself. Also, the term native comprises applications compiled at runtime, such as an Android app, or precompiled executable programs; (b) Data files are interpreted by a native shell, which provides the required infrastructure to present AR experiences.

(commonly structured in XML-based formats) that are interpreted by the software platform, as illustrated in Fig. 2 (b).

As an example, k-MART [4] allows designers to export AR solutions as X3D-based data files. In turn, these files are later executed on a separate content browser.

Furthermore, since the content does not need to be installed in the device, a major advantage is the possibility of implementing a cloud-based deployment service. This increasingly popular variant approach uses a server infrastructure as a backbone. The remote server is responsible for content storage and retrieval as requested by the clients. The clients, in turn, are responsible for presenting the retrieved content on end-user devices. Also, a client comprehends a cloud-based software platform that reaches into the cloud for contents on demand. In turn, all data files remotely accessed are here referenced as cloud-based applications.

As an example, AR companies like Layar and Wikitude developed Layar App and Wikitude World AR browsers, respectively. To the end-user, an AR browser looks very similar to a typical native app: it is downloaded from an app store, stored on the mobile device, and launched just like a native app. However, the most prominent advantage of AR browsers is that end-users need only one app for multiple contents. Once installed, they pull new cloud-based apps on demand.

3.3 General Models

Given the authoring and deployment trends explored in the previous subsections, it was possible to elaborate four general dataflow models that represent the content design tools' dataflow analyzed in this work.

3.3.1 Model 1: Stand-Alone PS Model

As can be observed in Fig. 3 (a), this dataflow model embodies a stand-alone authoring approach combined with a native distribution strategy. In this sense, the designer first creates AR experiences through stand-alone content design tools. Then he exports the project as platform-specific files, which are used to deliver standalone, native applications for Android, iOS, Windows or other operating system.

3.3.2 Model 2: Stand-Alone PI Model

Similarly to the previous model, the designer first builds AR experiences through stand-alone content design tools. However, these models greatly differ in their content delivery services, which apply a platform-independent strategy for reaching interoperability and maintainability. In this sense, the designer exports the authored AR solutions as data files that run on a separate software platform. Note that, a content design tool can generate one or more data files which can be interpreted and run in a single software platform and in different periods of time, as seen in Fig. 3 (b). Yet, since each standalone content design tool is a brand new software, the data files created by distinct tools generally differ in their structures, logics, and formats.

3.3.3 Model 3: All-in-One Model

As illustrated in Fig. 3 (c), in this model, both designers and end-users utilize the same environment to build and access AR solutions. In the sense, the designer creates and saves AR solutions as data files. Eventually, these files are read and executed within the same environment in order to present the AR experience to end-users. Hence, similarly to the previous model, the all-in-one comprehends a stand-alone authoring approach combined with a platform-independent distribution. However, the major difference resides in the fact that production and delivery services are merged within a single system.

3.3.4 Model 4: AR Plug-in PI Model

In this dataflow model, the designer first builds AR projects through hosting software integrated with AR plug-ins. Then, these projects are saved as data files that are later retrieved and executed on a separate software application. In other words, this model includes a plug-in approach combined with a PI deployment strategy, as can be observed in Fig. 3 (d).

All the content design tools that were selected and analyzed in this work are listed in Table 1. The table divides the commercial and academic tools and indicates to which of the four general dataflow models each tool belongs. It is important to keep in mind that it is not mandatory for a tool to be categorized into only one general model since a content design tool can provide different distribution approaches and, consequently, different dataflow models.

Content Design Tools	Year	M1	M2	M3	M4
Metaio Creator [18]	-	🟩	🟩		
Metaio AR Creator Plugin [17]	-				🟩
Wikitude Studio [22]	-	🟩	🟩		
Layar Creator [13]	-		🟩		
Build AR [10]	-		🟩		
AR-media Plugin [11]	-				🟩
Powerspace [9]	2002		🟩		
Authoring Wizard [23]	2003		🟩		
AMIRE [1]	2004		🟩		
DART [15]	2004				🟩
CDT1 [14]	2004			🟩	
ComposAR [21]	2008			🟩	
VREditor [16]	2009		🟩		
ARBookCreator [6]	2009			🟩	
AR Scratch [19]	2009			🟩	
k-MART [4]	2010		🟩		
CDT2 [12]	2012		🟩		
AVATAR [7]	2012				🟩
CDT3 [3]	2013		🟩		

Table 1: Classification of each commercial (without year) and academic tool according to the general dataflow models.

4 Conclusion

AR authoring tools can provide several levels of abstraction, thus targeting audiences within a range of different technical expertise. Particularly, those categorized as content design tools allow non-technologists to explore the AR creation medium and, therefore, these tools are an essential component for helping AR to gain popularity in different application domains. Due to their relevance, several content design tools have been developed recently however, no work was found that presents an analysis and classification of those tools.

In this sense, this work analyzed 19 commercial and academic content design tools in order to identify tendencies of such tools.

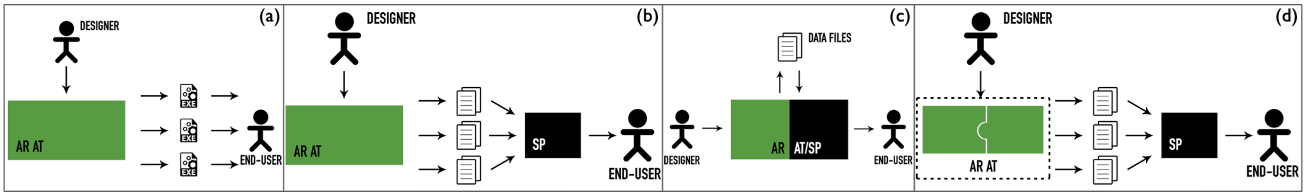

Figure 3: (a) Combines a stand-alone authoring with a platform-specific distribution. Therefore, each generated native application is individually installed and accessed by end-users; (b) Unites a stand-alone authoring paradigm with a platform-independent distribution; (c) Combines a stand-alone authoring with a platform-independent distribution. Yet, both designers and end-users utilize the same ambient to create and visualize AR solutions; (d) Merges an AR plug-in authoring with a platform-independent distribution.

The investigation revealed that there are two authoring paradigms and two distribution strategies that have been widely used for such tools. The paradigms were named stand-alone and AR plug-in, while the strategies were named platform-specific and platform-independent. Furthermore, these authoring and deployment trends were combined to elaborate four generic dataflow models: stand-alone platform-specific, stand-alone platform-independent, all-in-one, and AR plug-in platform-independent. Therefore, it was introduced a taxonomy representing the different authoring and deployment trends, as well as each of the general models. Note that, these models were called "general" since each one of the analyzed content design tools can be classified as at least one of these four dataflow models.

As major findings by comparing one trend against its alternative, it was possible to see that the plug-in approach offers more features than the stand-alone since it provides existing features in the host software. However, bringing AR capabilities to non-AR authoring tools is neither straight forward nor trivial. As for deployment strategies, the PS approach requires the development of one app for each platform, which is more expensive. Moreover, native apps have a higher maintenance cost since they require an update for newer OS versions. On the other hand, they are usually faster because the absence of an abstraction layer and depending on the nature of the app, no internet is required after the download since it remains installed on the device.

Regarding the classification of the content design tools according to the four general dataflow models, it was possible to see that most of the content design tools are stand-alone platform-independent. Only two system uses a platform-specific strategy for deployment, and there is a trend towards using descriptive data formats and cross-platform solutions in order to reach more users and, thus, to leverage the growth of AR market. While none of the commercial content design tools use the all-in-one model, it is the second most used model on academic realm. This is because academic tools have different purposes when compared with industrial software development, in which they are often a manner to showcase and promote the experimentation of newly research concepts rather than developing software for commercial use.

The authors believe that the proposed taxonomy can guide researchers and companies to develop solutions aiming their needs. This analyzes helped the authors on the development of their own authoring tool. Future works will extend this classification with a deeper analysis regarding AR functionalities, such as the type of trackables each tool supports and what kind of content they allow. This information will also help users to find the best tools for them.

References

[1] D. F. Abawi, R. Dörner, and P. Grimm. A component-based authoring environment for creating multimedia-rich mixed reality. In *EGMM 2004*.

[2] Adobe Systems Incorporated. Adobe Director. `http://goo.gl/QnkQni`, jan 2015.

[3] J. Barbadillo and J. R. Sánchez. A web3d authoring tool for augmented reality mobile applications. In *Web3D 2013*.

[4] J. Choi, Y. Kim, M. Lee, G. Kim, Y. Nam, and Y. Kwon. k-MART: Authoring tool for mixed reality contents. In *ISMAR 2010*, pages 219–220, Oct 2010.

[5] M. G. Coleman. *Creating augmented reality authoring tools informed by designer workflow and goals*. PhD thesis, Georgia Institute of Technology, 2012.

[6] T. V. Do and J. W. Lee. Creating 3d e-books with ARBookCreator. In *ACE 2009*.

[7] G. Fei, X. Li, and R. Fei. AVATAR: Autonomous visual authoring of tangible augmented reality. In *VRCAI 2012*.

[8] A. Hampshire, H. Seichter, R. Grasset, and M. Billinghurst. Augmented reality authoring: Generic context from programmer to designer. In *OZCHI 2006*.

[9] M. Haringer and H. Regenbrecht. A pragmatic approach to augmented reality authoring. In *ISMAR 2002*, pages 237–245, 2002.

[10] HIT Lab NZ. BuildAR Pro. `http://goo.gl/1LtXvt`, jan 2015.

[11] Inglobe Technologies S.r.l. AR-media - AR-media Products. `http://goo.gl/TbLeM8`, jan 2015.

[12] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg. Sketching up the world: In situ authoring for mobile augmented reality. *Personal Ubiquitous Comput.*, 16(6):623–630, Aug. 2012.

[13] Layar. Layar creator. `http://goo.gl/rQ0P1P`, jan 2015.

[14] G. Lee, C. Nelles, M. Billinghurst, and J. Kim. Immersive authoring of tangible augmented reality applications. In *ISMAR 2004*, pages 172–181, Nov 2004.

[15] B. MacIntyre, M. Gandy, S. Dow, and J. D. Bolter. DART: a toolkit for rapid design exploration of augmented reality experiences. In *UIST 2004*.

[16] N. Mavrogeorgi, S. Koutsoutos, A. Yannopoulos, T. Varvarigou, and G. Kambourakis. Cultural heritage experience with virtual reality according to user preferences. In *CENTRIC 2009*, pages 13–18, Sept 2009.

[17] Metaio GmbH. metaio — AR Creator Plugin — Products. `http://goo.gl/Z5E9ym`, jan 2015.

[18] Metaio GmbH. metaio — Creator Overview. `http://goo.gl/6xwgyF`, jan 2015.

[19] I. Radu and B. MacIntyre. Augmented-reality scratch: A children's authoring environment for augmented-reality experiences. In *IDC 2009*.

[20] H. Ramireza, E. G. Mendivila, P. R. Floresa, and M. C. Gonzalez. Authoring software for augmented reality applications for the use of maintenance and training process. In *International Conference on Virtual and Augmented Reality in Education*, pages 189–193, Oct 2013.

[21] H. Seichter, J. Looser, and M. Billinghurst. ComposAR: An intuitive tool for authoring ar applications. In *ISMAR 2008*, pages 177–178, Sept 2008.

[22] Wikitude GmbH. Wikitude Studio - the world's easiest augmented reality creation tool. `http://goo.gl/Lr0uOK`, jan 2015.

[23] J. Zauner, M. Haller, A. Brandl, and W. Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. In *ISMAR 2003*, pages 237–246, Oct 2003.