ICAT 2004

Augmented Real World (ARW) Framework and Augmented Reality Modeling Language (ARML)

Kei Nishimura and Yoichi Muraoka
Muraoka Laboratory, Waseda University, 3-4-1, Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
nishimura@muraoka.info.waseda.ac.jp

Abstract

This paper describes how to construct an augmented real-world (ARW) framework for modeling an augmented real (AR) world and proposes an augmented reality modeling language (ARML) that can easily describe the AR world. ARW Framework enables virtual and real worlds to be combined using AR techniques. (In this paper, the term "virtual world" implies both the Web and the usual virtual reality (VR) world. "Real world" is where humans live and it is extended by data in the virtual world. The combination of the real and virtual worlds is defined as "AR world"). ARML is the bridge between the virtual and real worlds and enables users to flexibly and easily describe linkages between both worlds at the same level as HTML. For example, it is possible to embed virtual objects in the real world, control objects according to user's behaviors, and link real objects and Web contents. Once ARW Framework has been constructed, it is possible to create an Intelligent Character Agent as an application over ARW Framework. This is the goal of this research. The application aims to provide autonomous support that offers appropriate information to human users by interacting with them.

Key words: AR, framework, ARML, VRML, agents

1. Introduction

One new virtual world "Web" is being born while the foundation of global transmission of information called Internet is being established. Technology that combines the virtual and real worlds has developed in recent years as a result of mobile, wearable, and ubiquitous computing. We can now access the network without being limited by place or time and can obtain useful information.

The virtual world usually means the closed world that is represented by computer graphics (CG) and called virtual reality (VR) in that field. The limitations of CG's representational power led to the idea of utilizing and augmenting the virtual world with data from the real world. It is one of the remarkable research domains. On the basis of the idea, techniques that can visually extend the real world by overlaying it with computer-generated data are called augmented reality (AR) techniques.

Although these techniques are different when seen from the viewpoint of each field, they are the same in that they aim to unit the virtual and real worlds—the only difference is the type of data used for the enhancement.

An enormous amount of information has been accumulated on the Web, which is a well-established source of information about human life. Advances in Web search technology have made it relatively easy to retrieve valuable information. The visualization of such information by AR techniques has the potential to enrich the real world.

In this paper, the virtual world implies both the Web and the VR world. The real world is where humans live and it is extended by data in the virtual world. The combination of the real and virtual worlds is defined as the augmented real (AR) world.

This paper describes how to construct an ARW Framework for modeling the AR world and proposes an augmented reality modeling language (ARML) that can easily describe AR world. ARW Framework provides the framework for combining the virtual and real worlds using AR techniques. Figure 1 shows an overview of ARW Framework.

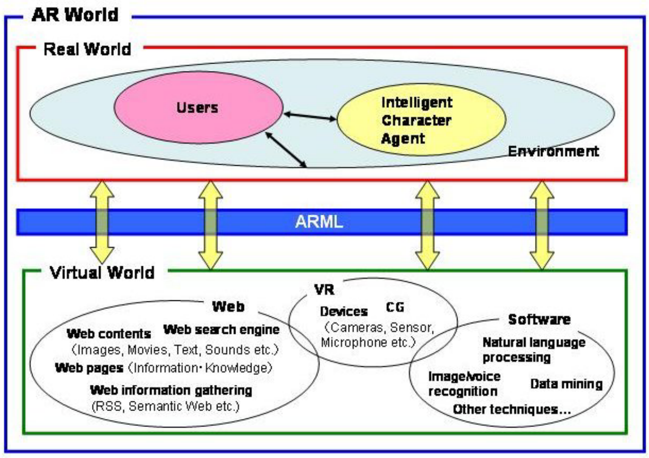


Fig. 1: Overview of ARW Framework

ARML is positioned as a next-generation modeling description language after VRML (virtual reality modeling language), which was much discussed in the latter half of the '90s. VRML is a language that can construct an interactive 3D virtual space on the Web. ARML, on the other hand, can construct one on the real world.

ARML has the potential to be a common interface for the virtual and real worlds. There are many techniques such as image/voice recognition and natural language processing that can obtain and analyze data from the virtual world, from the Semantic Web or RSS (RDF site summery) as web information gathering methods and hardware for doing this includes data gloves, gyro-sensors, and cameras. However, although these techniques are sophisticated, they are not coordinated because they are implemented using various programming languages. To enable techniques to be easily used and combined, the programming language should be standardized or a unique common interface should be prepared.

ARW Framework introduces agent technology [1][2][3] to solve the problem. This handles communication with other software or people in place of the user. Positioned as a new type of software, it has three features: autonomy, mobility, and cooperativeness. Agent technology creates agents for various techniques. Agent mobility makes it possible to integrate distributed techniques at any place and agent cooperativeness makes it possible for agents to communicate with each other through a common protocol. ARML provides such a common interface with the virtual world.

The ultimate goal of ARML is to create a bridge between the virtual and real worlds. The description format of ARML is XML. ARML lets users flexibly and easily describe linkages between both worlds at the same level as HTML. For example, it is possible to embed virtual objects in the real world, control objects according to users' behaviors, and link actual objects and Web contents.

Once ARW Framework has been constructed, it is possible to create Intelligent Character Agent as an application over ARW Framework. This is the goal of this research. Intelligent Character Agent is an animated CG character (like Tinkerbell in the book Peter Pan) that can infer the user's thoughts from situational elements in the real world through ARW Framework. It also has Web information about the virtual world as knowledge. The purpose is to provide autonomous support that offers appropriate information to human users by interacting with them. The system integrates and utilizes the various agents. Intelligent Character Agent over ARW Framework provides a new common human interface to the virtual world. Figure 2 shows the concept of Intelligent Character Agent.

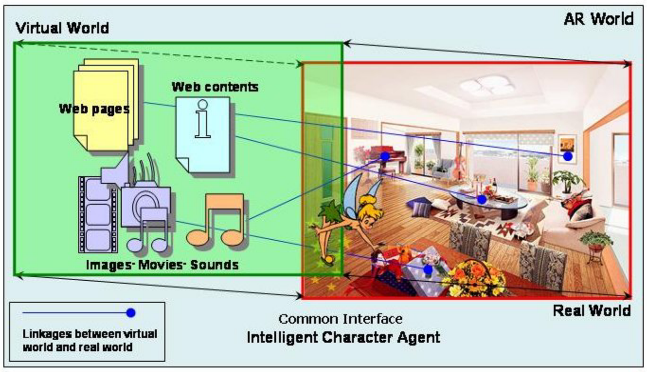


Fig. 2: Concept of Intelligent Character Agent

2. Related Works

2. 1 Augmented Reality (AR)

Generally, augmented reality (AR) enables users to interactively experience worlds that are created by overlapping virtual data on live video from cameras through head mounted displays (HMD).

In this field, there are some display systems that use impressive projection technology (IPT) like CALVIN [12] in comparison with HMDs. These systems can generate virtual worlds by projecting 3D stereoscopic vision on huge multifaceted screens using high-resolution projectors. However, they need to use huge equipments, and are limited by place. Users have to be in the space.

While, HMDs enable seeing through first-person perspective. Because users can wear the display on their head and move around with it, HMDs are not limited by space. In other words, whole space of the real world can become a display unit. Consequently, the HMD is suitable for our system. However, HMDs are not so popular because of a lack of resolution, silly-looking, and wearing difficulty. In fact, there are a few manufacturers that provide HMDs. Though, a glasses-styled see-through display called holographic see-through display that was developed by Konica Minolta Japan received a prize, Best of WPC EXPO 2003 in Japan. Display device technology has potential for advancement. We believe we will be able to easily wear the display like general glasses.

2.2 VRML/X3D

It is said that VRML released in 1997 is the first 3D computer graphics (3DCG) on the web. VRML developed by Silicon Graphics, Inc. did not become widely used at that time because the hardware and software for creating VRML was expensive, we had to use special browsers to load VRML, and the resolution of VRML was low. However, VRML came back in 2000 as X3D (Extensible 3D) by Web3D Consortium [11] (former VRML Consortium). X3D is described by XML in order to solve above-mentioned problems, and is said to be next-generation VRML.

X3D is useful in terms of affinity with ARML. However, it is a new technology, and there are a few X3D contents in the world. It will take a long time to become common. We see how it is going now. Although VRML was not able to become so popular, most of major 3DCG rendering software can import VRML, and we can obtain a number of VRML contents from the web.

We review the VRML technique that is intimately related to Java3D. Java3D is used as the implementation of ARW Framework. VRML has two versions; VRML1.0 and VRML2.0 (also known as VRML97). VRML2.0 is widely different from VRML1.0, and both are not compatible. VRML2.0 is more popular than VRML1.0. VRML2.0 was standardized internationally by ISO (International Organization for Standardization) and IEC (International Electro-technical Commission) in 1998 (ISO/IEC 14772-1). VRML1.0 enables users to only see static objects. VRML2.0 additionally includes following functions [6][7].

1. Animated 3D objects
2. Handling multimedia (sound or video)
3. Mouse interaction
4. Control by other language (Java or JavaScript)

Function 4 is the most noteworthy in listed above functions. It is called JASI (Java Script Authoring Interface). JASI can carry actions described by Java to objects through script nodes. We can flexibly configure the interface between VRML and Java. This know-how is a formative guide to construct ARW Framework.

2.4 Related Research

There are two kinds of related research. One is it3d (Interactive Toolkit library for 3D applications) developed by NIME (National Institute of Multimedia Education) [9] and the other is MR (Mixed Reality) platform system developed by Canon [10].

It3d is a toolkit library for creating artificial reality applications. It is welcome to use for education, research, and prototypes. The input/output library for distributed devices is more useful than other sub-libraries in it3d. It has a function to exchange events and data from various types of devices through integrative and effective methods. There are many kinds of devices in the artificial reality systems. Traditional applications depend on devices, and individually used to support each device. The library has a model to divide devices from applications or to integrate both on the network level, and enables changing applications without any configurations even when there are many different devices in systems. This know-how is also useful for ARW Framework.

MR platform system is a common infrastructure for creating various mixed reality applications. It includes a see-through HMD and software libraries called MR platform SDK. The libraries especially include a basic library for positioning between virtual and real space. MR platform system covers middle and lower layers of applications. Users need to develop higher-level layers of applications. ARW Framework lies on the higher layer, and will be able to use Canon's libraries through ARML.

3. Specifications of ARW Framework

ARW Framework is implemented using Java and Java3D [4][5] class libraries. Java3D calls native application programming interfaces (APIs) of OpenGL or DirectX to draw graphics and can use the extensive range Java libraries including network-related ones. It is possible to define the space and easily deploy 2D and 3D objects at specific spatial coordinates and operate (rotate, shift, and scale) those objects using a mouse. Java3D supports the development of ARW Framework.

3.1 ARW Framework

ARW Framework can be divided into three classes of functions, as shown in Figure 3: ARW Agents gather virtual and real world information using agent technology, ARW Objects are contents representing 3DCG like VRML whose motion in the real world can be controlled and ARW Core has an ARML parsing function that interprets a linkage between the virtual and real worlds.

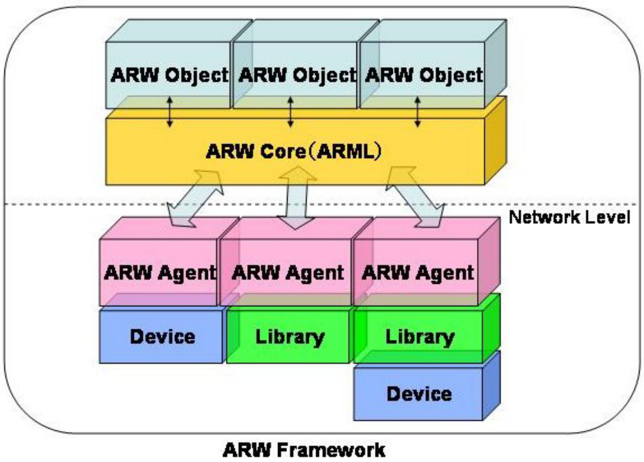


Fig. 3: Function Classification of ARW Framework

3.2 ARW Agents

ARW Agents are implemented by agent technology based on the model that devices and applications should be separated and integrated at the network level and application structures need not to be changed over any devices. Each agent can act automatically, move to any host, and communicate with other agents. ARW Agents support not only devices but also software that provides image/voice recognition, natural language processing, and web information gathering techniques etc. The lower-level part of each agent can be implemented using various programming languages. Programs in the upper-level part can call methods of remote objects as well as local objects by the Java RMI (Remote Method Invocation) technique. Any applications anywhere can access the agents at the network level and use resources in the hosts.

3.3 ARW Objects

Contents including 3DCG like VRML that can be managed in the real world are called ARW Objects. ARW Objects support various 3DCG contents including VRML, 2D images, sounds, movies, and other multimedia contents. It is possible to load these contents through 3DFileLoader as a Java3D standard function. Support for ARML is especially enhanced. It is possible to accept non-support 3DCG contents by creating original 3DfileLoaders. ARW Objects has an interface equivalent to JSAI (Java Script Authoring Interface) of VRML. It is possible to access all attributes of objects through ARML. ARW Objects attributes include geometry, appearance (material and texture), behavior and animation. The motion of the ARW Objects can be described by ARML.

3.4 ARW Core

ARW Core is the heart of ARW Framework. It provides the interface between the virtual and real worlds. Its main function is an ARML parser. ARML can describe linkages between ARW Objects and ARW Agents. ARW Core interprets the meanings of the ARML motion. ARML is written in XML format, and ARW Core includes an XML parser. ARML is also a completely object-oriented language, and ARW Objects and Agents are equivalent to objects. It is possible to call methods of each object, define space, set light sources, and program the behaviors.

4. Prototype System

We developed a prototype system for ARW Framework. It was implemented on Windows OS by Java that includes Java3D class libraries. Figure 4 shows the interface of the system.

Fig. 4: Prototype system

The current system has five functions as follows.

1. User view image
2. User view direction
3. User position
4. Object position and animation

The system consists of three ARW Agents, two ARW Objects, and an ARW Core in relation to each function as shown in Figure 5.

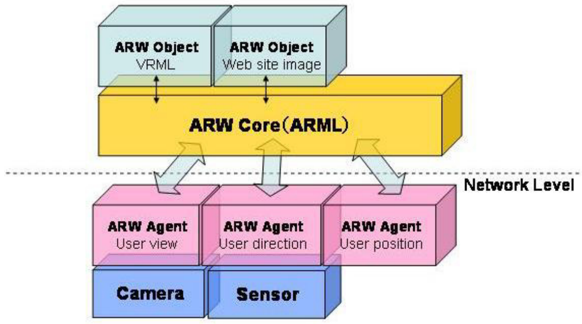


Fig. 5: The system structure

4.1 ARW Agents

We created an ARW agent to obtain user view images (Function 1). It was implemented by JMF (Java Media Framework) that can capture images from devices like USB cameras as shown in Figure 5 Left.

We also created an ARW agent to obtain user view direction (Function 2) using 3D motion sensor developed by NEC Tokin as shown in Figure 5 Right (the sensor is installed to a HMD). The direction data means roll, pitch, and yaw. The APIs that obtain the data are implemented by C language. The ARW agent uses JNI (Java Native Interface) to use the sensor.

We created another ARW agent to obtain user position (Function 3). The agent will be able to obtain the data from sensors or markers in the future. It can now obtain it by pushing buttons instead as shown in Figure 4.

Fig. 5: ARW Agents for devices (a camera and a sensor)

4.2 ARW Objects

We developed software as a part of the prototype system for handling ARW Objects. 3DCG contents in ARW Objects are especially called AR3D. Figure 6 shows the software called AR3D Creator. The current version can load VRML and standard 3D object files (the hide file extension is .obj) for Advanced Visualizer developed by Wavefront as 3D coordinate data. We can easily change the appearance of AR3D (Diffuse, Ambient, Specular, Emissive and Texture), and save and read the setting data through object configuration files (the hide file extension is .ocf).

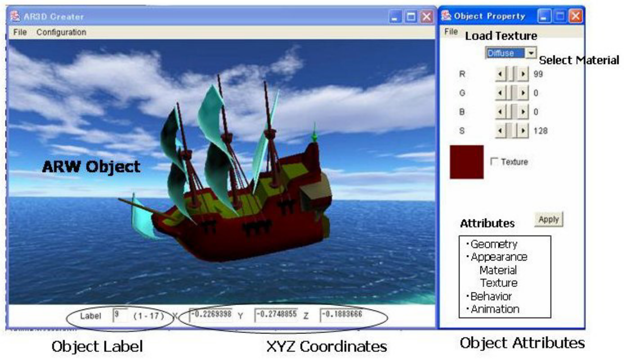


Fig. 6: AR3D Creator for handling ARW Objects

The prototype system can load the 3DCG contents created by AR3D Creator, position them at specified 3D coordinates, and animate the object by spline interpolation after the position that users want to move to is specified through ARML as shown in Figure 7 (Function 4-1). The system can also place 2D objects like images at any specified positions. For example, an image of a web site is placed in the AR world as shown in Figure 8 (Function 4-2).

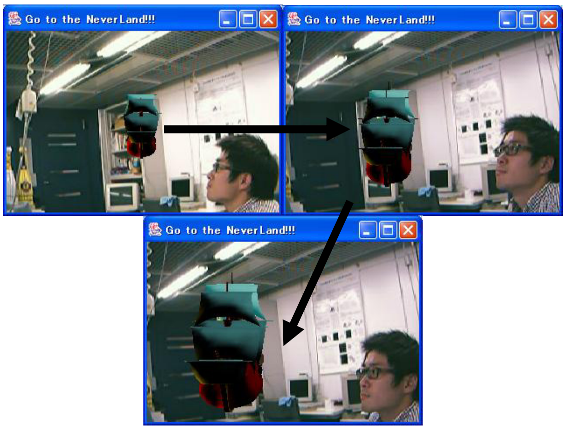


Fig. 7: Object animation

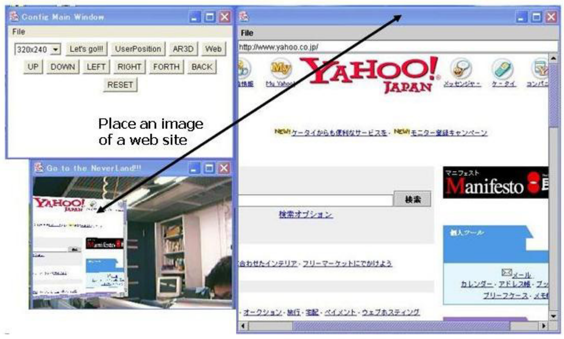


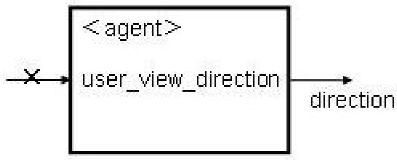
Fig. 8: Positioning of a web site image

4.3 ARW Core

ARW Core parses ARML. ARML is described by XML. ARW Core has XML parser called Xerces Java Parser. We defined some XML descriptions as follows to construct the system. Figure 9 shows the schematic of the system.

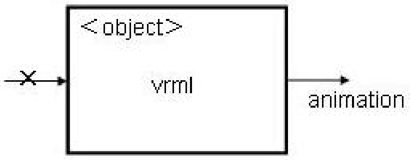
<agent>
<name>camera</name>
<output>image</output>
<comment>Function 1: get images from a camera for user view</comment>
</agent>

<agent>
<name>user_view_direction</name>
<output name=direction>raw, pitch, yaw</output>
<comment>Function 2: get user view direction from a sensor</comment>
</agent>

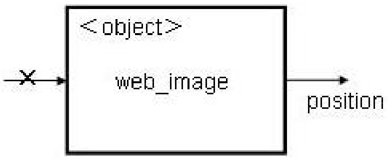


<agent>
<name>user_position</name>
<output name=position>x, y, z</output>
<comment>Function 3: get user position from buttons</comment>
</agent>

<object>
<name>vrml</name>
<animation>x0, x0, x0: … : xn, xn, xn</animation>
<comment>Function 4-1: a sample of object animation</comment>
</object>



<object>
<name>web image</name>
<position>x, y, z</position>
<comment>Function 4-2: a sample of a 2D object</comment>
</object>



<connect>
<from>agent.user_veiw_direction.direction</from>
<to>core.view.direction</to>
</connect>

<connect>
<from>agent.user_position.position</from>
<to>core.view.position</to>
</connect>

<connect>
<from>agent.camera.image</from>
<to>core.view.image</to>
</connect>

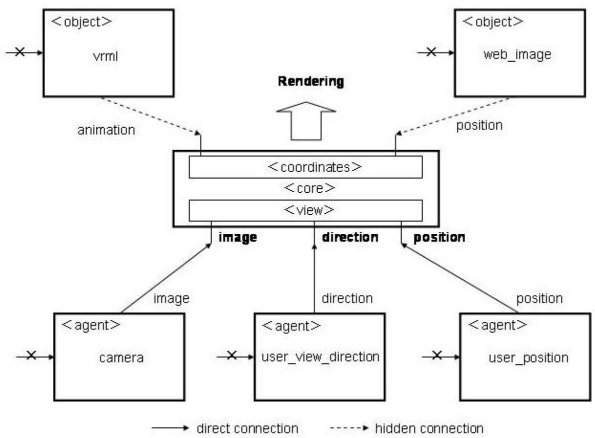


Fig. 9: Schematic of the system

<agent> means an ARW Agent, and <object> means an ARW Object. We have to necessarily put </xxx> like </agent> or </object> at the end of the description as an XML rule and can put data between the tags. We basically put <name> and argument tags, for example, <input> or <output>, <animation>, <position>, or <comment> in this case. <name> is unique like ID. It is an essential tag to identify and access from other tags. <input> and <output> are data interfaces to connect with each other. <position> and <animation> in <object> are peculiar to connect with 3D coordinates in AR world through <coordinates> in <core>. We can write down some comments in <comment> tags. <connect> indicates data flow, and <from> values is transferred to <to> values. We describe the values that we want to access in <from> or <in> tags by linking tags with dots (.) like method access of object-oriented languages. <core> means an ARW Core. <core> is not on the list. It exists from the beginning, and is hidden. We can access it through "core." like core.view.direction. <core> has <view> and <coodinates> as shown in Figure 9. <view> is similar to view of Java3D, and has interfaces like image, direction, and position to connect with.

5. Future Works

This paper described ARW Framework, which can combine the virtual and real worlds, and proposed ARML to describe linkages between these worlds. However, these specifications are only the first version. The next version, which will be improved as a result of discussion and feedback from ARW Framework implementation, will be released soon.

References

1. Kei Nishimura, "ARW Framework and Intelligent Character Agent", IPSJ SIG Technical Reports, Vol.2004, No.28, pp.95-100 2004 in Japanese.
2. Toyoaki Nishida, Creative Network by Creating with Agents (Book style). Baihukan, 2003 in Japanese.
3. Joseph P. Bigus and Masayuki Ida, Constructing Intelligent Agents using Java (Book style). Softbank Publishing, 2002 in Japanese.
4. Takashi Nagao, Advanced Agent Technologies (Book style). Kyouritsu, 2000 in Japanese.
5. Henry Sowizral and Yukio Ando, Java/Graphics Programming The Java3D API Specification (Book style). ASCII, 1999 in Japanese.
6. Yasuyuki Endo, Java3D Programming Bible (Book style). Natsume, 2003 in Japanese
7. Lodger Lea and Kouichi Matsuda, JAVA for 3D and VRML Worlds (Book style). Pearson Education Japan, 1997 in Japanese.
8. Tetsuo Hirouchi, Web3D Graphics – Virtual World by creating VRML (Book style). Pearson Education Japan, 2001 in Japanese.
9. it3d: http://open.nime.ac.jp/software/it3d/ in Japanese
10. MR Platform System: http://web.canon.jp/mr-system/product/mr_platform_system/top.html in Japanese

11. Web3D Consortium: http://www.web3d.org/
12. J. Leigh, A.E. Johnson, C.A. Valsilakis and T.A. Defanti, “Multi-Perspective Collaborative Design in Persistent Networked Virtual Environments”, Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS 96), Santa Clara, CA, pp.253-260 and 271-272, March 1996.