Proceeding of ICITS 2020 e-ISSN: 2716-6732
6th International Conference on Information Technology & Society
November 11th, 2020, Selangor, MALAYSIA

Components Role Differences Between Augmented Reality Framework For Distributed Collaboration (ARTiFICe), Distributed Wearable Augmented Reality Framework (DWARF) And VARU Framework: Enabling Rapid Prototyping Of VR, AR And Ubiquitous Applications.

Marliza Abdul Malik, Suhaimi Mohd Noor, Suziyanti Marjudi and Salyani Osman

Faculty of Communication, Visual Art and Computing, Universiti Selangor

marliza@unisel.edu.my

Abstract. This paper summarizes the differences of component roles between three Augmented Reality frameworks. The framework are ARTiFICe : Augmented Reality Framework for Distributed Collaboration, DWARF : Distributed Wearable Augmented Reality Framework and VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications Components of Frameworks. Possibility the combination of suitable components from these three frameworks that can help development of Augmented Reality Application.

Keywords: Augmented Reality Framework, ARTiFICe, DWARF, VARU.

INTRODUCTION

Augmented reality has been put to use in a number of fields. The goal of augmented reality is to add information and meaning to a real object or place. According to Azuma (1997), Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him.

According to Steve Chi-Yin, Gallayanee Yaoyuneyong and Erik Johnson (2011), AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it.

There are many Augmented Reality Frameworks such as The Studierstube Augmented Reality Project, ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning and many others frameworks.

Augmented Reality Framework for Distribution Collaboration (ARTiFICe) by Annette Mossel, Christian Schönauer, Georg Gerstweiler and Hannes Kaufmann (2012) main objective is to help students getting over the initial hurdles of creating quick prototypes of an embodied AR experience. There are three components (1) a graphical user interface and scene management for rapid prototyping of a VR/AR application (2) an adaptable interaction and distribution framework for collaborative applications for mobile as well as workstation-based VR/ARsetups and (3) it supports versatile VR/AR setups (mobile,semi-immersive, immersive) on different operating systems and platforms, integrates various input devices such as 2D markers, 3D mice, video game controllers, depth cameras, 6 DOF targets,and supports a range of output devices e.g. smartphone.

Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riß, Christian Sandor, Martin Wagner (2001) proposed DWARF : Distributed Wearable Augmented Reality Framework This framework has three basic aspects: services, middleware and architecture. The primary goal of this framework is to develop a research platform for AR on wearable computers and in intelligent environments. Besides this research focus, the project aims to bring well-established parts of AR technology into practical use within a short time frame. This idea was proposed last year by Dave Mizell.

VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications by Sylvia Irawati, Sangchul Ahn , Jinwook Kim and Heedong Ko (2019). The VARU framework is designed for the rapid prototyping of a tangible space application. It is designed to provide extensibility, flexibility and scalability. Extensibility refers to the ability to add a new functionality to the current framework without modifying the overall system. The primary goal of framework is to make the development of VR, AR and UC applications easier and more efficient by having a unified representation of interaction spaces. Depending on the available resources, the application developer could easily switch from one interaction space to another without significant effort. Since VARU supports multiple interaction spaces, it enables collaboration across the various spaces.

ARTiFICe

This framework has two types of input. There are mobile devices and workstation devices. Mobile devices such as smart phone or tab and workstation are Kinect, 3D mouse, 6DOF interaction and Razer Hydra. This paper only focusses in mobile devices as an input layer.

DWARF

First, DWARF consists of software services such as trackers, running on hardware modules. Each service provides certain abilities to the user or to other services.

Second, DWARF contains the middleware necessary to match these services dynamically and set up the communication between them, so that the system configuration can change at runtime.

Third, the conceptual architecture of DWARF describes the basic structure of AR systems that can be built with it. This ensures that service developers agree on the roles of their own services within the system and on interfaces between them. The architecture is also easy enough for end users to understand, so that they can reconfigure their wearable system by simply plugging together the appropriate hardware modules.

VARU Framework

The VARU framework is designed for the rapid prototyping of a tangible space application. It is designed to provide extensibility, flexibility and scalability. Extensibility refers to the ability to add a new functionality to the current framework without modifying the overall system. Depending on the application scenario, new components could be added and connected to the existing component. Flexibility refers to the ability to adapt to the different application scenario. The VARU components can be classified into two main parts: the VARU Server and the VARU Client.

The VARU components can be classified into two main parts:

1. The VARU Server (Application Layer)
   Consists of three main components - the Simulation Server, the Object Server, and the Object Database.
   The Simulation Server is responsible for generating a simulation of the physics law in the virtual environment.
   The Object Server is responsible for synchronizing the objects in different interaction spaces. It receives the inputs from VARU Clients and updates
   The Object Database using those inputs and/or using the results from the simulation.
2. The VARU Client.

Kernal : bridges the communication between the VARU Server and the VARU Client and also the communication among the VARU Client components. It uses a router mechanism to dispatch the values from component to component. (Middle ware)

Space Manager : each interaction space has its own space manager: VR Manager, AR Manager and UC Manager, which is responsible for managing its interaction space. It is also possible for the VARU Client to have a combined space manager, (Middle ware)

Components of Framework

a) First Component: Can be group as input/output device

   a. Dwarft: Services- trackers, running on hardware modules. Each service provides certain abilities to the user or to other services

b. Artifice: Input - Tracking Devices & Integration. creation of mobile AR applications by using the display of the mobile device as a "magic lens" into an augmented world

c. VARU : VARU Client (Device Manager) is responsible for managing the input/output devices used for interaction. It provides a device abstraction, such as an analog, button, tracker, force, sound, and speech devices. This device abstraction provides an interface for communication with the peripheral devices, e.g., tracker, joystick, and speech. The Device Manager gets the device input values and maps those values into the high level abstraction data, then sends the data to the other components through the Kernel for further processing.

b) Second Component: Middleware because create communication between user and applications.

a. Dwarft : Middleware - tracking and modeling the user's environment, representing structured information, access to legacy services, and platform-independent representation of multi-modal user interfaces. tracking subsystem consists of software services that can be combined dynamically. Its hierarchical architecture is based on low-level trackers which provide information of an object's position and orientation and a tracking manager which combines their output.

b. Artifice-: Middleware - A tracker object is used to access raw tracking data. This data is then processed by the specific IT and handed to the interaction framework. The abstraction layer ObjectSelectionBase provides a straight forward and clean interface of data handling for workstation as well as mobile setups and offers a transparent layer to integrate new techniques into the framework. The only information which must be handed over to ObjectSelectionBase is a list of the selected object(s) and the absolute pose of the interaction object, calculated by the IT. This data is then processed by the InteractionBase class and delivered to all selected virtual scene objects. Virtual scene objects which should be selectable must have the ObjectController class attached. Depending on the given pose the ObjectController manipulates the position and orientation of the selected scene object.

c. VARU : - the VARU Client.

Kernal : bridges the communication between the VARU Server and the VARU Client and also the communication among the VARU Client components. It uses a router mechanism to dispatch the values from component to component.

Space Manager : each interaction space has its own space manager: VR Manager, AR Manager and UC Manager, which is responsible for managing its interaction space. It is also possible for the VARU Client to have a combined space manager,

c) Third Component: Application layer

a. Dwarf : - architecture - The application is shielded from the low-level services, such as for user interface or tracking hardware, and accesses these at a higher level of abstraction using the various DWARF services. It includes a special service which provides bootstrapping functionality and "glue logic". This provides the other services with models of the world and of tasks the user wishes to perform.

Tracking and modeling the user's environment, representing structured information, access to legacy services, and platform-independent representation of multi-modal user interfaces.

Tracking Subsystem > World Model > (special service) > Task Flow : 1) Context-aware service access 2) User Interface engine

b. Artifice: Application Layer - Using interaction framework.

Raw tracking data which is fed into the transformation node of a virtual scene Unity3D.GameObject should be controllable to provide interaction techniques (IT) for 3D objects selection and manipulation. The abstraction layer ObjectSelectionBase provides a straight forward and clean interface of data handling for workstation as well as mobile setups and offers a transparent layer to integrate new techniques into the framework. The only information which must be handed over to ObjectSelectionBase is a list of the selected object(s) and the absolute pose of the interaction object, calculated by the IT. This data is then processed by the InteractionBase class and delivered to all selected virtual scene objects. Virtual scene objects which should be selectable must have the ObjectController class attached. Depending on the given pose the ObjectController manipulates the position and orientation of the selected scene object.

c. VARU Server consists of three main components - the Simulation Server, the Object Server, and the Object Database. The Simulation Server is responsible for generating a simulation of the physics law in the virtual environment. The Object Server is responsible for synchronizing the objects in different interaction spaces. It receives the inputs from VARU Clients and updates the Object Database using those inputs and/or using the results from the simulation.

TABLE 1. Differences of Component Roles.

COMPONENT	FRAMEWORK	DIFFERENCES
Input/Output	ARTIFiCE	No differences because each component refer to same device or hardware. Each of the device or hardware did same task.
	DWARFT	
	VARU	
Middleware	ARTIFiCE	data is then processed by the specific IT and handed to the interaction framework
	DWARFT	tracking subsystem consists of software services that can be combined dynamically. Its hierarchical architecture is based on low-level trackers which provide information of an object's position and orientation and a tracking manager which combines their output.
	VARU	Besides the Kernel, the VARU Client must also have at least one Space Manager. As previously mentioned, each interaction space has its own space manager: VR Manager, AR Manager and UC Manager, which is responsible for managing its interaction space. It is also possible for the VARU Client to have a combined space manager, such as a combined AR and UC Manager.
Application	ARTIFiCE	The abstraction layer ObjectSelectionBase provides a straightforward and clean interface of data handling for workstation as well as mobile setups and offers a transparent layer to integrate new techniques into the framework.
	DWARFT	It includes a special service which provides bootstrapping functionality and "glue logic". This provides the other services with models of the world and of tasks the user wishes to perform.
	VARU	synchronize the object across various spaces. three levels of descriptions for the objects in a tangible space: Class, Individual, and Extension.

FUTURE WORK AND CONLUCSION

Currently, we are working to create Augmented Reality Framework that suit with marker less Augmented Reality application. Possibility the combination of suitable components from these three frameworks that can help development of Augmented Reality Application. Therefore, we are currently evaluating the components with the related existing augmented Reality applications. Moreover, we plan to provide our framework as reference to develop marker less applications.

ACKNOWLEDGMENTS

The authors would like to thank supervisor and co supervisor because guide in fulfill the need of study.

REFERENCES

1. D. Mizell. (2000.) Augmented Reality Applications in Aerospace. In Proceedings of ISAR 2000, Munich,
2. VARU Framework: Enabling Rapid Prototyping of VR, AR and Ubiquitous Applications Sylvia Irawati 1,2 Sangchul Ahn 1,2 Jinwook Kim 1 Heedong Ko 1
3. ARTiFICe ± Augmented Reality Framework for Distributed Collaboration Annette Mossel, Christian Schönauer, Georg Gerstweiler and Hannes Kaufmann
4. Design of a Component–Based Augmented Reality Framework Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riß, Christian Sandor, Martin Wagner
5. Steve Chi-Yin, Gallayanee Yaoyuneyong, Erik Johnson (2011), *Augmented Reality: An Overview and Five Direction for AR in Education* , Journal of Educational Technology Development and Exchange, 4(1), 119-140.
6. Wolfgang Broll¤, Irma Lindt¤, Jan Ohlenburg¤, Michael Wittk¨amper¤, Chunrong Yuan¤, Thomas Novotny¤, Ava Fatah gen. Schiecky, Chiron Mottramy, Andreas Strothmannz (2004) , ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning.
7. Dieter Schmalstieg, Anton Fuhrmann Gerd Hesina,Zsolt Szalavári,L. Miguel Encarnação, Michael Gervautz And Werner Purgathofer (2020) The Studierstube Augmented Reality Project.