Synchronisation between Real and Virtual-World Devices in a VR-IoT Environment

Anderson Augusto Simiscuka
School of Electronic Engineering
Dublin City University, Ireland
anderson.simiscuka2@mail.dcu.ie

Gabriel-Miro Muntean
School of Electronic Engineering
Dublin City University, Ireland
gabriel.muntean@dcu.ie

Abstract— **Virtual Reality (VR) can be used for many applications in diverse fields such as engineering, gaming, healthcare, education, etc. Internet of Things (IoT) devices, including CCTV monitoring cameras and smart watches, support diverse services including providing sensor data, intelligence in appliances and multimedia. This paper proposes and describes VRITESS, a novel VR-IoT environment synchronisation scheme which facilitates a seamless user experience for IoT devices through use of VR, and synchronised representation of IoT devices in the virtual world. Certain IoT devices, which are located in extreme environments or are not straightforward in terms of operation for many users, can be manipulated in a simpler way in a virtual environment. A synchronisation mechanism keeps the real devices up-to-date, according to actions and events that happened in the virtual world and vice-versa: actions applied on the real devices are transferred to the virtual world. The proposed solution is tested in a real-life testbed developed in the Performance Engineering Laboratory at Dublin City University, Ireland, with the Oculus Rift and several IoT devices.**

Keywords—multimedia internet of things, virtual reality (VR), three-dimensional visualisation, user quality of experience.

I. Introduction

The Internet of Things (IoT) is expected to connect 500 billion devices by 2030, aggregating and analysing data and delivering insight to users [1]. Devices such as thermostats, wearables (such as smart watches and wrist bands), medical devices (e.g. blood pressure monitors) and even cars, will offer new ways of interacting with users. Unfortunately many of these are still complex to use and do not offer a mature and simple experience to users [2]. Beyond the employment of multimedia solutions [3, 4, 5], the use of Virtual Reality (VR) technology is a powerful avenue to bring users a rich media experience and an increase sense of reality, by offering a realistic, visual, auditory, and tactile sense of the world. The possibility of operating virtual objects using human senses, language, and gestures, can make easier for users to understand and interact with the IoT devices [6]. Devices located in extreme environments (e.g. landslide sensors, water level monitoring devices) or which are not very user-friendly, can be manipulated in a simpler way in a virtual environment.

This paper proposes an innovative VR-IoT Environment Synchronisation Scheme (VRITESS) for devices in both real and virtual worlds. VRITESS allows users to seamlessly operate IoT devices, such as beacons and sensor functions in

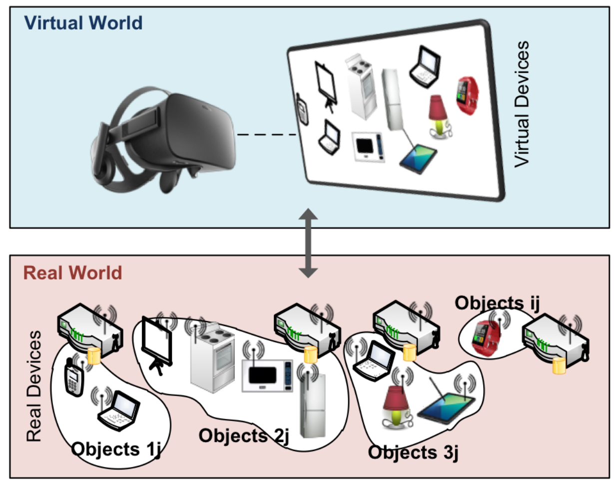

Fig. 1. Real-world and virtual-world devices

single-board computers such as the Raspberry Pi, in a VR environment, using a simpler interface which supports interaction with the devices. When operating a device in the virtual environment, users' decisions will be shown in the virtual environment and will also affect real devices, as illustrated in fig. 1. Changes in the real devices will also be shown in the virtual devices, so there is a complete synchronisation between real life and virtual reality.

VRITESS will be deployed and tested in a real test-bed including an Oculus Rift, Raspberry Pis and beacons. We expect users to find the operation of devices easier and more user-friendly using the VR equipment. The 3D environment will recreate our office with the virtual devices located within this office. The VRITESS synchronisation keeps the devices up-to-date, according to actions and events that happened in both virtual and real life, reflecting changes on each other.

The rest of this paper is organized as follows. Section II discuses related works and section III introduces the architecture for the proposed solution. Section IV presents our proposed synchronisation mechanism and section V describes the test-bed, test scenarios and discusses the results. Finally, the conclusions and future work directions end this paper in a dedicated section.

II. Related Works

The VR and IoT-related research works discussed in this paper are classified in categories: *Virtual Reality background*, *Multimedia IoT*, *IoT and VR*. These topics are closely related to the solution proposed in this paper and therefore it is important to review the state-of-the-art in each topic.

A. *Virtual Reality Background*

In [7], authors defined Virtual Reality (VR) as a technique for simulation of the real world by applying the theory of immersion into a virtual 3-D space in which real human senses are similar to that of real world within a certain area. Authors identified areas where VR can be applied: *military* (e.g. training), *education* (e.g. a collaborative classroom), *healthcare* (e.g. robotic surgery, phobia treatment), *entertainment* (e.g. games, virtual museums), *business* (e.g. virtual tours, product prototyping), and *engineering* (e.g. modelling tools, project viewing).

Authors highlighted the importance of user interface (UI), audio and graphics for VR in [8]. UI design needs to consider end-user control, notion of space and integration with the experience. Audio needs to receive important attention when developing VR environments, as sounds guide users. The graphics, which can be polygonal or 360° recordings of the real world, must not be distorted, and need to efficiently map the available space the user has to move.

A VR approach to explore software-based cities by using a head-mounted display and gesture-based interactions was introduced in [9]. A reusable gesture design for 3D model manipulation was also created. Interviews were conducted, where participants had to solve comprehension tasks and rate the usability of the gestures and VR experience. Participants praised gestures for translation, rotation, and selection, however, the zooming gesture was less favoured.

The analysis of these works were fundamental for the development of an enjoyable user-friendly 3D environment.

B. *Multimedia IoT*

Several solutions for supporting multimedia in relation to IoT have been proposed. A novel web application framework, based on Google Web Toolkit, aimed at enhancing the interaction among things and between humans and things was presented in [10]. An application framework, based on the Google Web Toolkit, provided users with simple methods for integrating smart things into a flexible visualization tool (visualizer), for manipulating them both graphically and functionally, and for managing them and their interactions through a thing-centric design (provider), modularity (manager) and web service communications (RESTful). HealthCare applications were tested using the framework, however, a larger spectrum of IoT applications have yet to be developed on the WebIoT framework in order to evaluate the technical feasibility of shipping plugins as separate pieces of the framework, so as to enable loading them at run time on the application server through the WebIoT interface itself.

A novel concept, Multimedia IoT (MIoT), was introduced in [11]. Besides introducing MIoT, a layered QoE model was also developed for it. MIoT consists of layers: physical devices, network, combination, application and context. Tests used a vehicle application for remote tutoring. Simulations considered 12 different scenarios (using different configurations of bitrate, map sync, speed/rpm sync, speed/rpm visualisation), and presented the best QoE scenario where accurate data is the most important factor for QoE and synchronism second most, and formulas for composite virtual objects. The paper, however, considered only one kind of application (assisted driving), and the QoE model can only be applied to IoT with multimedia content and user-oriented.

In [12] the authors proposed an IoT approach that can improve education and streamline organisations. The aim was to use IoT technologies to improve multi-party audio-visual collaboration during online classes. Factors such as noise caused by open windows, poor lighting or high volume of speakers can cause disturbance to the overall collaboration in the online sessions. In order to improve these conditions, sensors could be installed in the classrooms analysing the ambient conditions and report if the room is ready for remote collaboration. The continuous monitoring can send notifications or take some actions automatically (e.g. decrease the volume of incoming audio or adjusting the lights). Automatic detection of voice activity could unmute microphones improving the quality of communications. during an online class. During online sessions, an accurate 3D location system (e.g. Ultra-WideBand), based on tags location, can improve zooming and position of cameras. Even though the ideas are very interesting, the authors did not implement and test them yet.

A multicast routing solution for supporting multimedia communications in IoT was introduced in [13] by proposing fast multi-constrained multicast routing algorithms. Point-to-multipoint trees are created for meeting multiple QoS constraints required by real-time multimedia communications. In their experiments, the authors have used a network size from 100 to 500 nodes, and the number of destinations is set to 5 and 10. Analytical and experimental results demonstrated that the proposed algorithm is superior to a representative multi-constrained multicast routing algorithm in terms of both speed and accuracy. The algorithms proposed, however, assume the availability of current network topology and the QoS constraint information was associated with the network topology.

A novel data model and storage infrastructure for the IoT was introduced in [14], using a document-oriented approach to support both heterogeneous and multimedia data. The solution was built on top of the CouchDB NoSQL database server and offers a RESTful API that provides a set of features to IoT applications, such as replication for load balancing, distributed query processing, and notifications. Proposed optimisations reduced the transfer time of documents containing multimedia data, however the solution is not a complete framework for web-of-things yet, and authors want to introduce support to online social networks.

All these solutions offer insights to the development of our solution: the integration of devices to a virtualised version of them, the user experience analysis that can be done, different applications for our approach, a multicast approach for devices discovery and a data model that we use for the communication between virtual and real devices.

C. IoT and VR

The integration of IoT and VR is still very recent, therefore, it has received little investigation in the literature. Some of the works that were done in the area are presented in this section.

A model representing a connection and interaction layer between smart objects and people was presented in [15]. Authors proposed a virtual platform able to engage users and show the locations of smart objects deployed in the city, as well as what functionalities are offered. A prototypical Virtual Environment of Things (VEoT) was developed within a 3D environment in which users can explore the virtualised urban area and interact with the available smart objects through gestures and VR devices. The VEoT receives real-time data produced by a multi-protocol sensing middleware for the interaction with physical devices through high-level RESTful APIs.

In [16] a network virtual reality (VR) engine for interactive analysis and smart city immersive visualisation was proposed. It can be used in big spatial data (e.g., remote sensing data) organisation and presentation while achieving the online sharing by hash-based P2P networks. In the network, a map containing the real geographic space users and the virtual scene with user avatars and virtual network nodes was developed. To evaluate the system, a user study was conducted by comparing it with a virtual community system. The performance evaluation results showed higher usability and user satisfaction on the authors' system.

The Hyper-connected IoT-VR Platform was proposed in [17]. It was designed for interconnection of humans, devices in a virtual space. Authors utilised the high interoperability of IoT platform-based services and the intuitive services from VR. The platform could provide customisable and intuitive remote services and was created based on the idea that IoT and VR technology can be complementary when linked together. A real test bed was also evaluated.

III. VRITESS Solution Architecture

This section presents the architectural components and communication process employed by our proposed solution.

A. Architectural Components

The VRITESS solution is implemented on top of an IoT architecture shown in fig. 2. This architecture has been proposed and used in [18, 19, 20, 21], and its components are: *IoT objects* (e.g. wireless devices, sensor based devices, appliances), which provide one or more services to other objects – these devices will be present in the scenario in two

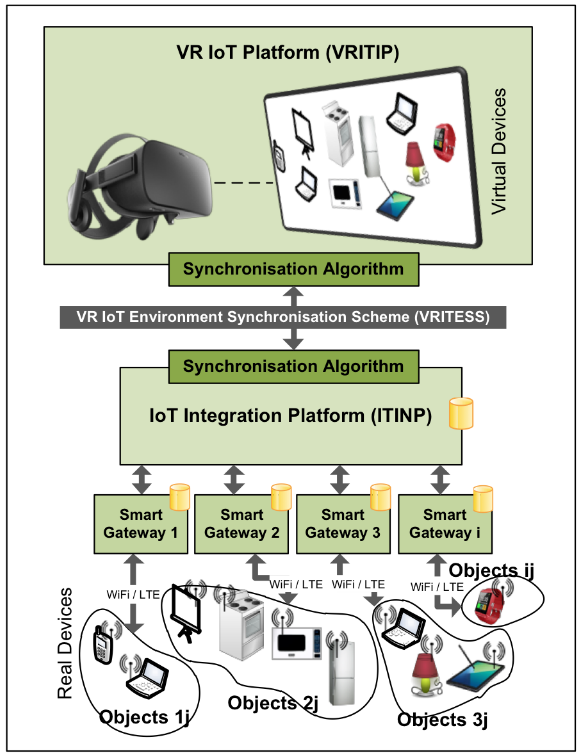

Fig. 2. VRITESS architecture

forms: real (the real IoT devices) and virtual (the virtualised version of each device); *smart gateways,* which provide connectivity to IoT objects, and allow objects communicate to each other via the smart gateway they are currently attached to; the *IoT Integration Platform (ITINP)*, a cloud-based server platform that integrates services and functions and performs the networking among smart gateways, and the *VR IoT Platform (VRITIP)* which is in control of the *VR IoT Environment.* VRITIP is responsible for controlling and presenting to users a user-friendly virtual version of IoT devices, and also responsible for keeping these virtual devices updated with the latest alterations made in the real world. VRITIP sends alterations made in the virtual world to ITINP, which controls the real world devices.

The proposed *VR IoT Environment Synchronisation Scheme (VRITESS)* is deployed between ITINP and VRITIP.

B. Communication Process

We follow a status structure that the gateways use to register the actions from the devices, with ITINP receiving actions from real devices and VRITIP receiving actions from virtual devices, allowing a two-way communication. Both virtual and real devices write instructions to this status structure, and they need to keep updating their functionalities based on the latest instruction. Examples include turning on/off and measuring temperature on the Raspberry Pi, and monitoring sensor activity on the beacons (e.g. light detection, motion, etc.). Virtual and real-world status are generated by VRITIP and ITINP, respectively. The two platforms are synchronised between them to keep the virtual and real devices' statuses, functionalities and services also synchronised. Users are considered to exist on both virtual and real world as well, so it is possible to track the way users interacted to devices.

An example of a virtual device status structure can be found in the following (a virtual user turning a virtual device off):


```
{
"timestamp":      "2018-01-04T13:31:37.459Z",
"last change":    "2018-01-04T13:31:35.133Z",
"user id":        "asimiscuka",
"type of user":   "virtual",
"type of device":"virtual",
"tags":           ["POWER_INSTRUCTION"],
"device_id":      "0024:4F31:0000:01CB",
"data":           ["OFF"]
}
```


This interaction results in the real object to be turned off as well:


```
{
"timestamp":      "2018-01-04T13:31:37.956Z",
"last change":    "2018-01-04T13:31:35.437Z",
"user id":        "asimiscuka",
"type of user":   "virtual",
"type of device":"real",
"tags":           ["POWER_INSTRUCTION"],
"device_id":      "0024:4F31:0000:01CC",
"data":           ["OFF"]
}
```


In this example, the virtual user is turning off a virtual representation of a device and the same action is being transmitted to the real device. Functions that a user can perform in the virtual environment (e.g. turning on or off a device, changing working hours of a device, increasing or decreasing temperature of a thermostat, etc.) will be synchronised in ITINP and will be then updated on the database of the smart gateway that contains that object, passing the instruction to the object.

VRITESS synchronisation algorithm will be responsible for keeping the devices up-to-date, according to actions and events that happened in the virtual and real versions of the devices, respectively.

IV. VRITESS Synchronisation

The VRITESS synchronisation algorithm acts on both real world, at the level of ITINP (acting on the real devices), and virtual world, at the level of VRITIP (acting on the virtual devices), maintaining consistency.

As seen in algorithm 1, the timestamps and data structure make the real and virtual devices maintains the status up-to-date and synchronised with each other. The communication follows the structure presented in section III.B of this paper. Some of the instructions may be informative (e.g. show temperature of real device in the virtual device in a gauge) or action (e.g. turn on/off both virtual and real devices).

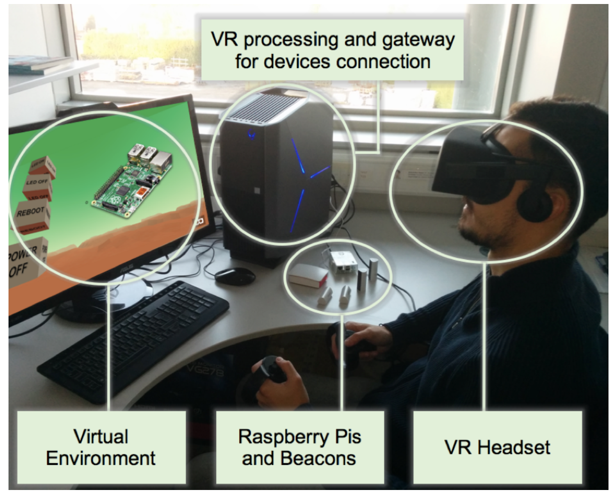

Fig. 3. The implemented test-bed

Algorithm 1 – Synchronisation of real-world and virtual-world IoT Devices

```
Require: old_t: The timestamp sent by ITINP and VRITESS
         datas: The set of data, following the structure in section III.B
Output: old_t: New timestamp sent back to ITINP and VRITESS
        updated.datas: Updated data sent back to ITINP and VRITESS
new_t = old_t
for data in datas
    if data.timestamp > old_t
        updated_datas.add(data.value)
        new_t = max(data.timestamp, new_t)
return new_t, updated_datas
```

V. Test-Bed Implementation

Our test-bed is implemented in the Performance Engineering Laboratory at Dublin City University, Ireland, as seen in fig. 3. We are currently using the Oculus Rift [22], Raspberry Pis [23] and Beeks Beacons [24]. A powerful Alienware computer is being used as a gateway to interconnect devices, the Oculus Rift, and to render the 3-D virtual world. The specifications of the Alienware computer, Beeks Beacons and Oculus Rift used, are available in Tables I, II and III, respectively.

The applications developed can be divided in parts: The Raspberry Pi Java application, the MacBook Air running a Glassfish 4.0 server [25] with a Java application, a JavaServer Faces (JSF) [26] web application, a MySQL database for statuses recording and access control, and a web application for user interaction in VR devices.

The development of the test-bed started with the creation of the two Java applications in the Glassfish server and the Raspberry Pi. Threaded Java ServerSockets are used on both sides so several devices can connect to our Glassfish instance. Objects are exchanged using the sockets and Input/OutputStreams every 0.5 seconds. These objects carry all statuses for the five types of interactions our web application enables, from virtual device (in the web application) to real device and vice-versa. The five types of

TABLE I
ALIENWARE

Parameter	Value
Model	Alienware Aurora R6
Processor	Intel Core i7-7700
RAM Memory	16GB
Hard Drive	1TB
SSD	256GB
Graphics Card	NVIDIA GeForce GTX 1080 8GB
Operating System	Windows 10
3D Development	Unity Personal 2017.3

interactions are: Temperature measurement on the Raspberry Pi and beacons; and turn led on and off, reboot and restart, these last four only on the Raspberry Pi.

Our JSF web application also runs on Glassfish server and was designed to work on mobile device screens and VR devices. JSF creates the binding between the user interface and the core of the application that sends instructions to the IoT device. The HTML pages of our application include Mozilla's A-Frame [27], an HTML-based framework for VR development. A-Frame contains tags that enable head movements in smartphones, natively reading smartphones' accelerometers and gyroscopes, and everything in a 3D environment. It also mirrors content for the use of the device inside of VR headsets such as Google Cardboard (each half of the smartphone screen is seen individually on the headset, adapted for left and right eyes, creating a 3D effect). In fig. 4, it is possible to see how the Web-VR application displays the virtual device turned off. In figs. 5 and 6, the virtual device has its LED off and on respectively. The application also allows users to control objects they have permission to, and share objects with other users, creating a social network of inter-connected IoT devices. In order to enable this functionality, users are assigned user IDs and they can share their devices with other user IDs. A-Frame is fully compatible with browsers such as Google Chrome and Mozilla Firefox, and with the Oculus Rift.

The MySQL database receives all information necessary for the monitoring of the solution, with timestamps, ids for user, device and tasks, and the lists of which users can see which devices, something achieved by the web application with login and share functions.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

The VR-IoT Environment Synchronisation Scheme (VRITESS) was proposed and described. VRITESS allows users to seamlessly operate IoT devices and includes a synchronisation algorithm that keeps the devices up-to-date, according to actions and events that happened in both virtual and real life, reflecting changes on each other.

The test-bed consists of a web server which runs the core and web applications, with mirrored display for the VR headset. The core application which also runs on the Raspberry Pi, is threaded and allows multiple VR devices to access it.

TABLE II
BEEKS BEACONS

Parameter	Value
Battery	3.6V / 2600mAh – Primary Lithium
Size	2.36" x 0.85" (60mm x 21mm)
Weight	1.0 oz (28 gr)
Temperature Range	-30°C to +77°C
Bluetooth Type	Bluetooth Low Energy 4.1
Bluetooth Sensitivity	-97dBm
Bt. Max Power Output	+5dBm
Bluetooth Antenna	0dBm Single Antenna, Omni Directional
Bluetooth Data Rate	1Mbit/s / 2Mbit/s*
Bluetooth Security	128 bit AES
Power Consumption RX	7.5mA RX Active Mode
Power Consumption TX	6.5mA TX Active Mode
Power Consumption Sleep	1.6 μA (SRAM retention and RTC running)
Power Output	• -40dBm to +5dBm.
CPU	Dual Code: ARM Cortex M3 and M0
Sensors	*High Accuracy Temperature sensor* *3 Axis Accelerometer* • Detect.: Freefall. Motion, Pulse, Transient • Custom detection: Door opening/closing with counter; human walking detection; driving detection, motor vibration learning *Magnetometer* • Custom detectable modes: Door opening and closing, Metal nearby trigger, car detection, electric motor, efficiency/torque *Light Sensor*
Internal Flash Memory	55KB Flash standard
LED	Red LED

TABLE III
OCULUS RIFT

Parameter	Value
Display	PenTile OLED
Graphics	2160x1200 (1080x1200 per eye) @ 90 Hz
Sound	Integrated 3D audio headphones (user removable/exchangeable)
Input	6DOF (3-axis rotational tracking + 3-axis positional tracking) through USB-connected IR LED sensor, which tracks via the "constellation" method.
Controller input	Oculus Touch motion tracked controllers.
Connectivity	HDMI 1.3, USB 3.0, USB 2.0
Weight	470 g (1.04 lb)

Our future work includes testing of multiple gateways with users accessing devices in different gateways with their VR headset, and also introduction of other smart devices and beacons into the virtual world. Measuring network performance when several IoT and VR devices are interconnected in a social virtual reality network is also envisaged. Testing targets delay, loss, and other QoS and Quality of Experience (QoE) metrics.

ACKNOWLEDGMENT

This work was supported by the Irish Research Council and Dublin City University, grant number EPSPG/2015/178, and in part by the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 688503 for NEWTON project (http://newtonproject.eu).

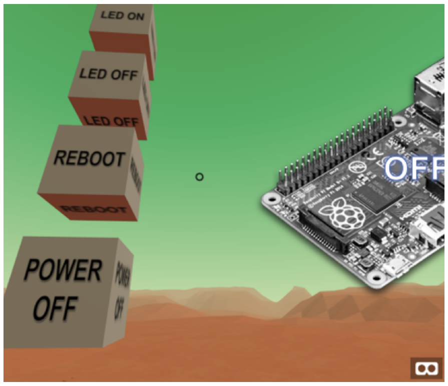

Fig. 4. Virtual Raspberry Pi – Device Off

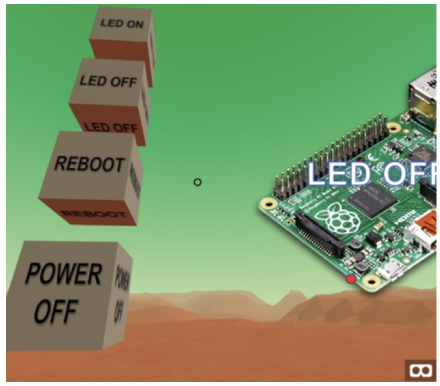

Fig. 5. Virtual Raspberry Pi – Led Off

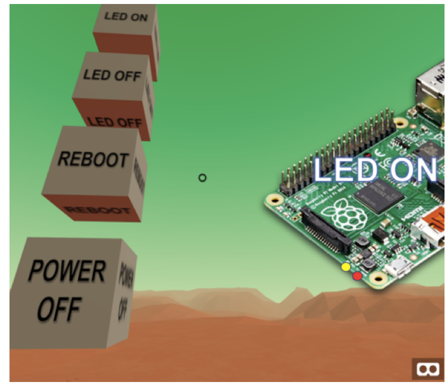

Fig. 6. Virtual Raspberry Pi – Led On

REFERENCES

[1] Cisco, "Internet of Things: Connected Means Informed", Cisco, 2016. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf. [Accessed: 12- Dec- 2017].

[2] C. Rowland, "What's Different About User Experience Design for the Internet of Things?", UX Magazine, 2015. [Online]. Available: https://uxmag.com/articles/whats-different-about-user-experience-design-for-the-internet-of-things [Accessed: 12- Dec- 2017].

[3] G.-M. Muntean, P. Perry and L. Murphy, "Objective and Subjective Evaluation of QOAS Video Streaming over Broadband Networks", *IEEE Trans. on Network and Service Management,* vol. 2, no. 1, 2005, pp. 19-28

[4] G.-M. Muntean, P. Perry, and L. Murphy, "Subjective Assessment of the Quality-Oriented Adaptive Scheme", *IEEE Trans. on Broadcasting*, vol. 51, no. 3, September 2005, pp. 276-286

[5] A. N. Moldovan, A. Molnar and C. H. Muntean, "EcoLearn: Battery Power Friendly e-Learning Environment for Mobile Device Users", *in Learning-Oriented Technologies, Devices and Networks,* A. Lazakidou and I. Omary (Eds), Lambert Academic Publishing, 2011, pp.273-296

[6] C. Peng, X. Tan, M. Gao, and Y. Yao, "Virtual Reality in Smart City," *Geo-Informatics in Resource Management and Sustainable Ecosystem*. New York, NY, USA: Springer, 2013, pp. 107–118.

[7] N. Singh and S. Singh, "Virtual Reality: A brief survey," *Int. Conf. Inf. Commun. Embed. Syst.*, 2017.

[8] P. Lelyveld, "Virtual Reality Primer with an Emphasis on Camera-Captured VR," *SMPTE Motion Imaging Journal*, vol. 124, no. 6, pp. 78-85, 2015.

[9] F. Fittkau, A. Krause, and W. Hasselbring, "Exploring software cities in virtual reality," *IEEE 3rd Work. Conf. Softw. Vis.*, pp. 130–134, 2015.

[10] A. Castellani and M. Dissegna, "WebIoT: A Web Application Framework for the Internet of Things," *IEEE Wirel. Commun. Netw. Conf. Work.*, pp. 202–207, 2012.

[11] A. Floris and L. Atzori, "Quality of Experience in the Multimedia Internet of Things: Definition and practical use-cases," *IEEE Int. Conf. Commun. Work. ICCW 2015*, pp. 1747–1752, 2015.

[12] L. Czekierda, S. Zielinski, and M. Szreter, "Benefits of Extending Collaborative Educational Cloud with IoT," *IEEE 26th Int. Conf. Enabling Technol. Infrastruct. Collab. Enterp.*, pp. 80–85, 2017.

[13] J. Huang, Q. Duan, Y. Zhao, Z. Zheng, and W. Wang, "Multicast Routing for Multimedia Communications in the Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–1, 2016.

[14] M. Di Francesco, N. Li, M. Raj, and S. K. Das, "A storage infrastructure for heterogeneous and multimedia data in the Internet of Things," *Proc. - IEEE Int. Conf. Green Comput. Commun. GreenCom*, 2012, pp. 26–33, 2012.

[15] M. Alessi, E. Giangreco, M. Pinnella, S. Pino, D. Storelli, L. Mainetti, V. Mighali, and L. Patrono, "A Web Based Virtual Environment as a Connection Platform Between People and IoT," *2016 Int. Multidiscip. Conf. Comput. Energy Sci. Split. 2016*, 2016.

[16] Z. Lv, T. Yin, H. Song, and G. Chen, "Virtual Reality Smart City Based on WebVRGIS," *IEEE Internet Things J.*, vol. 4662, no. c, pp. 1–1, 2016.

[17] M. I. Choi, L. W. Park, S. Lee, J. Y. Hwang, and S. Park, "Design and implementation of Hyper-connected IoT-VR Platform for customizable and intuitive remote services," *2017 IEEE Int. Conf. Consum. Electron. ICCE 2017*, pp. 396–397, 2017.

[18] A. Simiscuka, C. Muntean, G.-M Muntean, "A Networking Scheme for an Internet of Things Integration Platform," IEEE Int. Conf. Commun. Work. ICCW, 2017.

[19] A. Simiscuka, M. Bezbradica, G.-M Muntean, "Performance Analysis of the Quality of Service-Aware Networking Scheme for Smart Internet of Things Gateways," IEEE Int. Wireless Com. and Mob. Comp. Conf. (IWCMC), 2017.

[20] A. Simiscuka, G.-M Muntean, "A Relay and Mobility Scheme for QoS Improvement in IoT Communications," IEEE Int. Conf. Commun. Work. ICCW, 2018.

[21] A. Simiscuka, G.-M Muntean, "Age of Information as a QoS Metric in a Relay-Based IoT Mobility Solution," IEEE Int. Wireless Com. and Mob. Comp. Conf. (IWCMC), 2018.

[22] "Oculus Rift." [Online]. Available: https://www.oculus.com/rift/

[23] "Raspberry Pi." [Online]. Available: https://www.raspberrypi.org/

[24] "Beeks Beacons." [Online]. Available: http://bluvision.com/wp-content/uploads/2017/09/Specs-BEEKs-Industrial_1.pdf

[25] "Glassfish." [Online]. Available: https://javaee.github.io/glassfish/

[26] "JavaServer Faces." [Online]. Available: https://javaee.github.io/javaserver faces-spec/

[27] "A-Frame." [Online]. Available: https://aframe.io/