The Virtual Environment of Things (VEoT):

A Framework for Integrating Smart Things into Networked Virtual Environments

Jih-Wei Wu, Ding-Wei Chou, Jehn-Ruey Jiang
Department of Computer Science and Information Engineering
National Central University
Jhongli City, Taiwan, R.O.C.

***Abstract*—The key concept of the Internet of Things (IoT) is linking physical smart things to the Internet to make them active participants on the Internet. The Web of Things (WoT) uses the World Wide Web platform and its associated technologies as a framework to realize IoT. Inspired by WoT, we propose a new paradigm, the Virtual Environment of Things (VEoT), to integrate real-world smart things and virtual-world avatars/objects in a computer generated virtual environment so that entities in either worlds can interact with one another in a real-time manner. To prove the VEoT concept, we design a hierarchical smart gateway, REX (Resources Exchange), to embody smart things as RESTful resources that can be combined with networked virtual environments (NVEs) to realize the integration and the interaction of the virtual environments and the real world. In addition, we implement a VEoT-based NVE, called X-Campus, for guiding visitors to navigate a building of National Central University to verify the feasibility of VEoT. This implementation shows the VEoT framework works properly.**

Keywords—VEoT; Virtual Environment of Things; IoT; WoT; NVE; RFID; WSN; REST

I. Introduction

The **Internet of Things (IoT)** [2][3] brings us a big vision of the Internet and attracts much attention recently. The IoT regards the Internet as a network interconnecting not only computers operated by humans, but also trillions of autonomous smart things in the physical world which can store and send out information of themselves and/or their surrounding environments. The Internet thus becomes an interoperable communication framework that integrates people-to-people, people-to-things, things-to-people, and things-to-things information exchange paradigms. However, realizing the IoT vision is challenging due to the heterogeneity of smart things and the lack of standardizations of connecting the things. In view of the maturation and success of the Web technologies, the **Web of Things (WoT)** [7] employs the existing web protocols and technologies (e.g., HTTP, URL, Web Services) as the standardized way to access things in the physical world. In the WoT framework, the operations to access smart things are identical with those to access traditional Web pages. The WoT framework not only reduces the complexity of handling diversified things in the networks, but also enriches the applications of physical things by mashing up existing Web software resources.

The **Networked Virtual Environment (NVE)** [9] is an important Internet application attracting much recent attention. An NVE is a computer-generated virtual world where multiple geographically distributed users can assume virtual representatives (or ***avatars***) to walk through the virtual world and to concurrently interact with virtual objects and other avatars in real time. NVEs have been applied to in various fields, such as education, training, simulating, marketing and entertainment. Examples of NVE applications include early DARPA SIMNET and DIS systems as well as currently booming Massively Multiplayer Online Games (MMOGs). A new trend of virtual environment applications, such as Cross Reality (CR), is to exchange information between the real- and virtual-world [5][12]. A real-world smart thing has a corresponding virtual-world object, and a real-world people can be represented as a digitalized avatar in the virtual world. The people and things can interact with one another in either the real- or virtual-world interface. However, due to the diversity of virtual environment development kits and the heterogeneity of physical smart things, the ways to interoperate between virtual objects and physical things are dependent on the choice of software development kits and hardware devices. The design complexity of the mixed-real-and-virtual-world system thus dramatically increases.

In this paper, we propose a framework, called the **Virtual Environment of Things (VEoT),** to integrate real-world smart things and virtual-world avatars in a computer generated virtual environment so that entities in either worlds can interact with one another in a real-time manner. VEoT will expand the IoT applications to the domain of NVEs. The states of real-world things are presented as signals, appearances or actions of virtual objects in the virtual world; on the contrary, operating the virtual-world objects by avatars can drive the real-world smart things to change states. As WoT introduces the Web platform and related technologies into IoT, the VEoT framework introduces the NVE system and related technologies into IoT. Compared with WoT, VEoT faces more challenges, such as the real-time requirement to respond to both virtual-world objects and real-world smart things, and the adaption to various techniques of virtual environments that have not been standardized yet.

To prove the VEoT concept, we design a hierarchical smart gateway, **REX (Resources Exchange),** to embody smart things as REST-compliant (or RESTful) resources that can be combined with NVEs to realize the integration and the interaction of the virtual environments and the real world. REST (Representational State Transfer) is a way using simple protocols, such as HTTP, to create, read, update or delete objects on a server. It is recently a popular software architectural style to manage objects on a server.

REX is composed of three modules: (1) **Resource Manager** to pack all resources in the form of RESTful API compliant web services and to manage the access of virtual environment systems, (2) **Smart Thing Manager (STM)** to encapsulate the resources of smart things in virtual devices, so that the developers need not to care for the details of device controlling and network communication, and (3) **Event Manager** to notify correlative components to handle the events triggered by external environment or the states change of smart things. With the help of REX, users are able to interact with smart things to execute physical operations on them. We then implement a VEoT-based NVE guiding system, called **X-Campus**, for the E6 Building of the National Central University to verify the feasibility of VEoT. This X-Campus implementation by REX shows the VEoT framework is feasible.

The remainder of this paper is organized as follows. We elaborate the proposed VEoT framework in Section II and propose the hierarchical smart gateway, REX, to prove the concept of VEoT. In Section III, we use REX to implement a multi-user interactive NVE system based on the VEoT framework to verify the feasibility of VEoT. Finally we conclude the paper with Section IV.

Figure 1. The arichtecture of VEoT

II. THE VEOT

A. Overview

The VEoT is a framework for integrating smart things into NVEs. As WoT has introduced Web platform and related technologies to the implementation of IoT, the VEoT introduces NVE techniques to realize the IoT concept. In VEoT, avatars or objects in NVEs can interactively access real-world things in real-time by retrieving data from devices associated with things and by sending control messages to actuate associated devices of things. Base on the concept of WoT, we define an intermediate smart gateway, named **REX**, as a main component of the VEoT framework to enable the real-time interaction between real-world smart things and virtual-world avatars/objects. Fig. 1 depicts the VEoT architecture, where the smart gateway is realized by REX. REX is a universal RESTful access interface based on HTTP URL semantics to provide VEoT service for virtual-environment avatars/objects to access and interact with heterogeneous real-world smart things in real-time. Below, we introduce the RESTful access interface.

B. RESTful Access Interface

In VEoT, all smart things are embodied as RESTful resources and are presented with the URI format shown below.

http://*{Server}*/*{Object ID}*/*{Function}*/*{Field}*/*{Value}*
{Server}: the URL of the smart gateway.
{Object ID}: the unique ID of smart things. The ID of components of compound objects can be presented with hierarchical namespace and separated with ".", for example, buildinge6.floor2.lamp01.
{Function}: the item of properties or actions.
{Field}: the attribute of functions.
{Value}: the value of the attributes.

For example, the following command: http://server/buildinge6.floor2.lamp01/light/switch/true can be used to turn on the lamp with id buildinge6.floor2.lamp01. The namespace of VEoT resources can be treated as the APIs for accessing the smart things in NVEs. The Function/Filed commonly used in NVEs can refer to the most popular development kits, such as Unity3D[10], Web3D [11], etc. Through the VEoT smart gateway (i.e., REX in our implementation), NVE applications can access and control real-world smart things by using the standardized HTTP request commands, POST, GET, PUT, and DELETE, corresponding to the create, read, update and delete operations, respectively. NVE applications thus need no specific (or non-standardized) operation commands.

C. The Components of REX

REX is composed of a core process and three modules: Resource Manager, Smart Thing Manager, and Event Manager, as shown in Fig. 2. Below, we elaborate the process and the three modules one by one.

Core process: REX is handled by the core process, which coordinates the three modules to drive the system to be effective. The core process takes charge of the system initialization, arranges the functional modules and deals with the dependency of components. The core process also provides the three modules with common system functions, such as the system I/O and the file management, etc.

Resources manager: The resource manager packs all resources in the form of RESTful APIs of web services and manages the access of virtual environment systems. The VEoT developers have to prepare resource files for attached smart devices to handle the HTTP request/response data format. In addition, the developers also must prepare validation files for devices to manage the resources' access authorizations. In the system initialization phase, resources manager makes mapping and index table to combine devices with their resource files and validation files. HTTP requests come from NVE applications can be packed as per the plain text, XML or JSON format. The resources manager will parse the requests and distribute them into individual tasks sequences, then assign smart things manager to process these tasks. The HTTP responses also can be packed as per the plain text, XML or JSON format to reply to the requests.

Smart things manager (STM): STM is the key component of REX, which dominates the physical functioning of smart things and resources allocation of REX. STM groups its functions into four logical layers, in which a layer serves the layer above it and is served by the layer below it.

(1) Device Abstraction Layer (DAL): DAL describes the hardware and drivers of smart things with abstract high-level pseudo instances, which are virtual devices and are independent to the characteristics of the things. The virtual devices are capsulated as objects and can be accessed by the resources manager. STM employs a dynamic resources allocation mechanism to periodically monitor and synchronize the status of smart things.

(2) Command Access Layer (CAL): The smart things are published as Internet resources in the form of web services, which implies Internet users could access them arbitrarily. However, most of things are exclusive in operations. CAL reconciles the configurations of devices and manipulates a task queue to carry out access control of things. Furthermore, CAL handles the exceptional events detection and recovery.

(3) Device Driver Layer (DDL): As an intermediate interface, DDL configures device-specific configuration messages and deliver them to device driver or actuator of physical devices. Messages retrieved from physical devices are also reorganized to be the operable form of data, and then are transferred to the upper layer. Integrating physical devices into REX is resilient and unsophisticated. The developers just need to register the device driver to REX and configure its corresponding device-driver mapping specification.

(4) Network Abstraction Layer (NAL): Communication mechanisms between a thing/object and the REX smart gateway can be widely diverse. For example, the communication mechanism may be TCP/IP, Wi-Fi, Bluetooth or ZigBee, etc. NAL arranges communication mechanisms as universal interfaces and abstracts them from the underlying network protocols. NAL thus simplifies the task of smart things deployment.

Event Manager: The event manager handles the notification tasks while unexpected events occur or states of real-world things change rapidly. In general cases, the virtual-world objects in the NVE application periodically retrieve data from real-world things and refresh their appearance by polling. If the states of real-world things change rapidly during the polling interval, the states of virtual-world objects in the NVE will not be synchronized with those of real-world things. To solve the problem, the event manager builds two intrinsic components: (1) the subscription manager to accept and manage the event notification subscriptions, and (2) the notification manager to handle the issue of event notifications. Either real-world things or virtual-world objects can subscribe events of some specific things/objects, or publish their own events to the event manager. The event manager then connects the events producers to events consumers, and actively dispatches event notifications to subscribers.

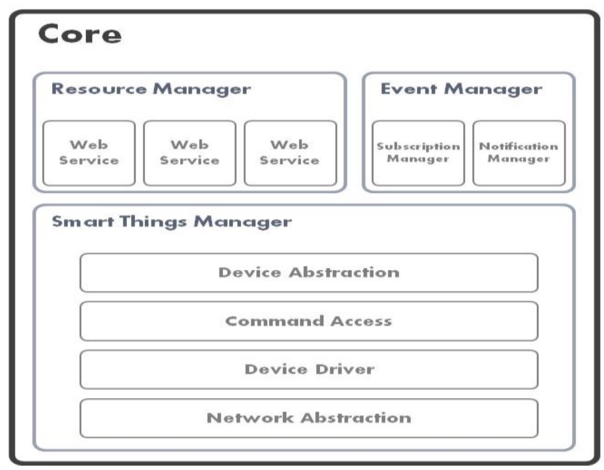

Figure 2. The components of REX

III. IMPLEMENTATION

We implemented a VEoT-based NVE, called X-Campus, for guiding users to navigate the E6 Building of the National Central University to verify the feasibility of the VEoT framework. In X-Campus, a virtual environment is established with the Unity 3D authoring tool for the exterior appearance, the first floor and the second floor of the real-world building. Fig. 3 shows the real-world scene and two screenshots of the virtual environment of X-Campus.

We realize the VEoT smart gateway (i.e., REX) on a web server that runs the Apache and Tomcat system. Many smart things are deployed in the physical world; they are electronic lamps, each equipped with a power supply and an on-off actuator, and an Octpus II, a wireless sensor deveped by National Central University and National Tsing-Hua University (see Fig. 4). The Octpus II sensor carries a micro processor, a ZigBee communication component and some sensors to sense physical phenomena, such as the temperature, humidity, light intensity, etc. The Octpus II sensor attached to the lamp runs on top of the TinyOS to perform certain tasks. The lamp is thus considered as a smart things, since it is capable of computing, sensing, communicating and being actuated.

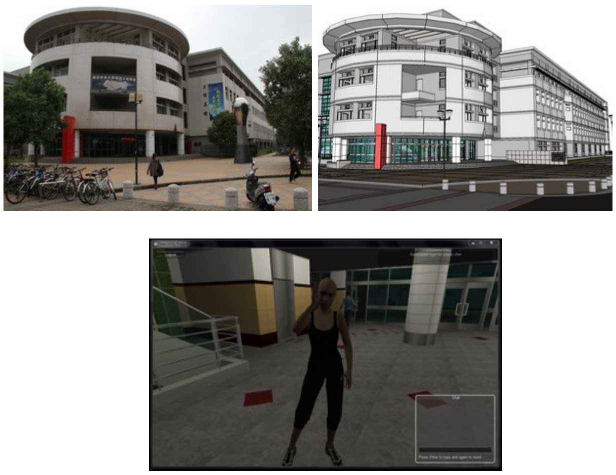

Figure 3. The real-world scene and the virtual environment of X-Campus

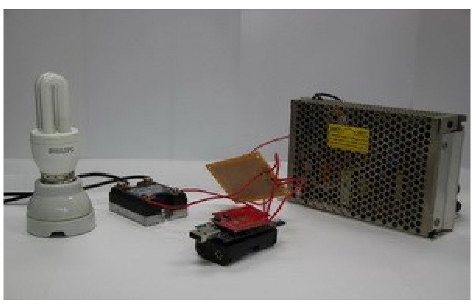

Figure 4. A real-world smart thing, a lamp, in X-Campus

The implemetation of X-Capmus using REX based on the VEoT framework works well. The virtual-world avatars/objects interact smoothly with the real-world smart things. Please refer to [1] for demonstration video of X-Campus and related information.

IV. CONCLUSION

The applications of IoT receive more and more attention in recent years. Many experts even believe that IoT is the most important core technology of the next-generation Internet. The Web of Things (WoT) uses the World Wide Web and its associated technologies as a platform for accessing, searching, and indexing smart things in the Internet. Inspired by WoT, in this paper we propose a paradigm, the Virtual Environment of Things (VEoT), to integrate real-world smart things and virtual-world avatars/objects in a computer generated virtual environment so that entities in either worlds can interact with one another in a real-time manner. In addition, we also design a hierarchical smart gateway, REX, to embody smart things as RESTful resources that can be integrated into virtual environments.

We implement a VEoT-based NVE, called X-Campus, for guiding visitors to navigate a building of National Central University to verify the feasibility of VEoT. With the VEoT framework, the interactivity and operability between the virtual space and the real world are satisfied in our experiment.

In the future, we plan to conduct more complicated and larger-scaled implementations to reinforce the robustness of the VEoT framework. We also plan to address the common open issues on IoT, such as the security, and applicability on mobile devices, as well as the plug-and-play capability of smart things [6][14]. Based on the concept of VEoT, there are some potential research issues we could concern: (1) How to make VEoT scalable? The capacity of the server running REX will restrict the performance of the system. To make VEoT scalable, we can run REX on a cloud centric platform [6]. (2) How to utilize the Machine-to-Machine (M2M) communication [4][8][13] in VEoT? M2M communication is a hot IoT research topic. We can integrate the M2M gateway function into the smart gateway to take advantage of M2M communication.

REFERENCES

[1] ACN Laboratory, X-Campus Demonstration Video, URL: http://in2.csie.ncu.edu.tw/~jrjiang/XCampus/xcampus_pc.wmv

[2] W. K. Ashton, “That 'Internet of Things' Thing,” *RFID Journal*, 22 July, 2009.

[3] L. Atzori., A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, Vol. 54, Issue.15, pp. 2787-2805, 2010.

[4] M. Chen, J. Wan, and F. Li. “Machine-to-machine communications: architectures, standards, and applications.”, *KSII Transactions on Internet and Information Systems*, Vol.6, No.2, pp.480-497, Feb 2012.

[5] B. Coleman, “Using sensor inputs to affect virtual and real environments,” *IEEE Pervasive Computing*, Vol. 4, No. 3, pp. 16-23, 2009.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, Vol. 29, pp. 1645-1660, 2013.

[7] D. Guinard and V. Trifa, “Towards the web of things: Web mashups for embedded devices,” in *Proc. of the Second Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web* (MEM '09), 2009.

[8] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, D. Boyle:” From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”. *Elsevier*, 2014.

[9] S.-Y. Hu. and G.-M. Liao, “Scalable peer-to-peer networked virtual environment,” in *Proc. of 3rd ACM SIGCOMM workshop on Network and system support for games* (NetGames 2004), 2004.

[10] http://www.unity3d.com/

[11] http://www.web3d.org/x3d/

[12] J. Lifton et al. “Metaphor and Manifestation Cross-Reality with Ubiquitous Sensor/Actuator Networks,” *IEEE Pervasive Computing*, Vol. 8, No. 3, pp. 24-33, 2009.

[13] M. Liu., T. Leppanen. E. Harjula, Z. Ou, A. Ramalingam., M. Ylianttila., and T. Ojala. “Distributed resource directory architecture in Machine-to-Machine communications,” *International conference on IEEE WiMob 2013*, 7-9 Oct, pp.319-324, 2013.

[14] O. Said and M. Masud. “Towards Internet of Things: Survey and Future Vision,” *International Journal of Computer Networks*, Vol. 5, Issue. 1, 2013.