BRIDGING THE GAP BETWEEN VIRTUAL AND REAL WITH SECOND LIFE CLIENT IN A VIRTUAL HOME AUTOMATION SYSTEM

SK Alamgir Hossain, Abu Saleh Md Mahfujur Rahman, Abdulmotaleb El Saddik

Multimedia Communications Research Laboratory,
University of Ottawa, Ottawa, Canada
Email:{skahossain, kafi, abed}@mcrlab.uottawa.ca

ABSTRACT

In this paper we propose the design and development of a prototype system to monitor and control a physical home and to bridge the interaction gap between the virtual and real world device control mechanism. We created a virtual 3D home in Second Life that mimics the look of a physical home. In the physical home environment different devices and sensors are connected in order to ensure a safe and automated home. Any event that occurs in the physical space of the smart home is then synchronized with the virtual environment. More importantly, the virtual home interface provides the option to control the physical smart devices. By using the Second Life virtual interface the home owner have a better look to monitor or control the home appliances. As per the initial experimentation, we found out that the proposed approach of Second Life based home automation system is appealing and useful to the user. Some of the other possible applications of the proposed system are public space surveillance, child monitoring scenarios etc.

Index Terms— Smart Home, Second Life, 3D Virtual Environment, Surveillance, Virtual Reality

1. INTRODUCTION

A 3D virtual world is a computer based simulated multimedia environment usually running over the Internet where a user can access and interact with the other users by using a 3D virtual persona called avatar. Second Life[1] which is developed by Linden Lab is one of the most popular 3D virtual environment that acts as a medium of social interaction. Here people can build their virtual 3D home, customize it with 3D furniture and populate that with 3D interactive devices. With the growing popularity of Second Life, users often design their virtual 3D homes mimicking that of their real homes. Our proposed method can be used to bridge the gap between the real home and the respective 3D virtual home in Second Life by incorporating services that control and synchronize the communication events to and from the smart devices. In this approach, through the interaction of the virtual home elements the user can intuitively maneuver the real world devices in a smart environment. In smart environment different wired and wireless sensory devices such as temperature, humidity, pressure sensors etc. are installed. By using these sensory data it is possible to measure different environment conditions and make decisions. The prices of these sensors and automation devices such as X10[2], WiFi are decreasing with time. At the same time, their usage to provide control and various entertainment facilities in smart spaces are becoming hugely popular. A graphical user interface (GUI) which is a medium to interact with programs plays an important role especially to control physical space elements like smart home. However, controlling a physical space by using the traditional icon based GUI might not be sufficient for effective management of a range of smart physical devices. In order to efficiently control different intelligent physical devices, researchers have placed attention on the intuitive GUI design and explored its usability issues [1] [2].

Collaboration between physical and virtual world is not new. Fleck et. al [3] presented the first notable work for controlling the physical space monitoring by using a virtual user interface. The authors presented an integrated 3D world model, where the physical objects were collected through inter-camera tracking system and embedded as live texture in the 3D world. Presumably the interface minimized the overhead of the observing person who was serving the system to understand the events that happened in the real environment. Another notable work related to Augmented Virtual Environment which fused dynamic imagery of the real world object into the 3D models is [4] in which authors presented the technical challenges about moving object detection, tracking and 3D visualization for effective dynamic event comparison. Many previous works found the adoption of virtual reality technologies in educational practices often presented problems that addressed the difficulties in navigation [5] while using 3D interfaces. Users risked to spend a lot of time to control their movements, to reach to a specific location or to gain the required point of view. User interface is not only im-

[1]Second Life, http://secondlife.com

[2]X10, http://www.x10.com/homepage.htm

CCECE 2011 Niagara Falls, Canada
978-1-4244-9789-8/11/$26 ©2011 IEEE

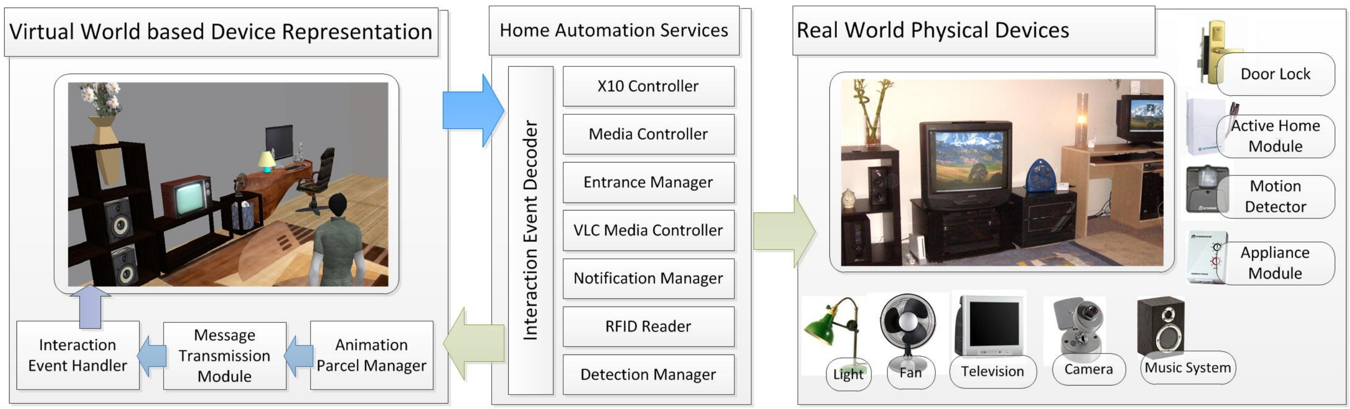

Fig. 1. An overview of the SOA based Second Life home automation architecture. Each physical device has a virtual 3D representation in the Second Life where user can control the physical devices through the virtual objects as well as the physical devices can change the animation by using the home automations services.

portant for efficient control of the system but also it depends on users' ages, intelligences. In the paper Bin Zhang et. al. [6] showed how those factors affect the design of user interface. Those factors are very important and have a high impact in users' cognitive task inside a home. The importance of 3D user interface (e.g. 3d virtual world) over a 2D interface were perfectly described in [1], in which user shows that how a 3D user interface can improve intuitively and adaptivity of smart home control instead of ordinary 2D user interface.

Also simulating real world environment in the 3D virtual world similar to Second Life is also popular now. Recently University of Arkansas installed a prototype system called Virtual Razorback [7] in Second Life which is a virtual hospital for modeling certain aspects of health care and a hospital's supply chain, from receiving goods to patient care. The virtual hospital has virtual hospital furnitures and equipments like a real world hospital. Moreover, in order to navigate and control the 3D virtual environment based interfaces user often requires training. However, as the Second Life users are already adapted, the users have almost no learning with the interface. Moreover, flexible authoring capabilities of 3D objects [8] [9], built in security features, scalability and portability feature make Second Life as the most popular 3D virtual environment. Currently it has 22 million active subscribers as of February 2011 [10]. In this paper we propose the design and development of a flexible virtual smart home like [1] [2] automation system for controlling those real life devices through a usable 3D user interface.

Our contribution in this paper is two-fold. First, in order to bridge the gap between virtual and real, we present a Second Life Client add-on, where we provide flexible SOA based architecture for controlling the physical space. Second, we incorporate virtual annotation mechanism for the Second Life object for physical space monitoring.

The rest of this paper is organized as: at first, in Section 2 we illustrate the various components of our proposed system and provide a general overview of the system. Further in Section 3 we describe the implementation issues, development challenges of different modules and some evaluation strategies. At the end we provide conclusion of the paper in Section 4 and state some possible future work directions.

2. SECOND LIFE HOME AUTOMATION SYSTEM

In this section we present the Second Life based home automation system architecture and the different components of the system. At first in Section 2.1 we illustrate the architecture of the proposed system. Further in Section 2.2 we describe the object annotation mechanism in Second Life.

2.1. General Architecture

Our system comprises of three primary components that are Second Life Viewer plug-in, home automation web services and the point-to-point physical connection between services to the wireless and/or wired devices. In order to communicate with the core part of the second life client program called *viewer* we developed a Second Life interaction event handler as an add-on to the viewer. This coupling architecture provides the option to listen to the communication channel of the Second Life without affecting the functionality of the core system. Furthermore, the event controller interacts with the other parts of the home automation or surveillance system by using Simple Object Access Protocol (*SOAP*). Interaction event handler receives and/or transmits messages to/from message transmission module and captures the events that are generated from the message transmission module.

The event handler performs actions by using text based messaging protocol. A message contains event trigger data, animation data or simple communication data. The Message Transmission Module captures all the messages that are generated in the core section of the Second Life. When through the avatar the user issues events in the 3D environment, e.g., *a click event to open a door*, the message transmission module captures those events and transfers the event messages to the nearby interaction event handler for processing. The event handler module determines the particular event handling routine for a specific event and then packs the event-handling message with the handling routine. The handler then sends the packet to the interaction event decoder to communicate with the actual home automation services. Message transmission module also receives animation data from the animation parcel manager and generates animation sequence for the objects in the 3D virtual world.

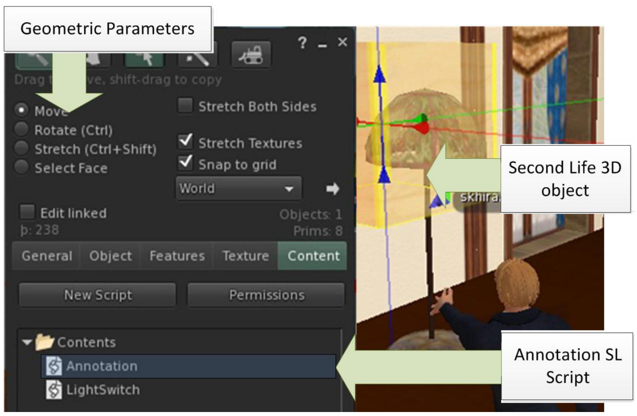

Fig. 2. The flexible object annotation scheme allows the user to annotate object and allow setting animation properties. When interacted by home users or automated triggering, the users see animation rendering on the screen.

2.2. Home Annotation in Second Life

In order to send or receive synchronized communication to/from the real or the Second Life viewer we create 3D virtual objects in Second Life for each of the physical devices that want to control. Further, we define annotation rules into these 3D objects and later on attach the annotated files to the object. The technique are described further in the following sections in details.

2.2.1. Object Annotation

In our system we annotated Second Life 3D objects and specified the corresponding physical device addresses. Every object in the real life has a unique id by which smart home services can control it. Physical devices connected with the home services by using *X10* or *WiFi* connections. The 3D objects are annotated using the Second Life's built-in script annotation mechanism that is demonstrated in Figure 2, where the 3D lamp is annotated with the help of a separate annotation file. The annotation file is an XML file that specifies the animation unique identifier (*UUID*), the animation file (*BVH*), the speed of animation, duration of animation etc. This file also specifies the physical device specific data like device address, name, type etc. The *BVH* or Biovision Hierarchical is a file format contain motion capture data and actively used in computer animation. *BVH* is developed by BioVision Inc.

Figure 3 depicting a sample XML specification that demonstrates a door-open-close scenario where two animations *"doorOpen1"* and *"doorClose1"* animations are specified for open/close the real door that is connected with a *X10* appliance module. Here the device address is also specified which is *"A1"* in this case.

```
<?xml version="1.0" encoding="utf-8"?>
<AnnotationRules>
  <animationModules >
    <animationModule id="doorOpen1">
      <UUID>6b61c8r8-6878-0h87-12hg-j1hj3h4us8g9</UUID>
      <animationBVH>doorOpen.bvh</animationBVH>
      <animationLooped>NO</animationLooped>
      <animationSpeed>30</animationSpeed>
      <animationSpeed>LOW</animationSpeed>
      <animationScaleTo>1</animationScaleTo>
    </animationModule>
    <animationModule id="doorClose1">
      <UUID>6f51c8r8-4555-0h87-89dw-a02j4b49sk2b</UUID>
      <animationBVH>doorClose.bvh</animationBVH>
      <animationLooped>NO</animationLooped>
      <animationSpeed>30</animationSpeed>
      <animationSpeed>LOW</animationSpeed>
      <animationScaleTo>1</animationScaleTo>
    </animationModule>
  </animationModules>
  <Appliances>
    <Appliance name="Main Door 1">
      <ApplianceType>Door</ApplianceType>
      <DeviceType>X10</DeviceType>
      <PhysicalAddress>A1</PhysicalAddress>
    </Appliance>
  </Appliances>
 </AnnotationRules>
```

Fig. 3. An overview of the target object specific interaction rules stored (and could be shared) in an XML file.

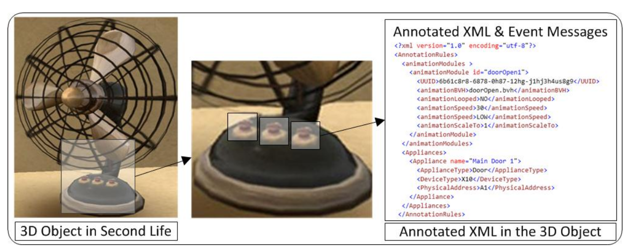

Fig. 4. Multi-region annotation in a 3D fan and specified a physical lamp address which is connected to a X10 or Wi-Fi.

In the annotation phase we specify the LSL scripts[1] using which the Second Life viewer obtains control of the objects and performs read/write operation into its communication channel. A sample LSL script for the door example is

[1]LSL Portal, http://wiki.secondlife.com/wiki/lsl_portal

depicted in Figure 5. In order to get refined response and to provide high flexibility to control the physical objects our annotation scheme introduces a generic interface. This is especially important for those devices in which different parts of a 3D virtual device needs different annotation rules. For example, a table fan have different switch for speed control whereas a lamp has different illumination intensity levels or a temperature sensor have different temperature levels. In order to provide this type of facilities we developed a suitable annotation mechanism that allows annotation into different parts of the 3D virtual object. In this manner we specify rules in the XML data that we need to attach with specific region of the object. This type of multi region annotation scheme is depicted in Figure 4.

```
state_entry()
{
    //Request animation permission to Second Life
    llRequestPermissions(llGetOwner(),PERMISSION_TRIGGER_ANIMATION);
    llSetTimerEvent(1.0);

    //Specify communication channel
    llListen(&iChannel,"",llGetOwner(),"");
}

start_animation(Object annotatedXML, string id)
{
    //load animation object from attached XML data
    loadAnimationModuleFromXML(id, &animationObj);

    //Render animation
    llStartAnimation(llList2String(this, animationObj));

    //popilate the device object list from attached XML data
    applianceObjList = getApplianceModuleFromXML();

    //write data to the channel to send the message
    Write_channel(llList2String(null, applianceObjList));
}
```

Fig. 5. Code snipped of SL Script to load the XML data to send or receive messages into the communication channel.

2.2.2. *Device State Synchronization*

In order to render synchronized animation in the virtual environment as per the state changes of the real device, we incorporated state based event communication mechanism. When a user clicks or touches a 3D object in the virtual environment the system received the event message which was then saved into an event queue. Further, the system processed the event queue sequentially and transmitted the message respectively to the home automation service. At this stage the physical device became activated to handle the input message. In this case if the device failed to initiate its services then an error message was displayed, otherwise it sent $ready$ or Acknowledgment (Ack) message.

In the next step the system sent the actual message, such as *open door, close door, increase light intensity to high, decrease sound system volume to low* etc. At this time the smart home service received the message from our developed Second Life plug-in. Afterwards, it sent the command message to the specific physical object *X10* or *WiFi* module at the same time home service sent to the plug-in to play a synchronized animation in Second Life environment that were specified into the annotated XML file. This type of animation gave a realistic intuition to the user. When the physical object tried to close the device or went to reset it's state the smart home service received this updated state and sent the confirmation to the Second Life plug-in to stop/resume the animation as the physical device changed its sate. This state based interaction scenario are illustrated in Figure 6.

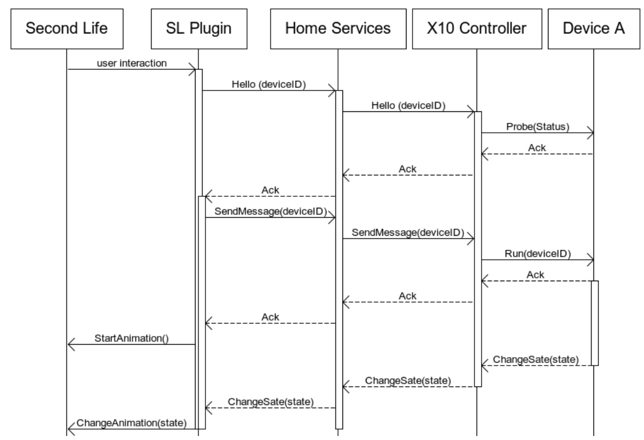

Fig. 6. Proposed system interaction diagram to control a physical device by using the SL Plug-in.

3. IMPLEMENTATION AND RESULTS

We implemented our proposed system by using Microsoft Visual Studio 2005 IDE and the primary language used was Visual C++. In order to develop the Second Life add-on, we locally build the Second Life open source viewer called *Snowglobe*[1] by using the latest version of *CMake*[2]. For the home automation service we used *ActiveHomeScript* Library and web services dynamic discovery (*WS-Discovery*) which is a multicast discovery protocol defined with the purpose of allowing dynamic discovery and advertisement of target services willing to find a specific service can query the network using multicast search messages, and services satisfying the query should reply.

In our prototype we also used web services eventing (*WS-Eventing*) which is a web service protocol that describes how a client (subscriber) can register to some events of a web service (event source). Thus, changes in the service can be notified to any client without requiring standard polling mechanism. At the beginning when any device wants to connect with the web service needs to send multi cast message to search target services by type or within a scope. Then a unicast response to the sending client when the target service

[1]Snowglobe, http://wiki.secondlife.com/wiki/Snowglobe
[2]CMake, http://www.cmake.org/cmake/resources/software.html

matches a Probe message. In this way when the connection was established successfully then the device needs to subscribe an event to send/receive message to the server.

We calculated the transmission time from the sender machine to the smart home service by using Equation 1. Where user's average interaction time to interact with the Second Life (SL) viewer is I unit, average data transmission rate via the server is Π, n is the message size and the time for sending data from the receiver machine to the home service is β_1 unit.

$$R = I + \frac{n}{\Pi} + \beta_1 \quad (1)$$

After generating an interaction the system approximately requires $R = (3775 + 270 + 344)\, ms$ to complete the transmission. Here, in our experiments the average of the interaction time is $3775ms$, network overhead is $270ms$ and β_1 is $344ms$.

For the usability test we have incorporated the usability evaluation guidelines [11] to qualitatively measure our proposed system and designed our tests accordingly with the sensory analysis [12] of the system involving both the user and the targeted sensory communication. Before performing the usability test we designed a test plan where we defined our evaluation objectives, developed questions for the participants, identified the measurement criterion and decided upon the target users of the system. The test took place at our university laboratory with ten (10) participants comprising of different age groups.

Table 1. Usability test questions to the user.

#	Question
Q1	Perceived system response was acceptable
Q2	The device response are realistic and/or acceptable
Q3	Consider using the system in 3D Virtual Environment like Second Life
Q4	Perceived delay between real object response and animation rendering was tolerable
Q5	Easy to get familiar with

The average of the responses of the users were calculated in percentage form and measured after the usability tests. It is worth mentioning that more than 78% of the users would like to control their physical space using the enhanced system through virtual environment like Second Life. 10% users like to use the system through their mobile devices instead of from the PC. Overall the users were also satisfied with the synchronized animation and device responses and 75% of users consented to that.

4. CONCLUSION

In this paper we presented a Second Life prototype system to monitor and to control a physical home and to bridge the interaction gap between the virtual and real world device control mechanism. The developed system worked as an add-on and loosely coupled to the Second Life viewer. The animation and device control data were annotated in the virtual 3D object in Second Life. The 3D object representing a real device received inputs when interacted by a user and automatically responded with the state changes in the physical environment. We presented the implementation details of a preliminary prototype exploring the aforesaid Second Life based interactions in a real-virtual collaborative environment. In future we want to incorporate sound and gesture based device control mechanism to provide intuitive interaction capabilities to the user in order to effectively control the Second Life viewer.

5. REFERENCES

[1] L. Borodulkin, H. Ruser, and H.-R. Trankler, "3d virtual "smart home" user interface," in *IEEE International Symposium on Virtual and Intelligent Measurement Systems VIMS '02*, 2002, pp. 111 – 115.

[2] P. Rashidi and D. Cook, "Keeping the intelligent environment resident in the loop," in *2008 IET 4th International Conference on Intelligent Environments*, 2008, pp. 1 –9.

[3] S. Fleck, F. Busch, P. Biber, and W. Straber, "3d surveillance a distributed network of smart cameras for real-time tracking and its visualization in 3d," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, 2006, pp. 118 – 118.

[4] I. O. Sebe, J. Hu, S. You, and U. Neumann, "3d video surveillance with augmented virtual environments," in *First ACM SIGMM international workshop on Video surveillance*, ser. IWVS '03. New York, NY, USA: ACM, 2003, pp. 107–112.

[5] C. Youngblut, "Educational uses of virtual reality technology. technical report no. ida document d-2128," Institute for Defence Analyses., Tech. Rep., 1998.

[6] B. Zhang, P.-L. P. Rau, and G. Salvendy, "Design and evaluation of smart home user interface: effects of age, tasks and intelligence level," *Behaviour and Information Technology*, vol. 28, no. 3, pp. 239–249, 2009.

[7] C. Thompson and F. Hagstrom, "Modeling healthcare logistics in a virtual world," *Internet Computing, IEEE*, vol. 12, no. 5, pp. 100 –104, 2008.

[8] S. A. Hossain, A. S. M. M. Rahman, and A. El Saddik, "Interpersonal haptic communication in second life," in

Haptic Audio-Visual Environments and Games (HAVE), 2010 IEEE International Symposium on, 2010, pp. 1 –4.

[9] A. S. M. M. Rahman, S. A. Hossain, and A. El Saddik, “Bridging the gap between virtual and real world by bringing an interpersonal haptic communication system in second life,” *Multimedia, International Symposium on*, pp. 228–235, 2010.

[10] L. Lab, “Second life, online user count,” http://secondlife.com/xmlhttp/secondlife.php, Tech. Rep., Last accessed, Feb 2011.

[11] D. Chisnell, “Usability testing: Taking the experience into account,” *Instrumentation Measurement Magazine, IEEE*, vol. 13, no. 2, pp. 13 –15, 2010.

[12] E. Kukula, M. Sutton, and S. Elliott, “The human biometric-sensor interaction evaluation method: Biometric performance and usability measurements,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 4, pp. 784 –791, 2010.