Presented at Intelligent Environments 2014, Shanghai Jiaotong University, China 2-4 July 2014

Towards MMO Intelligent Environments

Marc Davies, Vic Callaghan
Intelligent Environments Group (IEG)
University of Essex
Colchester, UK
{midavi, vic}@essex.ac.uk

Daniyal Alghazzawi
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia
dghazzawi@kau.edu.sa

***Abstract*—Massively Multiuser Online (MMO) Computer Games arguably provide some of the best examples of large groups of users, potentially distributed all across the world, coming together to interact with each other in a single common environment. Depending upon the scenarios presented by the game designers, such user interaction may consist of countless individuals each operating with their own set of priorities and goals. Alternatively, other situations may require the collective collaboration of multiple users, each performing a role in order to achieve a larger overall objective. This paper outlines some initial research for an investigation taking inspiration from some technologies and frameworks used by MMO Computer Games, to create a new global network of distributed Intelligent Environments. This research has been performed as a part of the ongoing ScaleUp project, focussing on creating smart hybrid systems to increase the functionality within Intelligent Environments.**

Keywords- computer games, intelligent environments, middleware, pervasive computing, virtual worlds

I. Introduction

A. From Vision to Virtuality

For some time now research projects focused around Pervasive or Ubiquitous Computer Science have been presenting research dedicated to building novel device designs or architectures that can be used to create Intelligent Environments. In more recent times, much of this research has since been used by the commercial sector to develop new 'smart' products that can be sold to consumers, either as standalone devices or augmentations to pre-existing technologies. In turn, the widespread uptake of these 'smart' products has led to the foundation or extension of several research areas within Computer Science, such as Big Data analysis, the Internet of Things and to a lesser extent Web Security.

Another area of Computer Science that has significant influence in the commercial sector is computer games and their related technologies. Like Intelligent Environments research, over the last couple of decades this field has seen some enormous advancements, perhaps most notably in the area of Computer Graphics. However, it is not only internally that computer games have seen advancement. With the advent of Cloud Computing an entirely new global distribution framework emerged, (e.g. Valve's Steam engine [10] and EA Games' Origin [8]). Even before this, Massively Multiuser Online (MMO) games have allowed millions of registered users located around the world to simultaneously inhabit expansive virtual worlds, full of different interaction points and stimuli, not to mention encounters with other users. Furthermore, this trend only looks set to increase in popularity as new games are developed with budgets similar to typical Hollywood blockbuster movies. This in turn is likely to prompt further advances in the underlying technologies as developers and hardware manufacturers seek to get their products noticed by customers in a highly competitive marketplace.

B. MMO Computer Games & Intelligent Environments

Several similarities exist between typical examples of MMO Computer Games and Intelligent Environments. For example, the functionality of both is often largely dependent upon user interactions and reactions to some stimuli, providing data for control software to process. The actions of individual users are monitored in real-time and then used by an agent or some other Artificial Intelligence program to calculate an appropriate response. These programs could possess learning capabilities, to gain knowledge about the preferences of users and subsequently adapt to their behavioural patterns. In an intelligent household this may be an agent learning the temperature or brightness levels specific inhabitants prefer a room to be. In a computer game this learning might exhibit a more negative outcome, such as an Artificial Intelligence learning the preferred tactics used by a player during battles and then taking steps to counter their future attacks. Furthermore, depending upon the scenario, developers of both MMO Computer Games and Intelligent Environments may need to consider the extent to which differing actions, priorities and opinions of multiple users sharing a common space can affect the environment as a whole. So given these similarities, how can MMO Computer Games benefit Intelligent Environments?

To a large extent, MMO Computer Games already possess many of the technological features and architectures potentially desirable, (but not necessarily readily available) to computer scientists researching Intelligent Environments and other related fields. This includes features such as;

- Global client-server based networks each capable of simultaneously hosting millions of registered users either in a single environment or by running different isolated instances of a common world.

- Robust authentication and information security systems to protect the system functionality and user privacy.
- Cloud-based hosting and distribution platforms for running and sharing software programs, plus archiving important operational data.
- The immersive properties of a computer game, where, if programmed correctly, players can quickly feel as if they become a part of the virtual world they inhabit.
- The ability to create and customise potentially large-scale, highly detailed bespoke environments. Developers can have complete control over factors such as the included content, functionality, accessibility and external influences, (e.g. climate and temporal changes), within the generated space.

Each of the features listed above have been developed and refined by the computer games industry over twenty-five years of research. The resulting technologies have been successfully deployed on a global scale and now regularly see thousands of users each day, proving their robustness in both handling large amounts of data and connected clients.

Earlier we highlighted some of the similarities MMO Computer Games can share with Intelligent Environments. However, in contrast, this proven global connectivity of MMO Computer Games highlights one of the most significant differences between the two systems. Often Intelligent Environments are created but only within a local domain. Even in cases such as a smart university campus or an office building containing multiple rooms, control over the intelligent functionality is likely to be restricted to a stand-alone LAN or WAN, with firewalls being used to block outside access. In the long term, one of the aims of this work is to design a system capable of representing multiple distributed Intelligent Environments and their contents via a single virtual space built using pre-existing MMO Computer Game technologies. The new system will allow users to access, observe and control, (assuming they have usage rights) sensors and devices in any of the connected environments from anywhere on the globe with access to the Internet. It is hoped that this work will allow researchers of Intelligent Environments, and also potentially other fields of Computer Science, (such as computer games), to benefit from the augmented functionality the new system will provide.

II. Background Research

Past research projects have investigated how computer games technologies can be used to augment Intelligent Environments and vice versa. In one instance, user data was gathered from Intelligent Environments to be used when designing Artificial Intelligence systems for computer game NPCs [3]. Another investigation created a Mixed Reality Intelligent Environment by augmenting a smart household with a virtual component modelled using a virtual world system based upon the online community Second Life [2]. There have also been projects where researchers used virtual worlds as a means of visualising the state of devices in an Intelligent Environment to users [1]. Each of these systems allowed users to remotely control networked intelligent devices through the virtual component and receive sensor feedback from the physical environment. However, even though some of the computer game technologies used included MMO capabilities, these features were not used. In each case, the environment implementations were locally deployed, with the virtual component being accessible via a LAN.

Computer Games technologies have also been used within Intelligent Environments research as a means of engaging users. For example, by following typical trends set by computer games, users can be presented with a familiar style of graphical interface and a series targeted activities with set goals for them to work towards. Liu et al used such an approach to engage users in a crowd-based application designed to encourage people to lower their produced CO^2 emissions [6].

Other research projects have used professional computer game graphics engines to devise simulations of real Intelligent Environments, to provide demonstrations or preliminary testing facilities for agent designs and device configurations. For example, in their project, Jia et al use Unity 3D [9], (a downloadable graphics engine graphics engine used in the development of many commercial Indie game titles), to create a virtual world, duplicating a real room containing an intelligent lighting system [5]. Using virtual counterparts to the real intelligent devices they were able to test various light-maps within the Unity 3D world. The light generation and shadowing provided by the game engine allowed the research team to observe performance using near photorealistic levels of detail. Through their virtual world the team were able to evaluate and refine their lighting controller map patterns to create bespoke configurations suiting various different scenarios.

Each of the projects mentioned above used computer games or related technologies in different ways to create functionality that promoted or enhanced the capabilities of an Intelligent Environment research system. Several different types of Intelligent Environment were also covered, including household spaces, teaching or meeting rooms and outside locations. Several different levels of reality were also covered by these projects. Some systems existed completely in the virtual domain, while others utilised Mixed Reality architectures [7]. However, each of the aforementioned systems were also designed to only operate independently in a closed network environment, meaning they were each limited to using the functionality included in their respective designs.

III. MMO Inteligent Environments

The features of MMO computer games particularly interesting to this work are the aforementioned global usage and distribution networks. Currently in a game-based environment connected to an MMO network via the Internet, a client-server based architecture is typically used allowing registered users to access the system via their local computer, smart-phone, tablet or console machine. However, what if that client hardware was replaced by an entire Intelligent Environment, (be that a building, room or some other augmented arena), complete with its embedded devices, sensor array and any installed agent programs? Potentially, a vast network of real Intelligent Environments (located around the world) could be created and linked together via a shared space that replaces the common computer game used in a standard MMO system.

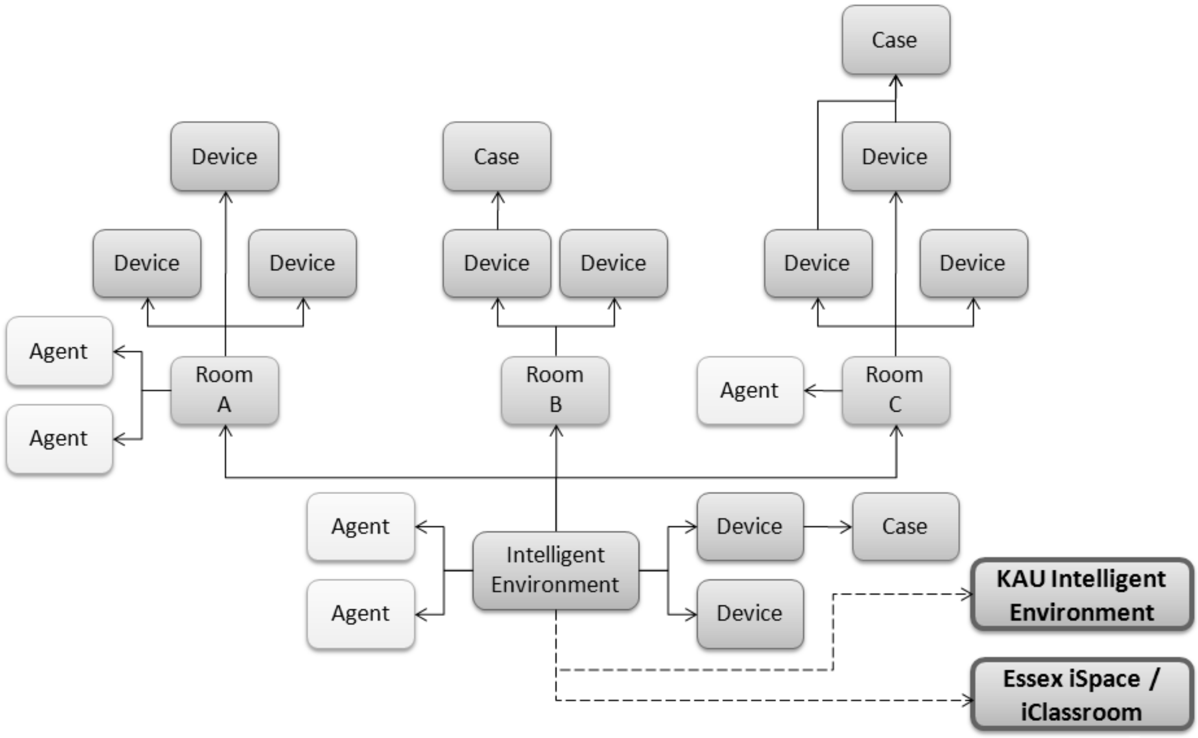

Figure 1. An example of the MMO Intelligent Environment architecture connecting multiple distributed spaces

By taking further inspiration from computer games the MMO Intelligent Environment network could be further augmented to include additional functionality such as;

- Social groups of several Intelligent Environments could be created to work collaboratively in order to achieve some higher level of functionality. This is similar to a group of gamers forming a 'party' to complete an assigned quest.
- Sensor arrays installed in the networked Intelligent Environments could be set to a 'public' mode and allow anybody or any agent to access them and receive real-time data updates.
- Users could download designs for agent programs from a Cloud storage system that was developed in one Intelligent Environment and subsequently shared with all others on the network. The agent could potentially be presented in an App like format to allow quick installation and provide automatic performance feedback to its creators.
- UPnP and similar control platforms allow Intelligent Environments to be accessed and smart devices or sensors to be controlled via a network, but generally this system is limited to a local area, perhaps with a remote smart-phone or web-interface portal system providing some level of limited remote control. By basing MMO Intelligent Environments on existing computer games technologies, UPnP and other control platforms could all potentially be represented using a single interface common to every environment, rather than each having a bespoke design. Furthermore, game-based authentication systems could be incorporated to restrict access only to authorised users or owners and provide added security.

As part of the new MMO Intelligent Environment system, an interactive user interface platform visualising the entire global Intelligent Environment network will be created. This interface design will be based around several existing and well-established technologies used in the creation of MMO computer games, including the Unity3D engine. The new interface is intended to be usable on several different levels to accommodate variations in the priorities of separate research groups and their projects. The system will provide an overview of the entire global network, but also optionally allow users to focus in on specific Intelligent Environments. Within each environment representation, single and collections of deployed smart devices and sensors will be viewable in addition to any local agent programs installed. Users with the correct authorisation will be able to make ad-hoc modifications to the general operation of the environment, such as changing the agent programs currently running or altering the state of connected smart devices.

The architecture example shown in Figure 1 provides a more detailed breakdown of the hierarchy that will be used by the MMO system. Notice how different agent programs running in the example environment can be visually categorised by the room(s) in which they operate. Alternatively, the agents attached directly to the 'Intelligent Environment' node are considered to be affecting the entire space rather than a specific room. Intelligent devices are also categorised in the same way, with those attached directly to the Intelligent Environment node representing functionality such as an external weather monitoring station that can be accessed from any room. If some level of user identification or tracking system was installed in the Intelligent Environment, it is also possible to include any inhabitants themselves in the MMO architecture, represented on the same level as devices and agents. Furthermore, specific higher-level functionality

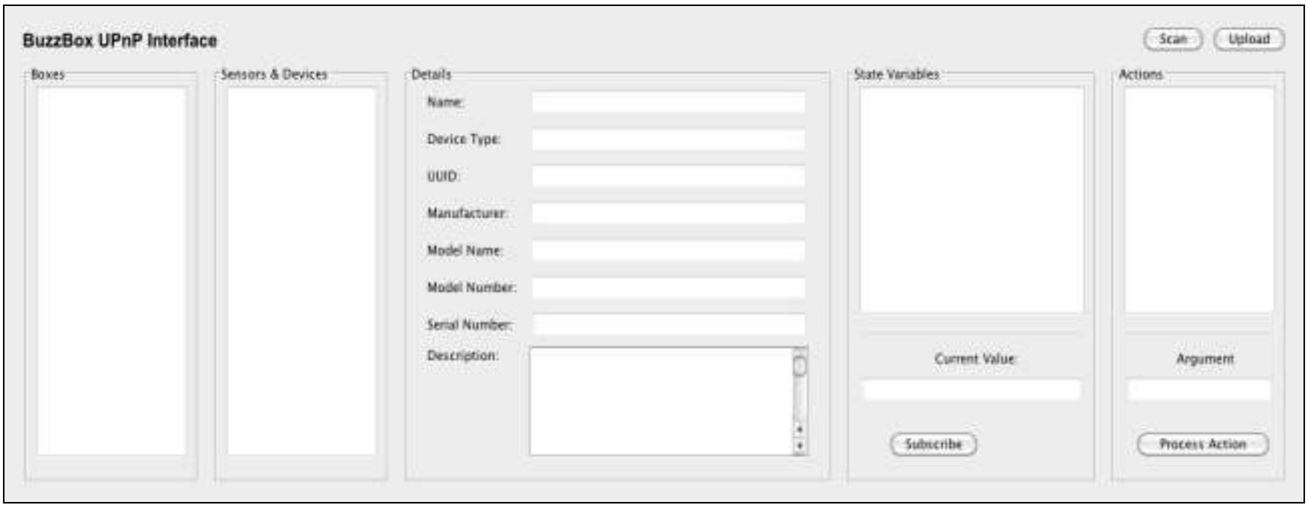

Figure 2. An early evaluation interface for the MMO Intelligent Environment system

present in the environment can also be independently represented using the MMO architecture. For example consider environment features such as a home security system or a media centre (e.g. a television, stereo, and games system, etc). Each of the intelligent devices and sensors encompassed by these systems can potentially operate, (and be represented), completely independently within the MMO architecture. However, it is also possible for these sensors and/or devices to augment each other, working together to perform more complex shared tasks, (i.e. monitoring an entire environment rather than just a small section, and creating a more immersive entertainment experience for users). Such higher-level systems, operating using multiple sensors or devices within an environment, can each be collectively represented in the MMO architecture as a 'Case'. These would exist within the architecture in a self-contained 'App'-like format, allowing them to potentially be easily transferrable between different Intelligent Environments connected to the MMO network.

Figure 3. Buzz Box – A Portable Intellgient Environment

IV. INITIAL DESIGN & EVALUATION STRATEGY

Prior to using the MMO system to represent full-sized test-beds the architecture is first being evaluated using multiple portable Intelligent Environments. The Fortito Buzz Box [4] contains many of the sensors and devices that would typically be found in a full-sized Intelligent Environment, such as a smart home, including multiple lights and a HVAC system. The sensors and devices installed on each Buzz Box can be accessed and controlled via a Raspberry Pi Model B board attached to the side of the unit. Control code is written for the box using Python, with an optional TCP Socket Server to accept commands from more complex programs, possibly written using other languages such as Java or C#. This setup allowed some initial proof-of-concept evaluations for the MMO Intelligent Environment system to take place. An interface (Figure 2), designed to represent the multiple Buzz Boxes using a single window was created using a combination of Java and UPnP. To connect to the networked Buzz Boxes, the interface took advantage of the aforementioned TCP Socket feature to access the installed devices and sensors.

Currently the new MMO Intelligent Environment system has been evaluated using four Buzz Boxes, (such as the example shown in Figure 3). Later evaluations expanded the system further to integrate multiple full-scale Intelligent Environments, including the iSpace and iClassroom test-beds located at the University of Essex. Each of the Buzz Boxes was placed in different locations around the University of Essex campus, yet networked together via a LAN or WAN. As the system is developed further this evaluation will be repeated using an increased number of Buzz Boxes, up to an eventual total of twelve.

The MMO Intelligent Environment architecture is intended to be robust, capable of representing various devices and facilities regardless of their location either geographically or with regard to their network connection. To that end the proof-of-concept evaluations have incorporated several different styles of network connection into the evaluations performed. As was mentioned earlier, the Buzz Boxes used in this evaluation were connected to a central MMO server via LAN or WAN connections. The iSpace and iClassroom Intelligent Environments, (each of

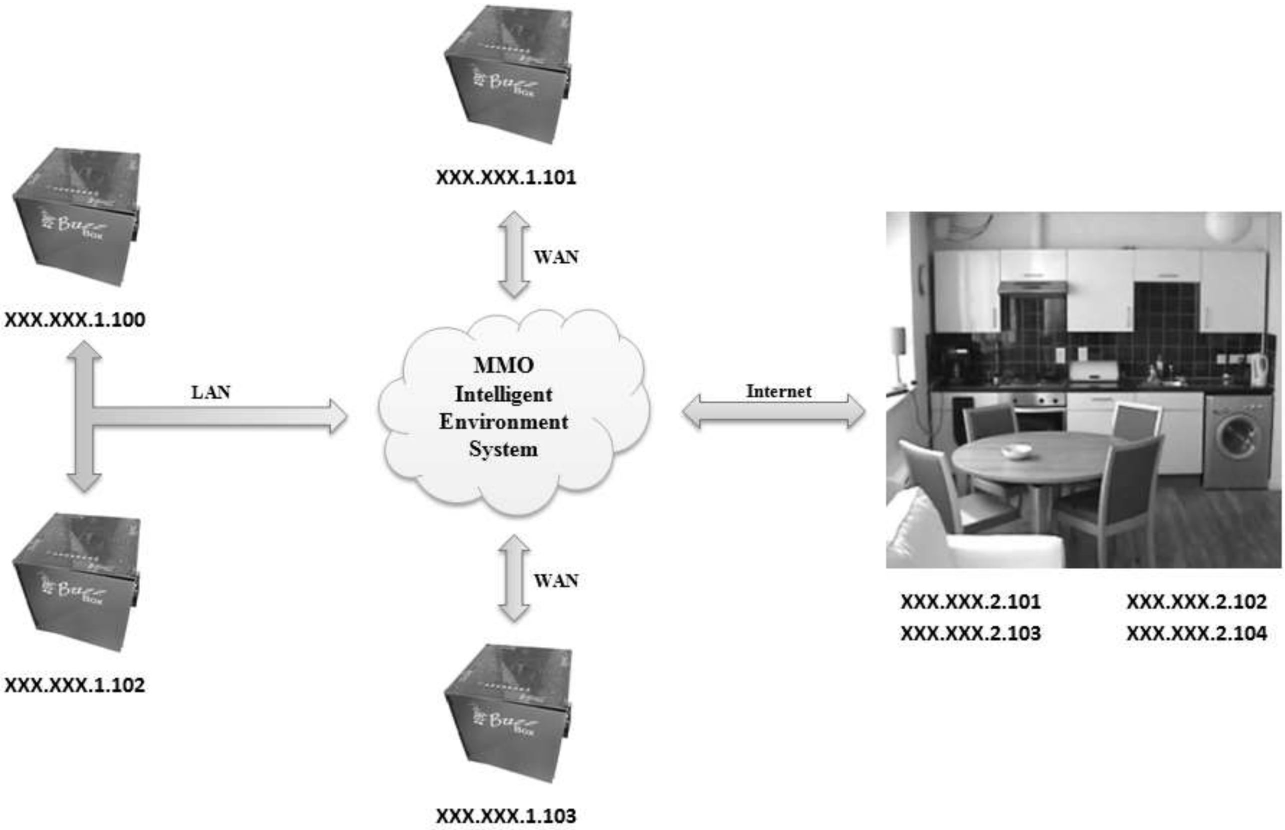

Figure 4. An example of several different distributed Intelligent Environments connected to the MMO system

which host their own independent networks), were attached to the MMO system via a standard Internet connection using a combination of UPnP and TCP/IP to act as a bridging mechanism. The intention in the current architecture was to have each Intelligent Environment attached to the MMO system behave similarly to a client or player in a typical online gaming platform. Figure 4 shows an overview of one of the MMO system designs used in these early proof-of-concept evaluations. This particular set-up involved the use of four Buzz Boxes, (two on a LAN, two in different locations connected via a WAN) and the University of Essex iSpace.

A. *Evaluation Aims & Initial Findings*

The intention at this ongoing phase of the investigation was to perform proof-of-concept evaluations for the proposed MMO Intelligent Environment architecture. This included identify issues surrounding the direct integration of multiple unrelated Intelligent Environments into a single large-scale network. For example, each of the Buzz Boxes being used during this phase was identical in terms of both hardware and software design, from the range of sensors and devices installed to the control code being used to access and manipulate them. Therefore, when representing the sensors and devices contained within each Buzz Box on a network, there would appear to be multiple declared instances of seemingly identical objects, (e.g. using the same names, id numbers, etc). This could present a problem for developers of agent programs who want to be able to access specific devices on one or more specific Buzz Boxes from external program code.

If using a control system such as UPnP, searching for specific devices can become much more complicated if there are multiple instances with the same name, service ID, UUID and serial number being represented on the network. Typically, unless somehow instructed otherwise, a UPnP system would always return the first matching device it detects matching the some pre-defined search parameters, regardless of how many matching objects were actually found during a single iteration of the declared list. Therefore, in cases such as the MMO system, where multiple devices with identical details may exist, there would be no guarantee any requested state modifications would be performed on the desired object.

B. *Early Design Modifications*

To address the particular problem of duplicate device declarations, the MMO architecture used the IP addresses individually assigned to each Buzz Box as a means of identifying specific devices. Initially, the entire distributed network of boxes was presented to users via the interface design shown in Figure 2. Only when a specific Buzz Box was selected from the relevant interface menu was it possible to view and modify the intelligent devices installed within that particular environment. When searching using UPnP code it was likewise first necessary to determine the IP address of the hosting box for each possible match until the actual device being sought was found. While perfectly functional this approach requires users to be aware of the IP addresses of any box they are attempting to access, (which may change if not static). This could prove especially challenging in larger-scale

Intelligent Environments which, if based around a distributed architecture, would likely use several different computers to host and run their systems. For example the iSpace Intelligent Environment shown in Figure 4 actually has four different networked computers exposed to the MMO system, compared to the single Raspberry Pi used by each of the connected Buzz Boxes. Additionally, extra processing and other resource overheads are placed upon the MMO system while looping through connected computers and their detected devices until finding the correct one. Following this investigation the system will be extended to attempt to address these issues, plus any others identified as a result of the ongoing evaluation.

V. Conclusions

In this paper we have presented an overview of our initial research into creating a new MMO system for Intelligent Environments, largely inspired by pre-existing computer games and their related technologies. Following some background research we discuss the various aspects of MMO computer games that are of interest to us and how those features could potentially be of benefit to Intelligent Environments research.

We then highlight our initial proof-of-concept evaluations and provide details of some early modifications already made to the MMO Intelligent Environment architecture, attempting to address the identified issue of duplicate descriptions of certain devices being present on the system network. We provide details of our current solution, namely using a bespoke interface and API to filter intelligent devices, declared on the MMO network using UPnP, by the IP addresses of their host computer systems. Some of the advantages and limitations of this solution are discussed, as are our plans for extending the current evaluation format in the future to incorporate different strategies and an increased number of networked Intelligent Environments. While using IP addresses of their respective host computers as a means of identifying specific intelligent devices did function perfectly well as a solution to the duplication issue, the solution did require pre-existing knowledge of the network of an environment on the part of individual users or programmers of agent / control code. As not all users may be aware of exactly which computers exist in an Intelligent Environment, or which devices they operate, (indeed many such facilities use embedded technologies with the specific intention of keeping their existence hidden from users), it was felt that the IP based solution may not always be ideal in certain situations. Therefore, as this proof-of-concept evaluation and subsequent research continues, alternative methods will be sought for segregating duplicate devices on an MMO Intelligent Environment network.

Finally, following these conclusions we build further on this discussion by outlining the intended next steps for this research as we continue into the future. Several potential expansion strategies for the research focus are available, potentially including investigations evaluating the use of multi-environment agents and distributed sensor networks as tools in MMO Intelligent Environments.

VI. Future Work

Some details of how this work will be extended in the future have already been provided in earlier sections of this paper. However, to summarise, the next steps of this research are to continue the current evaluation of the MMO Intelligent Environment architecture using increasing numbers of Buzz Boxes. Following a modification phase where any issues arising from this evaluation will be addressed, (such as the duplicate device issue discussed earlier), several full-scale Intelligent Environments will be integrated with the MMO system. Initially this integration will consist of the iSpace and iClassroom Intelligent Environments at the University of Essex, with further facilities in other locations around the world subsequently being added using an incremental evaluation, following a similar strategy to the earlier Buzz Box setup.

Once the MMO Intelligent Environment system is established, this research can be extended in other directions as new possibilities arise from the augmented functionality. Firstly, with multiple Intelligent Environments being networked together, agent programs designed to operate across several (potentially very different) test-beds can be investigated and evaluated. For instance, this may involve several environments customising themselves to suit individuals who travel between them, (e.g. an office and home environments). Alternatively, MMO agents could be designed to make generalisations on user preferences according to generic factors such as their age-group or nationality, the time of day or local climate conditions. On a slightly different tangent, it may also be possible for users of one MMO Intelligent Environment to download, (or access in real-time) sensor and/or device usage data gathered from equipment in another smart space on the system. Such functionality could be useful if for example a researcher in one part of the world was designing a system to be used in an environment located in another. The researcher could use the data from the target environment to customise their own intelligent space to resemble the eventual deployment platform. In another example, researchers of Intelligent Environments would be able to evaluate their agents and control systems using actual environmental conditions from several different world locations, (potentially encompassing several different seasons and climates), rather than being limited to testing using the data available from sensors and devices installed in their local area at that particular time.

References

[1] Cook D. J., Youngblood M., Heierman E., Gopalratnam K., Rao S., Litvin A., and Khawaja F., "MavHome: an agent-based smart home", Proceedings of the IEEE International Conference on Pervasive Computing & Communications, pages 521 – 524, 2003.

[2] Davies M., Callaghan V., "iWorlds: Building Mixed Reality Environments using Customisable 3D Virtual Worlds", Proceedings of the 6th IEEE International Conference on Intelligent Environments (IE '10), Kula Lumpur, Malaysia, 2010.

[3] Davies M., Callaghan V., "iWorlds: Generating Artificial Control Systems for Simulated Humans using Virtual Worlds and Intelligent Environments", Journal of Ambient Intelligence and Smart Environments (JAISE), Vol. 4, No. 1, Eds. Augusto J. C., Aghajan H., IOS Press, 2012.

[4] Fortito, “Buzz Box”, http://fortito.mx/en/?buzzboard=buzzbox, Retrieved: March 2014.

[5] Jia L., Afshari S., Mishra S., Radke R. J., “Simulation for pre-visualizing and tuning lighting controller behavior”, Energy and Buildings, Vol. 70, pages 287-302, 2014.

[6] Liu Y., Todorka A., Tatsuo N.. "Gamifying intelligent environments." Proceedings of the 2011 International ACM Workshop on Ubiquitous Meta User Interfaces (Ubi-MUI ’11), Scottsdale, USA, ACM, 2011.

[7] Milgram P., Kishino A.F., “Taxonomy of Mixed Reality Visual Displays,” IEICE Transactions on Information and Systems, Vol. E77-D, No. 12, pages 1321-1329, Dec 1994.

[8] EA Games, “Origin”, https://www.origin.com, Retrieved: March 2014.

[9] Unity Technologies, “Unity – Game Engine”, http://unity3d.com, Retrieved: March 2014.

[10] Valve Corperation, “Steam”, http://store.steampowered.com, Retrieved: March 2014.