

Bridging the Gap Between Networked Appliances and Virtual Worlds

P. Fergus, D. Llewellyn-Jones, M. Merabti, A. El Rhalibi

Networked Appliances Laboratory
School of Computing and Mathematical Sciences
Liverpool John Moores University
Byrom Street, Liverpool L3 3AF, UK

Email: {P.Fergus, D.Llewellyn-Jones, M.Merabti, A.ElRhalibi}@ljmu.ac.uk

Abstract: On-line gaming coupled with virtual world development and social networking are at the forefront of next generation entertainment. In parallel, there is an unprecedented push to equip every device with networking capabilities. However, there is little evidence to suggest that these distinct technological trends will converge rather than just co-exist. There are obvious benefits to support interoperation between real objects and virtual worlds in at least two ways: improving the possibilities for device interactions and improving the functionality that is available within virtual worlds. Moreover, it also allows the physical constraints associated with real objects to be removed, enabling them to enjoy the freedom offered by virtual worlds. To promote this vision, we propose an integration architecture that blurs the gap between real and virtual objects. This supports both conventional and unconventional usage of devices visually connected to digital avatars, thus freeing users to decide how they wish to operate devices. Although much work has been done in both networked appliances and interactive entertainment, there is a need to converge these disciplines to harness the power afforded by new and emerging digital worlds. This will make it easier to interact with devices through the obvious benefits physical disembodiment brings. We present a new approach demonstrated using a working prototype to implement a virtual world capable of operating a remote appliance, either via the device or via its digital avatar.

Keywords: Networked Appliances, Networking, Peer-to-Peer Virtual Worlds, Service-Oriented Architectures, MMOG, Disembodiment.

I. INTRODUCTION

In recent years devices have converged to provide new functionality and extend the operational capabilities they support beyond those they were initially designed with. This has led to new and novel applications designed to challenge conventional usage, capitalising on devices that enjoy the benefits network interfaces provide. Whilst the mobile phone communication infrastructure is seen as the predominant ubiquitous network, technological advances in wireless standards, particularly those built on IP are providing an interoperability platform capable of allowing every device with a digital heartbeat and network interface to form part of these networks. As this trend continues we can expect devices to become physical extensions to existing communication channels. This has the potential to have a significant impact on how and when we interact with the devices we own. Whilst at the simplest level we will continue to use devices much as we

use them today, or as appliances that can be accessed remotely, it also affords the potential to free devices and their users from the constraints that physical hardware currently imposes. Unconventional use of device operations will then blur the distinction between the real and virtual worlds.

Enabling technologies, such as *Second Life* and *World of Warcraft*, go some way to realising the vision of removing physical constraints. However there is currently no concept of real object integration and the only physical entities 'virtualised' in these environments are the players themselves. Their appeal stems largely from the application of global networking as a means of connecting distant users. This has resulted in new types of game play built on social networks, interactive Internet services and ad hoc networking. We propose to go a step further by allowing the dynamic integration of networked appliances into the virtual environment. This not only makes for more convincing virtual worlds through enhanced perception, but also provides a platform enabling bidirectional communications between real and virtual objects. For example, figure 1 illustrates a mobile phone perceptually connected to a digital avatar within a virtual world.

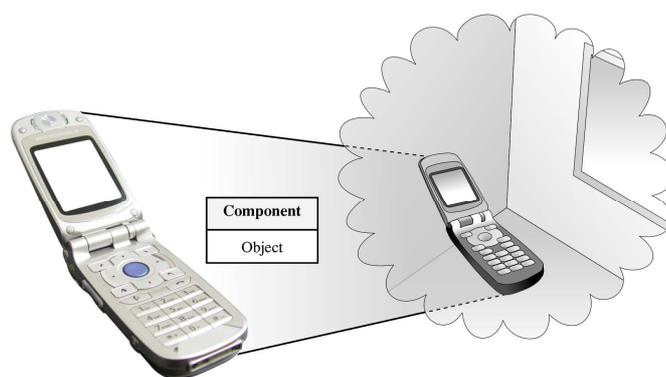


Fig. 1 Physical networked appliances can be integrated into the virtual world

Cursory consideration suggests this might be nothing more than a reimplementing of device interfaces within the virtual environment. However, a deeper analysis reveals that by blurring the gap between real and virtual worlds new and novel ways of using devices can emerge. Currently, if you leave your

II. BACKGROUND AND RELATED WORK

mobile phone at home you are prevented from using it. The exception to this is remote access to voicemail using a PIN, indicating a practical need for distributing appliance functionality albeit in a very limited way. Through new extensible frameworks it will become possible to interact with the full complement of mobile phone features via its associated avatar within a virtual world, just as if it were the same phone as the physical artefact. Interaction takes place between users and the real mobile phone via its associated avatar, connected to the real object through bi-directional communications. When answering the avatar phone, communication is re-diverted from the mobile phone to the avatar in the virtual world and *vice versa*. Taking this one step further we can build on the desktop metaphor, which has revolutionised modern day computing, to include day-to-day metaphors associated with real-world interactions providing a means of bridging real and virtual objects. For example, in a virtual world a user could browse the pictures in a photograph album ‘avatar’ connected to all real-world devices containing digital photos. The user does not need to know where each photo resides: the content is pulled into the photo avatar in conformance with the metaphor description. In this latter case, the facilities of multiple devices (phone, camera, computer storage) are combined to provide a single digital entity.

There is a need to enable this integration not only to overcome the physical constraints inherent in physical devices, but also because virtual worlds are becoming just as important as their real-world counterparts [1]. As such platforms need to be developed that embrace this change allowing use of physical and virtual objects in whatever environment they occupy. This is a difficult challenge because relationships between real and virtual objects do not currently exist. Nevertheless, there is a need to address this challenge to avoid object exclusion and ‘world specific’ object interactions.

We therefore propose a framework whereby networked appliances can automatically create and interact with associated avatars within a virtual world. The framework adopts a novel approach and provides a number of benefits. First, real and virtual worlds do not simply co-exist, but rather converge to produce an abstraction layer that eases how, when and where we interact with real and virtual objects. Second, it removes the physical constraints associated with real-world objects, allowing the use of devices in unconventional ways. For example, a physical mobile phone is not generally equipped with the ability to levitate. In the virtual world, such functionality could be added easily. Alternatively, the user could not only change the fascia of the phone, but an entire graphical user interface could be easily transplanted with a voice interface replacement.

The following will be discussed to demonstrate the applicability of the proposed framework. Current research initiatives within networked appliances and gaming technologies; our proposed framework; a case study; the prototype used to evaluate the proposed framework; and our conclusions and future work.

A number of networking solutions have been developed to simplify how devices are interconnected and operated. The trend suggests that this is set to become an increasingly prevalent requirement as networked appliances become more pervasive. To accommodate this, the Reconfigurable Ubiquitous Networked Embedded System (RUNES) [2] project is trying to provide a platform that allows a larger number of devices to form part of the Internet. To support such environments it aims to manage this integration, thus providing users, designers and programmers with the flexibility to interact with services, devices and sensors. It promises to deliver a platform to enable the formation of large-scale, distributed, heterogeneous network systems that can interoperate and dynamically self-adapt to environmental changes.

The Digital Living Network Alliance (DLNA) [3] project supports the RUNES approach; however it suggests that interoperability must be between devices that reside within three domains: the Internet, broadcast and mobile. DLNA advocates that the key to successful integration is to address customer demands where the devices they own work together within and across these domains.

Whilst RUNES sees every device as an Internet node, DLNA sees every device forming part of an interoperable platform connecting multiple communication channels. Clearly, these approaches provide mechanisms for flexible device management, liberating users from increased cognitive demands. However, they only provide a base solution, allowing for device integration using global communications. Although this is seen as important it is a low-level requirement and there is still a need for the development of higher-level abstractions to ease how networked-appliances are used.

This requirement, in part, is the focus of the Open Services Gateway Initiative (OSGi) [4, 5]. This specification assumes that network-enabled devices and networks already exist. Using set-top box technologies, networked appliances are interconnected and controlled. Its mission is to create open specifications that enable the delivery of multiple services over Wide Area Networks (WANs) to home networks. OSGi adopts a centralised approach and as such it suffers from the same problems, *i.e.* central points of failure and scalability. This approach dictates how devices and services are implemented and controlled with closed-world solutions making it very difficult to dynamically integrate, use and remove devices as and when they appear. There is no mechanism that integrates the core infrastructure of OSGi within virtual worlds – the device-avatar concept does not exist. Service providers and managers control software representations of devices and services, not the user.

In contrast to centralised approaches, ad hoc networking standards for networked appliances are beginning to realise the benefits peer-to-peer technologies might bring. In these networks devices are seen as the centre of the information space, where functionality is received from within the devices

immediate vicinity. The AMIDEN [6] project builds on this concept by allowing devices to disperse the functions they provide as independent services. A standard with considerable support that adopts this notion is Universal Plug and Play (UPnP) [7]. Devices are automatically discovered, interconnected and controlled within local area networks, thus extending the auto-configuration features of device Plug and Play (PnP). AMIDEN and UPnP provide mechanisms to interconnect devices, however the use of avatars and virtual environments are not considered. These approaches address low-level requirements and do not concern themselves with usability or management functions.

In this paper we suggest that advances within the gaming industry, and virtual environments such as *Second Life* could help to better utilise the devices we own and provide a more simplified interface. It is not sufficient to simply interconnect devices, but to also define how users will interact with them. Utilising game functionality may provide a platform on which this could be achieved.

Massively multiplayer online games already attract huge numbers of players and are expected to become increasingly popular, forming the basis of next-generation entertainment. Utilising Internet communications, games are beginning to converge with aspects of Internet computing [8]. This has changed how users view and play games. Many games such as *Planetside* [9], *Star Wars Galaxies* [10], *The Sims Online* [11] and *EVE Online* [12], are dependent on network communications. None more so than the game *World of Warcraft*, which became the fastest selling PC game in North America in 2004-2005 and in 2006 was reported to have 6 million subscribers worldwide [13].

An interesting strategy within the gaming industry that we are able to build upon is the decoupling of game specific components from the game engines. This has resulted in new plug and play platforms on which any game functionality can be built. This resembles strategies used within other entertainment sectors such as audio devices and home networking. More specifically we find analogies within research in the Networked Appliances community [14]. We can therefore expose a useful similarity between such devices and game components. Through the application of techniques from networked appliance development, scaled in order to disperse the operational capacity games support within and across different networks. This will allow components to be discovered and composed with other components used in other games. Building on this idea we can develop avatars within these worlds that are perceptually connected to real-world objects. Using bidirectional communications avatars and networked appliances can communicate between each other irrespective of where the user of the device resides. Combining advances in gaming and networked appliance development allows us to remove the constraints associated with physical devices. This allows devices to exhibit unconventional behaviours within the virtual world, and at the same time real-world functions provided by devices can be used within the

virtual world. This creates an invisible bridge where it becomes increasingly difficult to see where boundaries lie.

III. COMPONET INTEGRATION PLATFORM

We will first consider the relevant properties of networked appliances. From a hardware perspective, Moyer *et al.* [15] define networked appliances as “*a dedicated function consumer device with an embedded processor and a network connection.*” We can further characterise them as being devices that disperse their operational capacity as independent services that can be discovered, and used by other networked appliances in the network (LAN or Internet) to control, monitor, manage and extend the functionality they support beyond what they were initially designed to do.

Our approach, based on service-oriented computing, builds on this definition. It allows components (this could be a device, which may include game consoles, or software components) to implement a small footprint of code enabling framework and application specific functions to be dispersed within the network. Using framework services, components can connect to the network using any underlying communication protocol; discover and/or publish and use framework and application specific services locally (provided by the component itself) or remotely (provided by other components); perform semantic interoperability between different vocabularies used by component manufacturers; automatically form communication links with other components in the network; and self-manage their operations based on composite and environmental changes. Application specific services, on the other hand, provide a means of dispersing and utilising component functionality (such as audio and video, gaming engines, and player (AI behaviours) and non-player objects (tree, car or avatar).

Components connected to the network rely on three key characteristics. First, they provide access to framework and component functionality using services. Second, components automatically form communication links with other components in the network based on the functions they provide and the functions they require. Third, components manage communication links and self-adapt to unforeseen changes that occur within the environment, *e.g.* increased network traffic. The proposed framework allows high-level applications to be automatically created through component interactions. This reduces the cognitive demands placed on users, who are free to interact with devices without having to define how applications are created beforehand.

It has been a goal of our work to deconstruct as far as possible the holistic notion of devices into a set of autonomous, generalised and reusable components (services). Whilst the development process of our framework necessarily entailed the compartmentalisation of various aspects of traditional virtual environments and device usage, the final result must therefore be considered from the opposite perspective. Ultimately we aim to allow applications (gaming and conventional networked appliance integration) to exist as an ad hoc interaction between various services, the entirety of

which blurs the real-world and virtual environment. We decompose not only appliance functionality but also the elements of the virtual environment itself in terms of the physics engine, rendering engine and virtual objects that persist within the world. These are each provided as individual, replaceable, services in the network as shown in Figure 2. Therefore, whilst we can conceptually distinguish between components that are interpreters, those that are virtual objects and those that are networked appliances with an additional physical manifestation, the framework itself makes no distinction and provides a uniform platform across them all. A combination of components is needed to generate a convincing virtual world, and none of these components in isolation can therefore be considered to be the application itself.

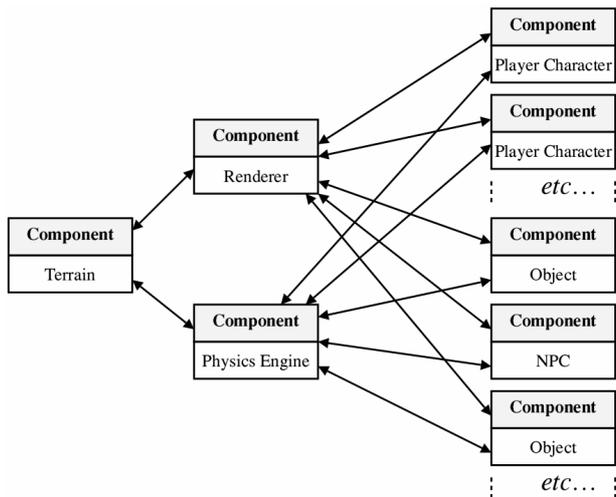


Fig. 2 Proposed Component Integration

Components automatically connect to the network without having to register themselves or the services they provide with any third party. When a component is initialised it has the ability to automatically integrate itself within the network and publish the functions it provides by multicasting XML service advertisements. Components may remove themselves and their functions from the network at any time.

The framework achieves this using a distributed unstructured services manager [16], which is the primary capability every component – be it a networked appliance of component solely from the virtual world – must implement. This is a peer-to-peer interface that can be mapped onto any middleware standard. Devices connect to the network as either specialised components or simple components as illustrated in Figure 3.

Here a specialised component has the ability to provide services as well as propagate service requests within the network. A simple component by comparison does not have these abilities. This type of component joins the network, propagates queries and invokes discovered services. This enables any game component irrespective of its capabilities to effectively choose how it will interact within the network. Figure 4 illustrates two extremes: components that are highly capable and those that have limited capabilities. However, it is

envisaged that there will be a myriad of other possibilities between these extremes.

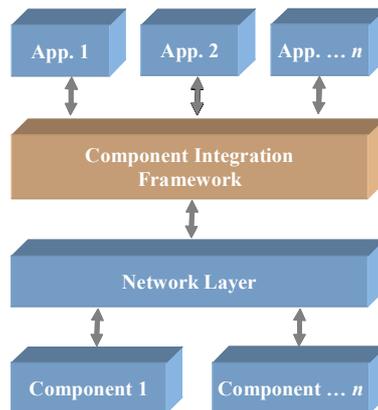


Fig. 3 Proposed Framework

The distributed unstructured services manager marshals all communication between a distributed application (virtual world – physical device integration) and its components within the network. It also provides adaptation functions used to monitor communication links between components in the network. By managing references to external components, emergent applications can adapt to changes that occur. This collective interaction between components results in the emergence of applications, including virtual world and physical device bridges. During execution each component monitors its own interaction within the composition and adapts to any changes that may occur. If a component becomes unavailable, an alternative component can be selected that provides the same or similar functionality.

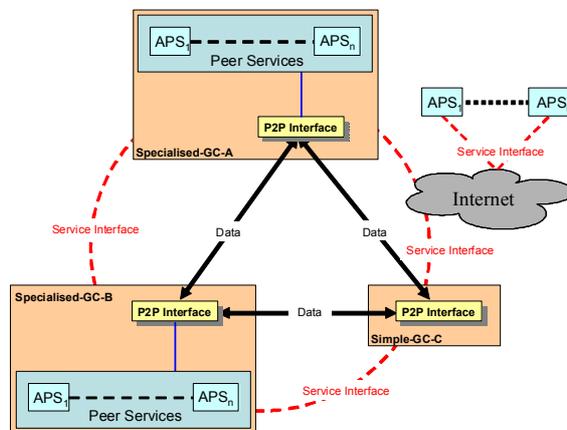


Fig. 4 Distributed Unstructured Services Manager

Using this architecture, we have designed a distributed service-oriented platform for use with virtual world and physical device bridging allowing operation from either environment. We have successfully been able to test our design and demonstrate how components can be dispersed and better utilised.

IV. IMPLEMENTATION

Using the architecture described above, we have designed a distributed service-oriented platform to bridge between networked appliances and associated avatars in virtual worlds. In this way we have been able to test our design and demonstrate how physical devices and avatars can be bridged, allowing interaction in either the real or virtual world.

We follow a multidisciplinary approach whereby several tools and standards have been combined to implement our framework. This is not simply an integration exercise, but instead provides a new approach to physical device and virtual world integration. The virtual environment components of our framework were developed using the Java Monkey Engine (JME) [17]. The rendering interpreter was implemented using the Java Monkey Engine, with the JME Physics Engine used as a basis for our physics interpretive engine. Based on the framework described above, we extended a standard JME terrain environment to incorporate peer-to-peer capabilities. The discovery of components within the framework was implemented with dynamic integration into the virtual world. All the services used to expose components were developed using the JXTA protocols [18]. These protocols allow such components to connect to the network. We note however that in essence the virtual environment implementation is transitory by design. Just as the system is intended to work independent of the networked appliance hardware, so we could replace the rendering engine with any other engine. In fact, this is an important feature of the design. It would be perfectly acceptable to create a virtual world using cell-shaded graphics, or even in the form of a text-based adventure game, as long as the rendering engine is able to interpret the data received from the component into a suitable human-perceivable form.

Components (devices and software components) propagate messages in the peer groups using the JXTA *ResolverService* protocol. We currently have two devices in the real world, which for simplicity are interactive 3D boxes. Each device provides a *BoxService* that is used by the virtual to interact with the box. Using *ResolverService* listeners, box devices define the type of messages they are able to process. The virtual world (*TestGenerateTerrain*) propagates *BoxMessage* messages, which the *BoxService* listens for. *BoxServices*, upon receiving these messages, return service advertisements back to the querying node. JXTA service advertisements are used to describe high-level information such as the provider and the purpose of the service.

Once *TestGenerateTerrain*, receives the advertisement, the pipe advertisement is extracted and used to bind to the service before the box information is requested. All successful connections result in the communication of information between the two services. This allows *TestGenerateTerrain* to project the appropriate facets of the *BoxService* into the virtual environment, in this case resulting in a box avatar being dropped into the virtual world. The characteristics of these facets are controlled via a loose connection to the service provider through the *JxtaBiDiPipe* it uses for bidirectional communications.

We demonstrate this using three computers within a peer-to-peer network (we used the JXTA specifications [18]). This simulates how three-dimensional objects can be discovered and attached to the *rootNode* of a virtual world developed using the Java Monkey Engine [17]. The first two computers host *BoxService* services that support a simple user interface as illustrated in Figure 5. This service could represent any physical device such as a mobile phone or MP3 player.

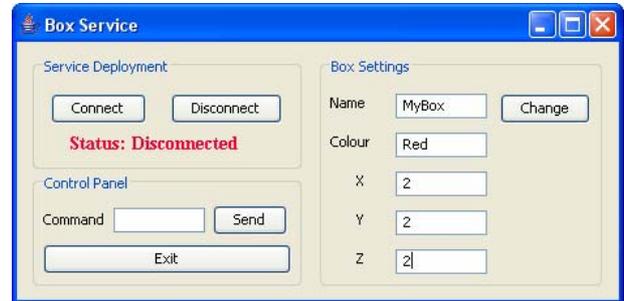


Fig. 5 Box Service Interface

Using this interface the *BoxService* can be deployed within the network allowing it to be discovered and used by another application, which in this case is the virtual world. The box settings panel allows the properties of the box to be changed. Such changes persist within the *BoxService* itself and are broadcast to peers within the network to update any instance of its use in the virtual world. The figure illustrates that the name, colour and dimensional coordinates can be dynamically changed. A subtle but important point is that the properties and behaviours associated with the box are not changed by the virtual world but rather by the owner of the box. Although the properties may be affected by feedback from a component such as a physics engine, it is the responsibility of the box component to perform any reaction and the box itself is controlled entirely by the owner.

In our prototype the third computer hosts a physics engine (JME Physics Engine) and rendering engine. When executed its appearance is dependent on what that virtual world provides and what components are available within the network. For simplicity, the prototype is a virtual world that tries to discover *Box Services* (these could be physical devices where boxes are their associated avatars) provided by the two computers connected to the network. For each box service discovered a box object is rendered in the world in conformance with the properties and behaviours the box service supports. Figure 6 provides an illustrative representation of the virtual world used in the prototype containing the two box services.

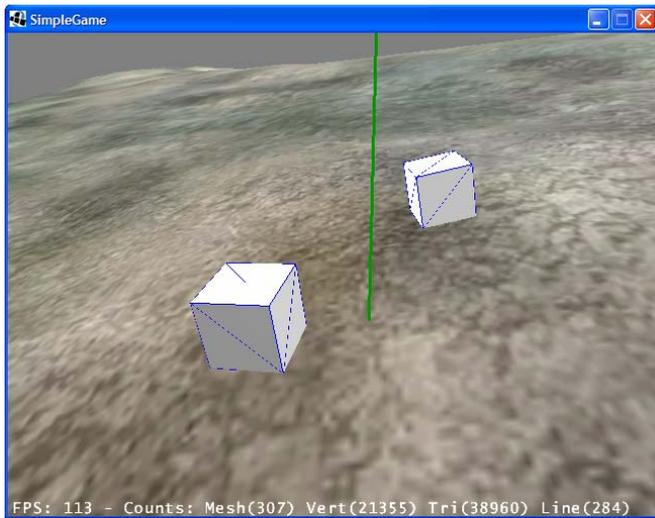


Fig. 6 Virtual world interpreting two box services

The integration between physical real world boxes and avatars that reside within the virtual world, require three steps as defined in the framework discussed above. First, the boxes and the virtual environment publish the service components they wish to share. Second, the virtual environment discovers all box services and renders them within the virtual world. Third, it renders any changes to the objects as and when component owners submit changes or execute behaviours.

V. CONCLUSIONS AND FUTURE WORK

Networked appliance technology is developing rapidly, but is currently based on a premise of using static interfaces in order to access device functionality. Simultaneously, virtual environments are becoming increasingly sophisticated, but lack any means for users to augment them with extended functionality, let alone allowing access to functionality provided by physical artefacts from the real world. Yet taking technology from both, a unique – symbiotic – relationship can be established to allow both massively increased flexibility in the way we are able to use and interact with networked appliances as well as the potential for far greater capabilities to be accessible from within virtual worlds. The extent of this is such that both the physical and virtual can be endowed with functionality that would be impossible to achieve otherwise. Our prototype clearly indicates the technological feasibility of such a system, however to become effective, cross-community standards are required to allow the two technologies to intercommunicate universally. We have detailed our initial progress in defining such technologies. In future work, we aim to extend the system further to investigate the use of a greater variety of peer services and especially the potential for using

multiple similar components simultaneously (such as rendering engines or devices) within the same virtual world.

REFERENCES

- [1] J. Cole, Online World As Important to Internet Users as Real World, University of Southern California: California, Whitepaper, 2006.
- [2] C. Koumpis, L. Hanna, M. Anderson, and M. Johansson, Wireless Industrial Control and Monitoring beyond Cable Replacement. 2nd Profibus International Conference, 2005, Warwickshire, UK, p.
- [3] DLNA: Overview and Vision. 2004, DLNA, Accessed: October 2006, http://www.dlna.org/about/DLNA_Overview.pdf.
- [4] L. Choonhwa, D. Nordstedt, and S. Helal, Enabling smart spaces with OSGI. IEEE Pervasive Computing, 2003. 2(3): p. 89 - 94.
- [5] The OSGi Service Platform - Dynamic services for networked devices. 2005, OSGi Alliance, Accessed: 2006, <http://www.osgi.org/>.
- [6] M. Minoh and T. Kamae, Networked Appliances and their Peer-to-Peer Architecture AMIDEN. IEEE Communications Magazine, 2001. 39(10): p. 80-84.
- [7] UPnP Forum. 2005, Microsoft Corp., Accessed: 2006, <http://www.upnp.org/>.
- [8] A. F. Seay, W. J. Jerome, K. S. Lee, and R. E. Kraut, Project massive: a study of online gaming communities. CHI '04 extended abstracts on Human factors in computing systems, 2004, Vienna, Austria: ACM Press, p. 1421-1424.
- [9] PlanetSide. 2006, Accessed, <http://planetstide.station.sony.com/>.
- [10] Star Wars Galaxies. 2006, Accessed, <http://starwarsgalaxies.station.sony.com/>.
- [11] The Sims Online. 2006, Accessed, <http://www.ea.com/official/thesims/thesimsonline/>.
- [12] EVE Online. 2006, Accessed, <http://www.eve-online.com/>.
- [13] N. Ducheneaut, N. Yee, E. Nickell, and R. Moore, Building an MMO with Mass Appeal: A look at Gameplay in World of Warcraft. Games and Culture - Journal of Interactive Media, 2006. 1(4): p. 281 - 317.
- [14] A. Mingkhwan, P. Fergus, O. Abuelma'atti, M. Merabti, B. Askwith, and M. Hanneghan, Dynamic Service Composition in Home Appliance Networks. Multimedia Tools and Applications: A Special Issue on Advances in Consumer Communications and Networking, 2006. 29(3): p. 257 - 284.
- [15] S. Moyer, D. Marples, S. Tsang, and A. Ghosh, Service Portability of Networked Appliances. IEEE Communications Magazine, 2000. 40(1): p. 116-121.
- [16] P. Fergus, A. Mingkhwan, M. Merabti, and M. Hanneghan, DiSUS: Mobile Ad Hoc Network Unstructured Services. (PWC'2003) Personal Wireless Communications, 2003, Venice, Italy: Springer, p. 484-491.
- [17] Java Monkey Engine. 2006, Accessed, <http://www.jmonkeyengine.com/>.
- [18] L. Gong, JXTA: A Network Programming Environment. IEEE Internet Computing, 2001. 5(3): p. 88-95