Federated Homes: Secure Sharing of Home Services

Zohar Etzioni, Kevin Feeney, John Keeney, Declan O'Sullivan
KDEG & FAME, School of Computer Science & Statistics, Trinity College Dublin
E-mail: {etzioniz,kevin.feeney,john.keeney,declan.osullivan}@cs.tcd.ie

Abstract — *This paper presents an architecture which allows consumers to securely share the services available in their home networks with remote third parties. It is implemented as a software service which can be installed on a home-gateway device. The implementation supports dynamic sharing of UPnP-compliant devices and multimedia content and enables complex service compositions across federated homes through a secure Federal Relationship Manager (FRM) service. We have extended UPnP to use XMPP as the underlying messaging protocol to maintain up-to-date state information about the availability of devices and content to federated third parties. The architecture, implementation and performance evaluations are described, demonstrating the potential for this technology to provide consumers with a mechanism for federating home services. This mechanism is sufficiently powerful to satisfy fine-grained sharing constraints through a user interface that is sufficiently simple for average consumers to master.*

I. INTRODUCTION

Home area network adoption has grown in recent years, allowing interconnected networked consumer devices and appliances such as media centers, printers, game consoles, cooling systems, smart-phones, etc to communicate with one another. These devices require automatic network integration and support for discovery and configuration to allow non-technical home users to deploy and use them with minimal setup overhead. Several architectures have been suggested for addressing these challenges including Universal Plug and Play (UPnP) [1], ZeroConf [2], Jini [3] and others. The Digital Living Network Alliance (DLNA) [4] is an industry consortium formed to promote interoperability of Internet, mobile and broadcast services through simple integration of consumer devices with home networks by defining a standard and certifying compliant devices. Many consumer devices already implement one or more of these technologies and Home Area Networks (HAN) have become common.

While these technologies typically provide good abstractions for interaction with devices in the home, they are limited to operating within a single HAN and are not ready to support sharing of content and resources securely across multiple HANs. The proliferation of broadband and the increasing importance of social networking have created consumer demand for technologies to facilitate the sharing of resources and content across HANs. By extending these technologies beyond the boundaries of the HAN, a wide range of desirable services could be offered to consumers including resource sharing, content sharing, remote control of home area networks, formation of home network communities, and composition of services from multiple home networks.

We motivate this work with the following scenario: Bob has a home area network with a UPnP enabled media server. His friend Alice has a UPnP enabled TV and a UPnP enabled DVR. Bob and Alice want to share some of their devices so that when Bob visits Alice they can watch content from his media server on her big screen UPnP media renderer. In addition Alice wishes to allow Bob to use her UPnP DVR when he wants to record his favorite TV programs.

There are, however, a number of significant challenges in extending HAN-focused technologies beyond the HAN.

1. **Performance:** the protocols and architectures that have been deployed on consumer devices are designed to support communication within the home. When applying these on an inter-HAN basis, there should be no significant degradation in quality of experience.
2. **Security:** security concerns generally have been a very low priority in comparison to ease of installation and use for consumer entertainment devices. While this may be a good trade off in the relatively secure environment of the home, it is important not to expose the vulnerable HAN devices, services or content to security threats or unauthorized use through the home gateway and firewall.
3. **Manageability:** users have relatively fine-grained and individualized requirements about what they share and with whom and these concerns are particularly important in intimate environments such as the home. Current HAN technologies lack even rudimentary support for the fine-grained relationship management required to support federated homes. Technologies from network management, for their part, are typically far too complex for end users.
4. **Dynamism:** in contrast to typical Internet services, HAN services are highly dynamic – consumer devices are often busy, powered off or otherwise unavailable. Furthermore, such devices are frequently introduced and removed from the HAN (e.g. mobile devices or new purchases).

This paper presents the design, architecture and implementation of a system that addresses these challenges and offers users a mechanism for federating home content and services that is sufficiently performant to support media streaming; sufficiently flexible to satisfy fine-grained sharing constraints; with a user interface that is sufficiently simple for average consumers to understand and master. Section 2 describes the technologies used and related research that addresses similar challenges. Section 3 describes the basic design and architecture, while sections 4 and 5 describe the implementation and analysis of our system.

II. BACKGROUND & RELATED WORK

While the various architectures proposed for HAN interoperability, control, and sharing differ in their implementation, they share some common abstractions, such as addressing, discovery, description, control, and event handling. In recent years, along with DLNA support and certification, UPnP has become the dominant architecture for controlling consumer electronic devices. UPnP is a peer-to-

peer architecture to allow network-enabled appliances to communicate with each other within a home area network. UPnP is designed for simple, easy to use integration based on standard TCP/IP technologies such as SSDP, HTTP, XML and SOAP. UPnP defines two basic classes: devices and control points. Devices obtain an IP address automatically and can be discovered in the network by responding to search requests by control points or by announcing their presence on the network. Control points can retrieve device descriptions by sending HTTP requests to devices. Descriptions are XML documents containing information about a device and its supported services. Control points can invoke actions on a device by sending a SOAP request and can also subscribe to its state change events. UPnP was not designed to run across multiple networks: its discovery is based on using local multicast addresses; its use of HTTP assumes seamless connectivity, which is unlikely across home networks where routers assign private internal IP addresses to devices.

Several researchers have proposed mechanisms for extending UPnP across multiple networks. Lee et al. [5] suggest an architecture for content sharing among UPnP devices, based on *HomeConnectors* communicating with remote *HomeConnectors* in other home area networks via a connection manager. A local SSDP manager listens to the local network and relays local SSDP announcements to remote HANs where they are repeated. However, this architecture does not traverse NAT or firewalls and assumes that all UPnP devices have public IP addresses. Chowdhury et al. [6] present a solution for connecting multiple UPnP networks based on a protocol for establishing trust groups of home networks. Once a group of home networks has been established, users can define which devices they wish to share with the group. Remote devices are represented as embedded devices in the home gateway device. This approach requires dynamic modifications to router and firewall configurations to enable sharing, which makes it less portable and resilient. Kang et al. [7] present an architecture based on UPnP and OSGi that allows users to consume multimedia services from multimedia servers outside their home network. The home gateway acts as a proxy media server from multimedia providers reachable outside the HAN. However, the approach is specific to multimedia services and is not general to UPnP services. Kim et al. [8] suggest using a SIP-UPnP bridge in the home gateway for allowing remote access to UPnP devices. In this solution secure VPN connections are established to support sharing between HANs.

Although there have been a number of architectures proposed to overcome the technical challenges of interconnecting HANs to facilitate sharing of content and services, attempts to provide consumers with tools that allow them to manage the border between their HAN and broader networks have been mostly limited to proprietary services that support a narrow range of applications with relatively rigid interaction patterns (e.g. the Slingbox [1]TV streaming device). However, the HAN is an open distributed environment characterized by high levels of device diversity, a rapid pace of technical evolution and complex and fine-grained variations in user-requirements as to how home content and services should be exposed to third parties. Thus, in order to enable practical HAN federation, the architectures and protocols which solve the technical challenges of sharing home content and services beyond the HAN must be supplemented with management frameworks that are flexible and expressive enough to handle the present and future diversity of devices and requirements. Over the last decade, a large number of Policy Based Management (PBM) languages and architectures have been developed with the broad goal of making it easier for humans to apply coordinated management across diverse and heterogeneous information resources. PBM languages such as XACML [9] allow the specification of highly expressive rules to govern the use of information resources. However, PBM research generally focuses on providing tools to support administrators of large IT infrastructures rather than end users and the complexity of the policy authoring process is such that PBM technologies remain largely inaccessible to consumers. Nevertheless, Barrett et al. [10] have demonstrated that, given interfaces that simplify and constrain the underlying policy languages, non-expert users can manage resources more successfully through specification of policy rules. In earlier work [11] we explored abstractions and visualizations to allow users to manage their relationships in the "web 2.0" ecosystem and very similar concerns apply in the HAN domain.

III. Design & Architecture

Our approach for supporting federated multi home networks is based on extending the home gateway with a fine grained sharing management system and a secure communication channel that would enable extension of HAN technologies such as UPnP. The architecture, shown in figure 1, has the following main components: an XMPP client – responsible for establishing and maintaining the communication layer between homes, a local UPnP Network Manager, and a Virtual Remote Device – responsible for implementation of multi-home UPnP, and a Capability Sharing Manager – responsible for definition and enforcement of sharing policies. A client application enables users to manage their buddy lists and sharing policies.

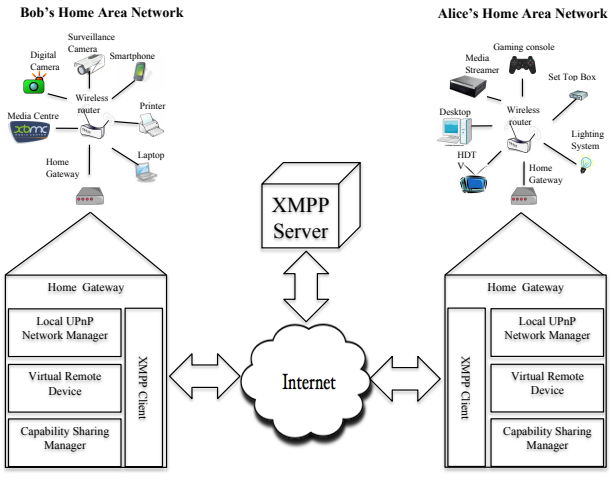

Figure 1 Multi-HAN Federation Architecture

[1] www.slingbox.com

A. Connecting Home Networks

Our system uses eXtensible Messaging and Presence Protocol (XMPP) [12] as the underlying inter-HAN messaging protocol. XMPP is an open, XML-based protocol for near real-time messaging and presence. XMPP was initially used as the messaging and presence substrate for Jabber Instant Messaging but it has since been used as communication substrate for various purposes including enabling complex communication between applications in the cloud (project Vertebra[2]), gaming (e.g. Chesspark[3]) and VOIP (e.g. GoogleTalk via the Jingle XMPP extension). Using XMPP as an infrastructure for connecting multiple home networks provides secure and standard communication, simple user roster-management and a powerful presence mechanism, which is useful for our extension of UPnP. The home gateway runs an XMPP client that connects to an XMPP server in order to communicate with the user's defined friends. The user can manage his roster by adding or removing friends dynamically. Friends are added by their user name, therefore there is no need for users to deal with complex addressing mechanisms. Once a friend becomes available, sharing processes can be initiated automatically based on user preferences. Users can manually override and define specifically what services to share for a given device with a given buddy or a group.

In our example scenario, Alice and Bob simply need to add each other as buddies via an IM client in order to initiate their sharing.

B. UPnP Extension

In order to enable secure and simple sharing of UPnP devices we have extended the UPnP protocol to work over the XMPP messaging infrastructure. The following components are responsible for the UPnP extension:

(i) Local UPnP Network Manager – responsible for the interaction with local network devices. It acts as a control point by finding all devices in the local UPnP network. When a remote friend comes online, it responds to an implicit SSDP search packet from this friend by sending a SSDP search response packet for each device/service discovered in the local network. Once Alice and Bob have established a connection by adding each other to their buddy lists, and they appear as available, the local UPnP network manager in Bob's home will send Alice's home gateway a SSDP search response for his media server. At the same time, Alice's local UPnP network manager will send Bob's home gateway an SSDP search response for her DVR. These messages will be sent over the XMPP connection between the homes. Whenever devices become available or unavailable in Bob or Alice's HANs, its presence announcement is propagated over XMPP to all friends with whom this device/service is shared. The discovery interaction between HANs is shown in figure 2.

In addition to discovery, the local UPnP network manager services remote description requests. When a device/service description request is received over the XMPP connection, it communicates with the local device over HTTP and sends back the description to the relevant friend. Before the result is sent, it is processed and may require removal of certain items that may not be shared with the remote friend. The description interaction flow is shown in figure 3. The local UPnP network manager responds to SOAP action requests by sending them over HTTP to the device and sending back the result or error over XMPP to the relevant friend. Before the SOAP request is forwarded to the device, the capabilities sharing manager is consulted to verify that the remote friend is allowed to execute the requested action. Event subscription requests are serviced by subscribing to the local device and propagating the notification over the XMPP connection to the subscribing friend. Presentation requests are similarly tunneled to the device locally and back over the XMPP connection.

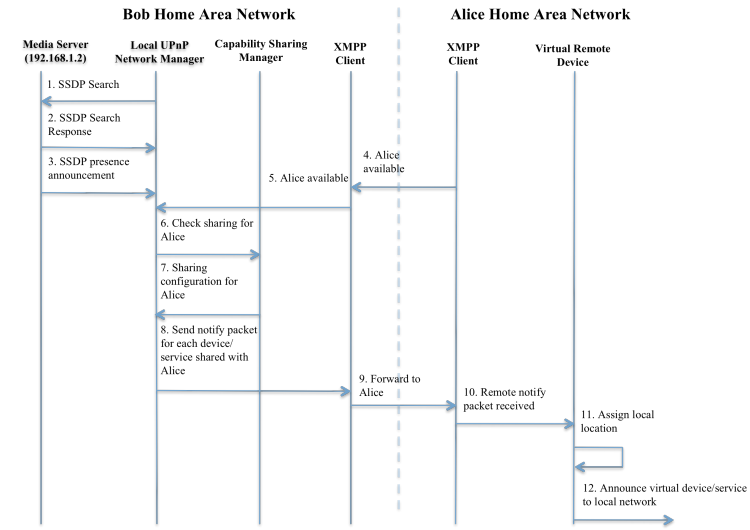

Figure 2 Multi-HAN Discovery

(ii) Virtual Remote Device (VRD) – responsible for representing remote devices in the local network - in our example, Bob's media server in Alice's network, and Alice's DVR in Bob's network. The VRD implements UPnP on behalf of all remote devices that are shared with the local user. It listens for SSDP search requests in the local network and responds with search responses for each remote device that is shared with the user. This means that if a control point in Bob's network sends a search request for DVR devices, Alice's VRD needs to respond on behalf of her DVR, and when Alice's DVR sends a presence announcement in Alice's network that will be propagated to Bob's VRD over the XMPP connection, the VRD should echo the presence announcement in Bob's local network.

The VRD learns about remote devices from SSDP announcements that are received from remote friends over the XMPP connection. When a user first goes online, and when new remote devices are added to remote networks that are shared with her, she receives appropriate announcements from her remote friends. Before the SSDP announcement can be repeated in the local network, the location of the remote device/service needs to be replaced with a location that represents the proxy of the remote device in the local VRD. This enables the VRD to service requests related to this remote device by relaying these requests over the XMPP connection to the relevant friend. All locations of remote devices are represented using the same IP and port just with a

[2] http://www.engineyard.com/
[3] http://www.chesspark.com

path that represents the user to which they belong and their unique identifier (USN). The VRD listens to HTTP requests on its port and services requests from local control points by relaying them to the relevant remote friend over XMPP. This is achieved by dereferencing the location to the actual remote device's location. The request is sent to the remote friend and is serviced there by the local UPnP network manager. This allows a control point to work with remote devices without knowing if the device is local or remote.

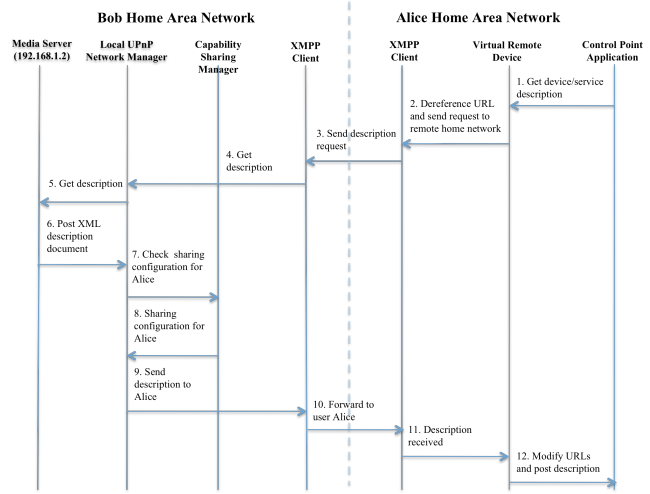

Figure 3 Multi-HAN Description

C. Capability Sharing Manager

Our capability sharing manager is an instance of our Federal Relationship Manager (FRM) [13]. The FRM was previously used to support capability sharing between large telecoms providers with heterogeneous technical platforms, and it is equally applicable to the problem of managing the sharing of home services between consumers. The FRM represents managed services and resources using a hierarchical *capability authority model*, which can be dynamically modified in order to create new aggregations of the basic capabilities made available by the underlying devices. By delegating nodes from a user's capability authority tree, representing their HAN, to a friend, that set of capabilities is made remotely available to that friend. The FRM maintains a map of which capabilities have been received and granted to other nodes and uses this map to verify all incoming requests for capability invocations.

The capability sharing manager filters which local UPnP devices and services are exposed to third parties – the granularity of this filtering can apply to devices, services, actions all the way down to individual pieces of content. Only those capabilities that are encapsulated by a capability authority that has been properly delegated to friends will be visible to those friends. Users delegate capabilities to friends by selecting nodes in the capability authority tree and delegating them to the groups or individuals in their rosters they wish to share these capabilities with. This allows relatively fine-grained sharing policies to be applied through simple tree manipulations, without requiring users to understand complex policy languages.

IV. Implementation

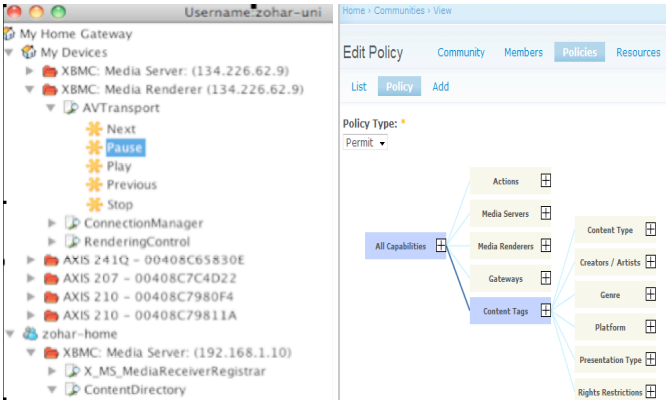

Figure 4 Client Application Screenshots Control point client (left), capability sharing manager (right)

We implemented the above architecture in Java and tested it both on OS X and Windows Vista operating systems. Figure 4 shows two screenshots of the client applications. The left pane shows the control point client which discovers local devices, some of which are actually virtual devices, representing devices in their friends' home area networks. Local network devices are shown under "My Devices". Under each device are shown shared services and under each service are shown shared actions. The client also allows the invocation of actions or subscriptions for events on remote devices. The right hand pane shows a screen shot of the capability sharing manager, implemented as a Drupal[4] module. This allows users to extend the UPnP resource models with extra semantics by adding tags which can then be incorporated into sharing policies.

V. System Analysis

To verify the operation of our system it was installed in three separate home area networks, making use of a public domain XMPP server. As sample UPnP devices, various commercial media servers and media renderers were used, including XBMC media centre and other media clients

A. Security

Using XMPP as a messaging infrastructure enables reuse of its powerful security mechanisms such as SASL and TLS and encryption. Our architecture does not require any additional ports to be open which reduces the vulnerability of home network to malicious attacks. Previous and ongoing work by the authors [11][13] has also proved the authority delegation model of the FRM and its underlying technologies.

B. Private IP Addresses (NAT)

As most home networks are behind routers, devices in such networks will have private IP addresses. Therefore they will not be accessible via HTTP to remote HANs. In order to enable communication with devices behind NAT our solution verifies that remote home gateways replace NAT-ed URLs with local ones and dereference those URLs back when communicating with remote home gateways. Communication with devices is always local, therefore our solution works well

[4] drupal.org

with networks behind routers. Moreover, it supports remote and local devices having similar private IP addresses as these addresses are prefixed with the user name and the device unique identifier when they are announced in the local network. In addition we do not require home gateways to know the IP of each other and therefore we are more resilient to changes, for example DHCP lease renewals.

C. Usability

As home area networks are operated by non-technical users, the system must be as self configuring and as easy to use as possible. Our system requires very little user intervention - limited to configuration of sharing policies and management of their roster. Once sharing policies have been defined, users and applications running in their home network can use remote devices and services transparently with no additional configuration or manual intervention. In order to give users the most simple user interface and intuitive control over their devices sharing a familiar "instant-message buddy-list" style interface was chosen to show the users their devices and the devices their friends are sharing with them.

D. Performance

The system described adds multiple layers of overhead to enable the desired flexibility:

- Communication overhead introduced by XMPP – this refers to the presence messages used by XMPP.
- UPnP protocol extension – propagation of SSDP packets to available friends introduces additional overhead on the network and processing of local network however this is proportional to the level of sharing that is desired by the user. In a setup with three home networks and 40 devices with 98 services the latency between the discovery of a device in a local network and its announcement on all remote networks has been less than 500ms on average and less than 750ms in the worst case. This requires very little (< 2 ms) overhead on the local host in notifying the device to peer networks.
- Processing overhead (CPU) – interaction with the capabilities sharing manager in order to allow or deny devices/services/actions adds a fixed overhead on the interaction with remote devices. In our evaluation on a low end Linux box (intel P4 3GHz) with 300 remote services, this does not exceed on average 4%.
- Memory overhead – in order to facilitate discovery, local network information is collected by the local UPnP network manager and remote network information is collected by the VRD. This uses memory on the home gateway, proportional to the number of local and remote shared devices. Our evaluation shows that 2-4 MB heap memory is required to represent 40 devices with 98 services shared with users.

In our early evaluations the system performed satisfactorily as expected. Communication overhead was negligible due to the limited size of SSDP notifications even under stress conditions (many control points, many shared devices). It is expected that, even with a large number of users sharing a large number of devices, this overhead will remain reasonable. The latency between discovery and announcement in remote networks has been shown to be satisfactory while not incurring apparent processing overhead. The memory and CPU utilization on the user's machine are negligible with the current prototype and can be further optimized still.

VI. Conclusions & Further Work

This paper has presented an architecture for federated HAN devices and services which enables the secure connection of multiple HANs and facilitates sharing of standard UPnP enabled devices and services across these networks. We have implemented a prototype of this system with preliminary evaluations for multiple users and different UPnP devices. It empowers users with simple yet flexible control over their resources how they share these resources with their friends. Initial evaluations have shown the approach to be reasonable with satisfactory performance and small overheads.

Ongoing work is undertaking further detailed evaluation and, in addition, extending this approach to include other types of services, including traditional web-services, Jini and OSGi services. This work will also incorporate ongoing work by the authors to support managed compositions/mash-ups of devices and services in a managed manner, focusing on end-to-end monitoring and fault management [14].

Acknowledgment

This work was partially funded by the Irish Government in the SFI Strategic Research Cluster ("FAME"): 08/SRC/I1403 (www.fame.ie)

References

[1] B. A. Miller, T. Nixon, C. Tai, M.D. Wood. "Home Networking with Universal Plug and Play". December 2001. IEEE Communications Mag.
[2] D. Stirling and F. Al-Ali, "Zero Configuration Networking", June 2003, ACM Crossroads.
[3] Jini specification: http://wwws.sun.com/software/jini/specs
[4] Digital Living Network Alliance: "DLNA Networked Device Interoperability Guidelines Expanded", http://www.dlna.org
[5] R. Chowdhury, A. Arjona, J. Lindqvist, and A. Ylä-Jääski, "Interconnecting multiple home networks services," International Conference on Telecommunications (ICT 2008), pp. 1-7, June 2008.
[6] H. Yong Lee; J. Won Kim; "An Approach for Content Sharing among UPnP Devices in Different Home Networks," IEEE Transactions on Consumer Electronics, vol.53, no.4, 2007.
[7] D. Kang, K. Kang, S. Choi, J. Lee, "UPnP AV architecture multimedia system with a home gateway powered by the OSGi platform", IEEE Transactions on Consumer Electronics, vol. 51, no. 1, 2005.
[8] J. Kim, Y. Oh, H. Lee, E. Paik, K. Park, "Implementation of the DLNA Proxy System for Sharing Home Media Contents", IEEE Transactions on Consumer Electronics, Vol. 53, No. 1, 2007
[9] XACML 2.0, Oasis standard from: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
[10] C.S. Campbell, E. Kandogan, A. November, R. Barrett, P.P. Maglio. "Policity: an experimental evaluation of policy-based administration in a city simulation," IEEE Workshop on Policies for Distributed Systems (Policy 2005), Stockholm, Sweden, 6-8 June 2005
[11] K. Feeney, D. Lewis, D.O'Sullivan, "Service Oriented Policy Management for Web-Application Frameworks", IEEE Internet Computing Magazine, Nov/Dec 2009, vol 13, no. 6, pp. 39 - 47
[12] P. Saint-Andre. (2004, October) Extensible messaging and presence protocol (xmpp): Core. IETF. {Online}. Available: http://www.ietf.org/rfc/rfc3920.txt
[13] R. Brennan, K. Feeney, J. Keeney D. O'Sullivan, J.J. Fleck, S. Foley, S. v.der Meer. Multi-Domain IT Architectures for Next Generation Communications Providers. IEEE Communications Mag. vol. 48, no. 6, pp 110- 117, Aug 2010.
[14] Z. Etzioni, J. Keeney, R. Brennan, D. Lewis, "Supporting Composite Smart Home Services with Semantic Fault Management", 5th

International Symposium on Smart Home (SH 2010), FutureTech 2010, 21-23 May 2010, Busan, Korea.

BIOGRAPHIES

Zohar Etzioni is a Ph.D. student in the KDEG at TCD. His main research interests are network management, service management, distributed systems, swarm intelligence, software engineering, and agile methodologies. He has 12 years of extensive industry experience as a system and software architect in various domains including telecommunications and medical imaging. He holds a M.Sc in computer science from the Open University Israel and MA in philosophy from Tel Aviv University, Israel.

Kevin Feeney is a research fellow with the KDEG at TCD. He holds Ph.D. and B.A. (Mod) degrees in computer science from TCD. He has experience as a researcher and software developer, designer, and architect in a number of national and international companies since 1997. His research on policy-based management has been widely published in several high-quality international journals and conferences.

John Keeney is a research fellow with the KDEG in the School of Computer Science and Statistics at TCD. His research focuses on the use of semantics in the management of autonomic adaptable systems, particularly networking and telecoms systems. He graduated from TCD in 1999 with an undergraduate degree in computer engineering. His Ph.D. in computer science, also from TCD, was completed in 2004. He has published in excess of 30 papers in significant journals, conferences, and workshops.

Declan O'Sullivan is director of the KDEG at TCD, and has over 20 years' R&D experience in both industry and academia. He holds Ph.D., M.Sc., and B.A. (Mod) degrees in computer science from TCD. His particular research interest is in knowledge driven approaches to achieving semantic interoperability, especially applied to network and service management in distributed networks. During his time in industry, he was involved in industry and fora such as TeleManagement Forum and Object Management Group (OMG). He has over 70 publications, and has contributed to several organizing and program committees in this field.