Deliverable D4.1: WP4 Report

Editor:	Dirk Pesch (CIT)
Release:	FINAL
Distribution :	Public Deliverable
Document Number :	**D4.1**

1 Introduction – D4.1

In order to evaluate the concepts, algorithms, architectures, designs, etc. developed in work packages 2 and 3 as well as through the three research themes of the M-Zones programme, experimentation platforms are required. This document provides an overview of the current status of development of experimentation platforms and their use as a research tools within M-Zones.

Due to the diversity of research interests and research foci more than one experimentation platform is required. An experimentation platform can be a software implementation of a model of a real world system, or it can be a real world implementation – hardware and software – of a smart space under study, or it can be a hybrid between real-world implementation and a software model in, for example, a computer simulation environment.

In this document the following outline template is used as a basis for describing the various platforms and testbeds in use within M-Zones.

Platform Title

Overview

The specification of the experimentation platform is included here. There is no definite template for the specification defined as it depends very much on the type of platform, e.g. hardware/software, simulation, or a real system.

Experimentation Methodology

Here, the methodology used to realise the platform or testbedis described. This is for example a stochastic event driven computer simulation or a real-world test-bed such as a real smart space installation.

Architecture and Design

Here a description of the architecture and the design of the experimentation platform is provided. This will typically include diagrams depicting the architecture, description, with references, of the design approach used, e.g. software design methodology, etc.

Evaluation Methodology

Here a description of the evaluation methodology used in evaluating and assessing results is described. In the case of a stochastic, discrete event driven computer simulation the evaluation methodology would be based on statistical data analysis methods such as Moments or Batch Means evaluation of random variables and random processes under observation within the simulation platform.

Use within M-Zones Research Programme

Here the use of the platform within M-Zones is described. This will include the theme(s) the platform fits into, reference to the problem areas identified in other deliverables and the state of the art review, etc.

---

2 WIT Experimental Platforms

2.1 Mobile IPv6 Testbed

2.1.1 Specification

The Mobile IPv6 platform is developed for testing mobile devices roaming from one smart space to another. Its purpose is to test:

- Uninterrupted use of services from one space to another;
- Security negotiations between smart spaces;
- Management of network bandwidth;
- Security issues relating to roaming between smart spaces.

This testbed will also be used in cooperation with the WIT UPnP services testbed (described in the following secton).

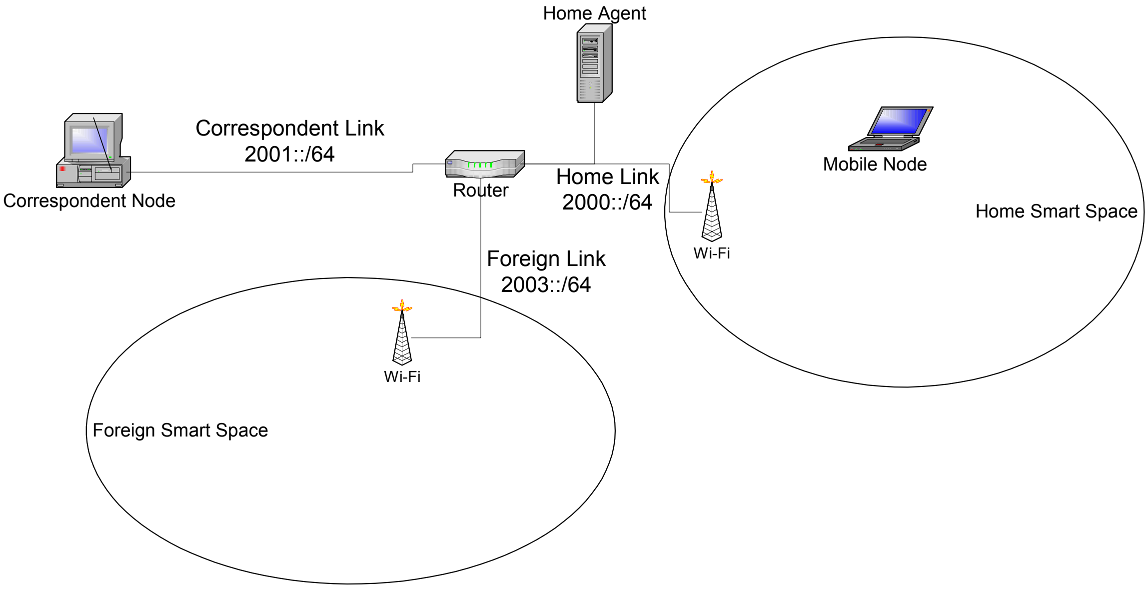
2.1.2 Experimentation Methodology

The testbed will be used to analyse aspects of:

- Maintenance of media streaming service sessions whilst roaming from one network to another;
- application level Quality-of-Service whilst roaming from one network to another.

2.1.3 Architecture and Design

- As illustrated below the test bed is made up of a server/router, a laptop and a desktop PC. Attached to the router are two wireless access points acting as two separate Wi-Fi networks. Each computer within the testbed is running Linux recompiled to support Mobile IPv6 for Linux (MIPL). More specifically:A Laptop with a Wi-Fi card will act as the mobile node;
- A workstation attached to a network interface of the router will act as a correspondent node;
- The Router will have three separate network interfaces:
    - A wireless access point on the home link for the Mobile Node when it is on its home network.
    - A wireless access point on the foreign link for the Mobile Node to roam to.
        - These two networks will each represent a distinct smart space.
    - A network for the correspondent node
        - This represents an external node communicating with the mobile node.
- A workstation attached to the Home Link, will act as the Home Agent of the Mobile Node.



2.1.4 Evaluation Methodology

The testbed will be used to perform a number of experiments involving users and services roaming between smart spaces. The following criteria will be used to assess the ability of Mobile IPv6 to provide seamless handover in the experimental scenarios:

- Efficacy: does Mobile IPv6 enable user/service mobility in a functionally correct manner?
- Efficiency: is Mobile IPv6 sufficiently efficient (in terms of response times etc.) to allow service session handover in a manner that is seamless from the perspective of the end user?
- Scalability: is Mobile Ipv6 likely to provide a scalable solution for user/service mobility?
- Security: are the handoffs between networks secure?

2.1.5 Use within M-Zones Research Programme

The purpose of setting up the Mobile IPv6 testbed is so we can define a smart space as its own network and allow users to roam with their user devices from one smart space network to another with ease. It will also provide a method of testing service and device migration between smart spaces. The management of mobile devices and services is a complex area, and with this test bed, major issues regarding security, management, and administration may be addressed. The testbed will contain two smart space networks.

2.2 Nomad Platform

2.2.1 Specification

The NOMAD platform is a set of common components, exposed through a well-documented and portable API, which can be utilised for smart service creation. The platform will consist of core components of service architectures such as authorisation, user management, subscription and profiling. This component suite will be built upon an abstract base hierarchy that will provide the platform developers with a formal and manageable authoring process. In addition there are experimental platform components that will be developed that will give the platform enhanced capabilities. These experimental components are:

- Groups component, that will support community services such as forums, blogs. Any service that has a community element will be able to take advantage of this facility. The group's components handle all group creation, member management;
- Messaging component, that will allow services communicate with each other; it will allow an instant message be sent to a user, the messaging component will determine the output channel of that message depending on how the recipient is connected to the platform;
- Trust component, will incorporate the idea that communication/information sharing between users is dictated by a level of trust. Two users with a high trust level between them will share more personal information with each other than 2 users with a lower trust level. The trust level will be allocated by the user, but a so called Ambient Intelligence Engine (AIE) will learn from this input, to the point that it will be able to allocate trust levels automatically;
- Presence component, this will handle the current status of a user. The status can be online/offline/busy. Services will have access to this information with the ability to modify it;
- Location component, this will handle the current location of a user. It will store the user's coordinates and a history of where they have been.

A suite of sample services will be developed to test and evaluate the platforms capabilities.

2.2.2 Experimentation Methodology

The testbed is intended as a platform that will support the composition of 4 categories of services:

1. Informational, these can be services that provide news on demand, blogs
2. Community, these can be services that provide community interaction such as forums
3. Communication, these can be services that provide user's with the ability to communicate with other users
4. Challenge, services that provide an entertainment value like games, possibly also incorporate a community aspect such as multiplayer games

The experimentation methodology will be a well-structured development cycle. The steps involved in this experimentation will be:

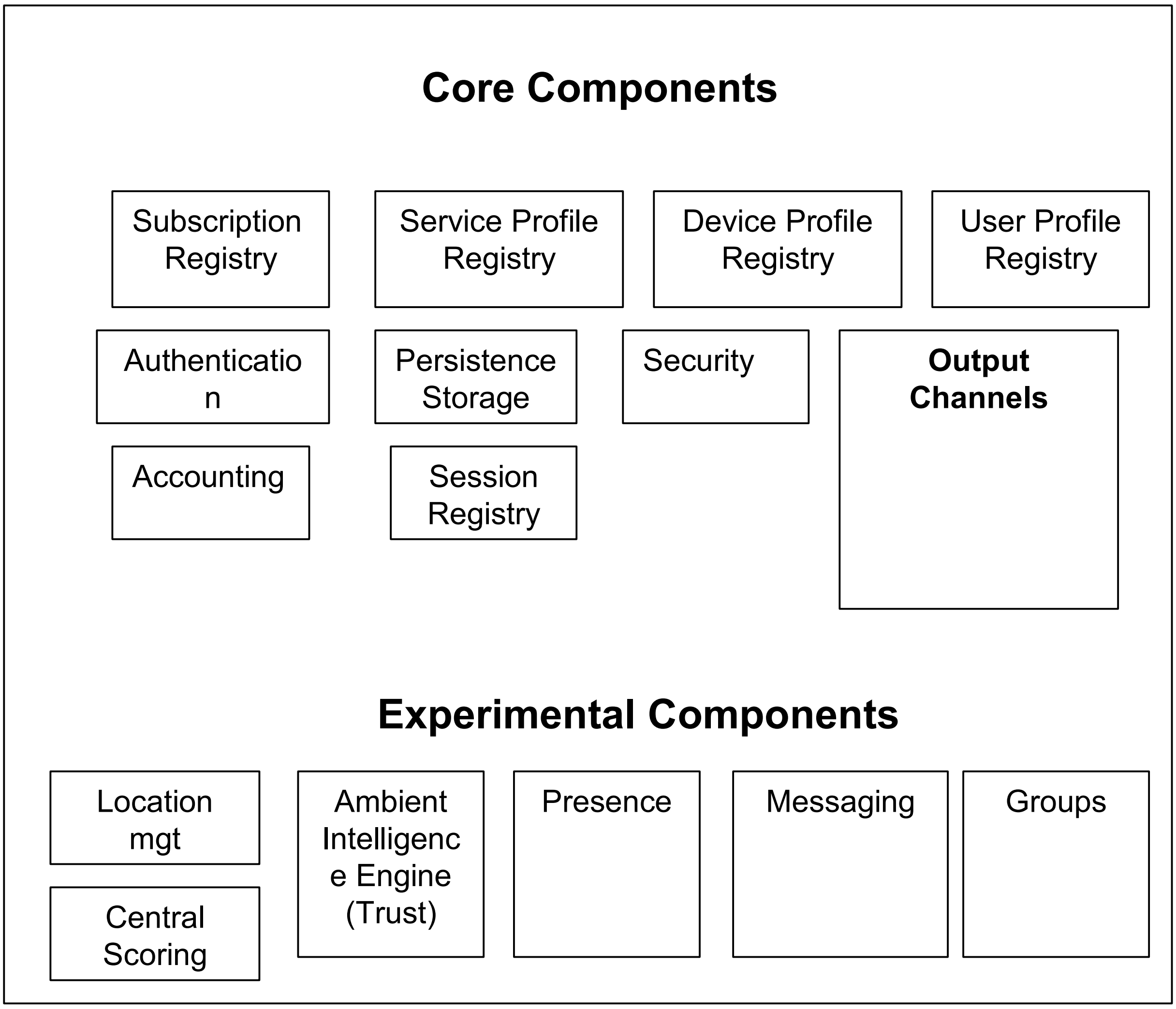
1. Assembling the development testbed as described.
2. Choose a service for each of the above categories to evaluate the platform components.
3. Develop a documented API that will allow service creators access to the platform components.
4. Specifying compositions of services and repeat the previous two steps.

2.2.3 Architecture and Design

The platform architecture will consist of java-coded components that will run on a standard java-enabled pc. The components will communicate with each other using RMI calls.

The platform will be structured as follows:

1. Java core components
2. Java experimental components
3. LDAP server for user authentication
4. Mysql server for persistent data storage
5. Tomcat to run a portal framework for any portalised services



2.2.4 Evaluation Methodology

The evaluation of the platform will revolve around the services that are developed. They will evaluate stability, scalability, performance issues and security concerns. A portal will also be developed to evaluate the ease of use of the services in a customisable environment.

2.2.5 Use within M-Zones Research Programme

This platform addresses issues involved in theme 3, "Engineering Open Smart Spaces", as it could possibly be incorporated within a smart environment to provide both legacy and experimental components for smart services.

The platform will be used as a testbed for the deployment of smart services. Experimental components like location based services and context aware services will play an integral part of the platform. An ambient intelligence engine (AIE) will be used for trust calculations, for the allocation of personal information between users.

The problems we envisage for this testbed include:

- Providing context aware information, Creating automated trust level allocation,
- Providing efficient security. Scalability to large volumes of smart space users.

2.3 UPnP testbed for the use, control and management of Smart space devices and their services

2.3.1 Specification

The aim of this testbed is to:

- Develop an architecture for the use, control, and management of smart space devices and their services, utilising Universal Plug and Play technology.
- Develop a hierarchical based management system for administration of smart devices within multiple smart spaces.
- Investigate the management requirements of smart space devices.

This testbed will be using both hardware and software based devices to evaluate the system. The testbed will be built on top of the Mobile IPv6 smart space networks described above.

2.3.2 Experimentation Methodology

Firstly a small range of simple services will be developed for testing that will represent the following four areas of service usage.

1. Media Streaming (“session”)
2. Service Control (“message passing”)
3. Service Querying (e.g. state of a device)
4. Asynchronous eventing (subscription, notification)

When the services are developed, they will be integrated into the UPnP testbed. These four areas of service usage will be the basis of the experiments within the test bed. The services must be able to be controlled by a central service broker known as the Local Control Point (LCP). The LCP must be able to set up service sessions between the services and the users, discover new services, and allow users to subscribe to service events. The Administrative Control Point is a manager of the Local Control Points. This keeps track of what LCP is governing what smart space network. It will add additional administrative functionality such as setting up profiles and access rights to the services for the users registered to the M-Zone system.

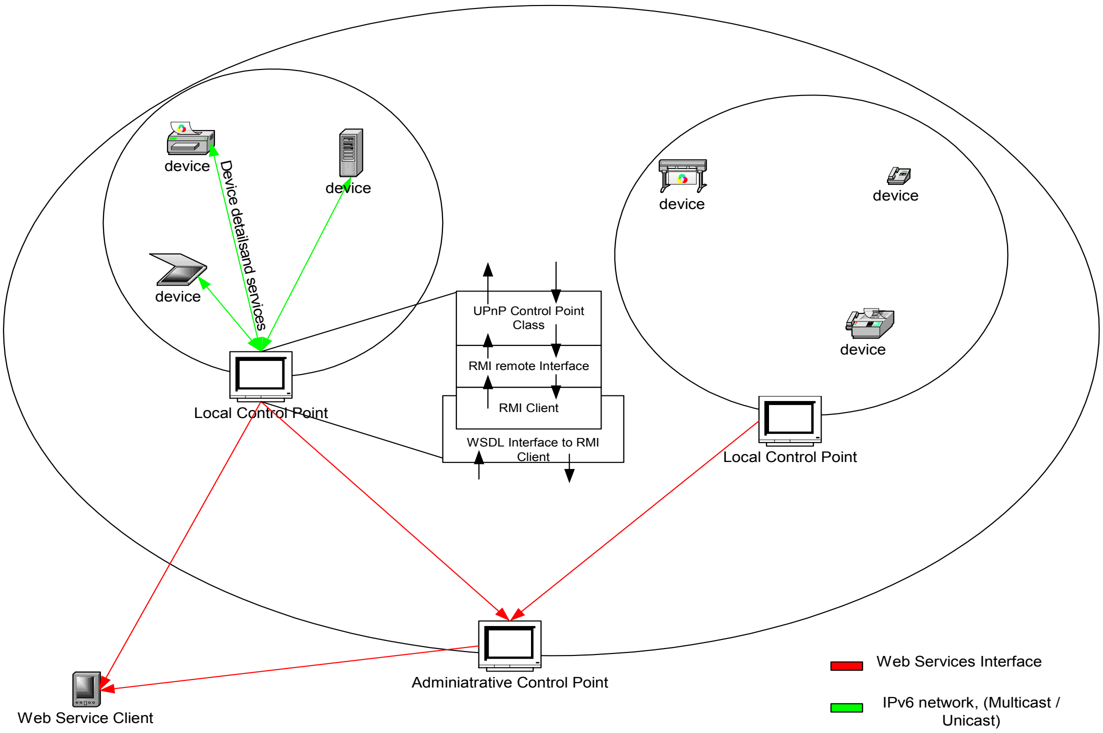
2.3.3 Architecture and Design

This testbed is a software and hardware implementation of smart devices within a smart network.

The technology that will be used is as follows:

- For device and service discovery, use and control a java implementation of the UPnP protocol will be used. This is an IPv6 enabled implementation.
- To allow external use of the smart space control point, the control point uses Java RMI (Remote Method Invocation) to allow its methods to be externally visible.
- Apache Tomcat and Apache Axis will be used to allow the RMI client be accessed through a web services interface.
- The Administrative Control Point (ACP) will be a web service client of the Local control points of the smart spaces. The ACP will have priority over management functionality and administration rights to devices and services within the smart spaces.
- A Web services client can use and control devices and their services through the web services interface of the Local Control Point.

- The Administrative Control Point can also provide its services through a web services interface, with security issues having to be resolved.



The advantage of this architecture will be that new devices can be added and removed from the smart space at will. There is also the possibility of having multiple control points running within the same smart space. A web services client can be almost any device with a network connection making the interaction between the devices and users completely heterogeneous with the only dependencies being the language of communication SOAP.

2.3.4 Evaluation Methodology

This testbed will be used to evaluate the proposed service architecture using criteria such as

- Ease of integration of new services.
- Level of heterogeneity available i.e. devices with different implementations of the UPnP stack should be controllable by any type of web service client;
- Responsiveness of the system.
- Extent of management functionality of devices within the smart space.
- Also mobility issues of User device clients roaming into smart spaces to utilise the smart space services. Or Smart devices themselves moving from one smart space to another (Mobile IPv6 test bed).

With this list of criteria the system can be evaluated and adjusted where needed. This may also bring up issues not yet considered that may have to be looked at in more detail.

2.3.5 Use within M-Zones Research Programme

This testbed will fit into theme 3 of the M-Zones research themes “Engineering Open Smart Spaces”. It will be the basis of smart device control within the WIT M-Zones testbed. It will provide a way of testing management issues between users and devices, and also management between smart spaces with regard to devices and their services.

3 CIT Experimental Platforms

The following are the descriptions of the Experimental Platforms being developed within the M-Zones programme at CIT. The descriptions focus on the architecture and purpose of the platforms with a view towards the experiments planned in the period up to the next deliverable.

The platforms covered are:

- *Heterogeneous Wireless Network Testbed:* This testbed is a computer simulation environment based on the CNCL simulation class library and implemented in C++. The testbed facilitates experimentation within a simulation model of a large scale heterogeneous wireless network. Access to resources is managed in an intelligent fashion based on policy-based management concepts. The environment when fully operational will allow testing of concepts for smart access to wireless communication resources.
- *Ad-Hoc Network Computer Simulation Testbed:* The ad-hoc network computer simulation environment is a comprehensive ad-hoc network simulation and experimentation environment that provides a platform for the development and test of multiple/medium access control, routing, address assignment, and services in an ad-hoc network environment. The simulation environment is implemented in C++ using the CNCL simulation library.
- *Ad-Hoc Network Services Testbed*: The ad-hoc network services testbed will be based on the Java/Jini service management environment. The testbed is based on physical devices, laptops, PDAs, mobile phones, and facilitates testing management of Jini enabled services within a smart space.
- *AWS Smart Home:* The smart home is a physical experimentation environment with the appearance of a one bed-room studio flat. The smart home will allow the implementation of management experiments to test new management and control concepts for wireless home environments. The smart space will include a wireless network for delivery of information and home entertainment and educational services as well as sensor networks for control of the ambient environment and its users.

3.1 Heterogeneous Wireless Network Test-bed

3.1.1 Specification

- Develop a call admission control mechanism for heterogeneous wireless networks using a policy based management architecture.
- Develop a handover management scheme to select and force users to handover connections to other available networks so as to maintain guaranteed QoS contracts to the end user.
- Investigate optimisation using evolutionary computing techniques (W. Pedrycz, 1998) so as to arrive at an optimal balanced load, spectrum efficient resource usage and maximum QoS to the end user.
- Develop an authentication policy to control user access to the networks within a heterogeneous environment based on user profile/role.
- Investigate the effects of service composition on the call admission/handover decision-making process.

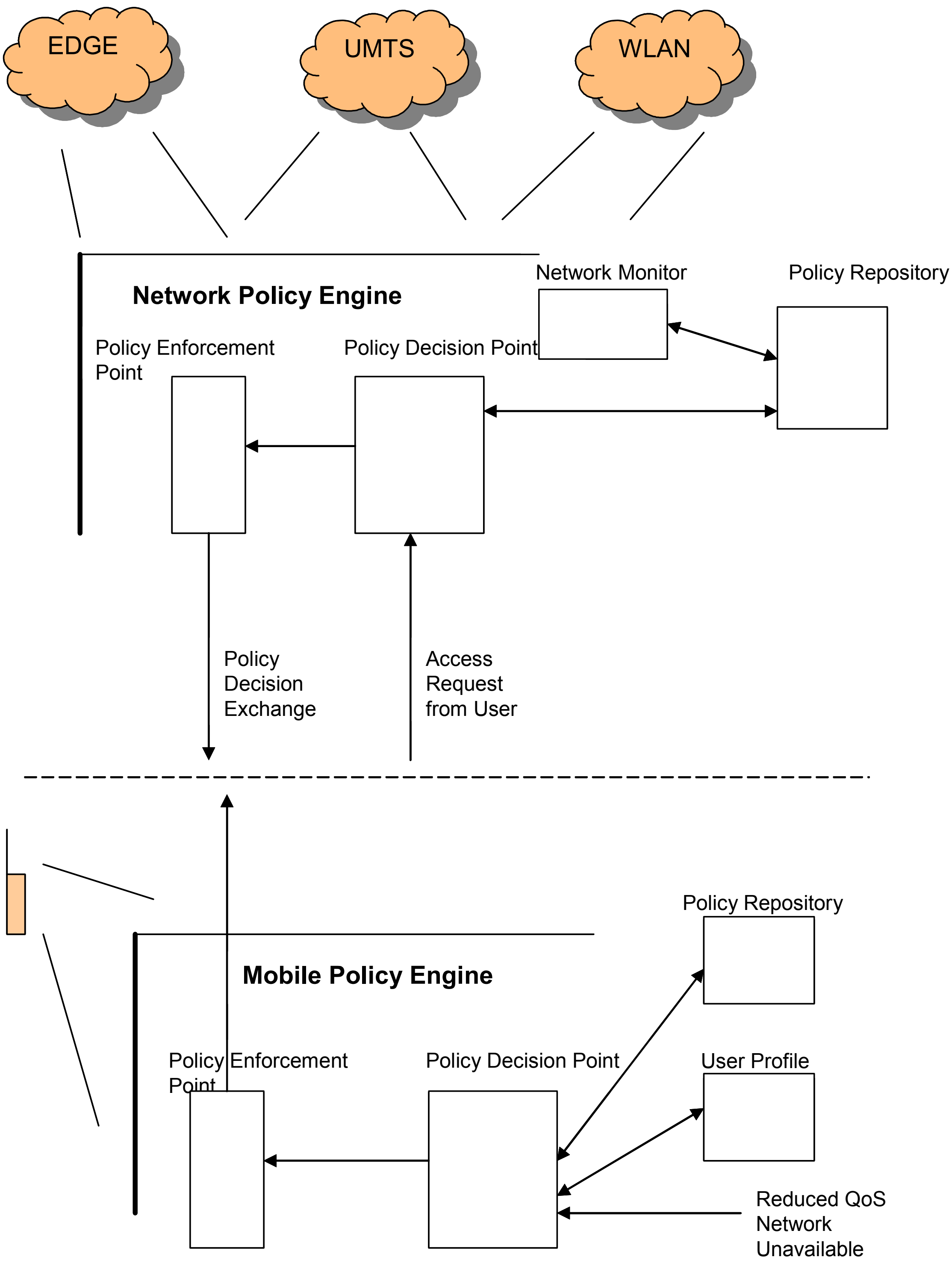
3.1.2 Experimentation Methodology

The heterogeneous network testbed is a stochastic event-driven computer simulation model of a heterogeneous wireless network. The model will focus on simulating the radio access layer of EDGE, UMTS and IEEE 802.11b/g WLAN. Multi-modal terminals will roam between the available access technologies. Policy based management (D. Kosiur, 2001) will control the call admission and handover management schemes as described below. Initially, three types of services will be implemented – real-time voice and video streaming and non real-time interactive web services. The heterogeneous network testbed will be used to assess the load balancing and QoS provisioning capabilities of the policy based call admission and handover management schemes. A number of test scenarios (user load, service mix, QoS requirements) will be used to evaluate the proposed schemes.

3.1.3 Architecture and Design

The testbed is a software model of a heterogeneous wireless network. The model is developed using C++ and the CNCL class library for event driven simulations (M. Junius el al, 1999). The simulation platform is structured as outlined in the figure below and contains the following components:

- The physical and radio access layer for UMTS, EDGE and IEEE 802.11b/g.
- Multi-modal terminals that can tune to any of the available access networks.
- User profiles defining user preferences, required QoS for each class of service, user role etc.
- Network Policy Engine for making call admission and handover request decisions.
- Mobile Policy Engine which decides when a mobile should issue a intersystem handover request due to reduced QoS, network coverage etc.
- Policy Repositories that contain network performance data used by the policy engines in the decision making process, such as video frame drop rate (FDR), block error rate (BLER), CIR, call blocking rates.
- Network Health Monitor. This entity periodically reports the network stability at the link level. It uses the performance data held in the network policy repository in the network health calculation. A networks health value varies between 0 and 1.



Network Architecture

The simulation platform considers a 49 cellular system. Continuous coverage for both UMTS and EDGE is provided throughout the network. A cluster of WLAN access points is positioned within a small number of cells, providing *hotspot* WLAN coverage to mobiles in that area. The inter-arrival time between service requests can be selected to reflect the most suitable model for a particular type of service. The inter-arrival time can be varied for different load scenarios.

At the call initiation stage, the network policy engine is enacted to make a decision to which network to allocate the new call request. The initial policy is to select the least loaded network while observing a service mix specified by the network operator. This service mix will initially be 60% of all resources to be allocated for voice service, while web and video traffic are allocated 20% each. Defining a service mix in this way allows a network operator to prioritise services and maximise revenue by defining the preferred load for each service type. The service mix can be adapted at run-time by varying the policy parameters. Each network determines its residual capacity via the network health monitor, as described above. The residual capacity is varied depending on the network health value reported by the health monitor.

If the perceived QoS of a current connection falls below the threshold stored in the mobile policy repository, the mobile terminals policy engine is enacted to make a decision whether to issue an intersystem handover request at this time or to continue with the current connection. The intersystem handover request is sent to the network policy engine to request network access.

Mobile terminals may be selected to undergo a forced handover to another available network, if the network to which they have current connectivity becomes congested. The mobile selection policy can be based on last the mobile to arrive with dual mode capabilities, mobile application, QoS requirement etc.

3.1.4 Evaluation Methodology

The call admission control and handover schemes will be evaluated by the degree of load balancing between the available access networks and how efficient the radio resources are been utilised. Link level performance statistics such as the mean BLER, FDR and CIR will be used to assess the QoS at the link level offered to the end users. Link level performance statistics and capacity measurements will be gathered at each stage of development to ensure that the overall architecture is functioning as expected.

3.1.5 Use within the M-Zones Research Programme

The simulation platform will provide a method to evaluate call admission and mobility/handover management scenarios in a heterogeneous wireless network as outlined in (K. Murray, 2003). Although this work concentrates on policy, the platforms modular architecture will facilitate a “plug and play” type approach where other methods of call admission and mobility management other than policy can be investigated. This simulation platform fits into the Network Connectivity and Configuration Theme.

3.1.6 References

Witold Pedrycz (1998), “Computational Intelligence An Introduction”, CRC Press, ISBN 0-8493-2643-5

Dave Kosiur (2001), “Understanding Policy-Based Networking”, Wiley Computer Publishing, ISBN 0-471-38804-1

M. Junius, M. Steppler, M. Büter, D. Pesch (1999), “Communication Networks Class Library”, Aachen University of Technology, Germany

K. Murray, D. Pesch (2003), “Admission Control and Mobility Management in Heterogeneous Wireless Networks”, M-Zones draft whitepaper, 18th Feb. 2003

3.2 Ad-Hoc Network Computer Simulation Testbed

3.2.1 Specification

The computer simulation implementation of ad-hoc networks is aimed to investigate three distinct aspects of ad-hoc networks:

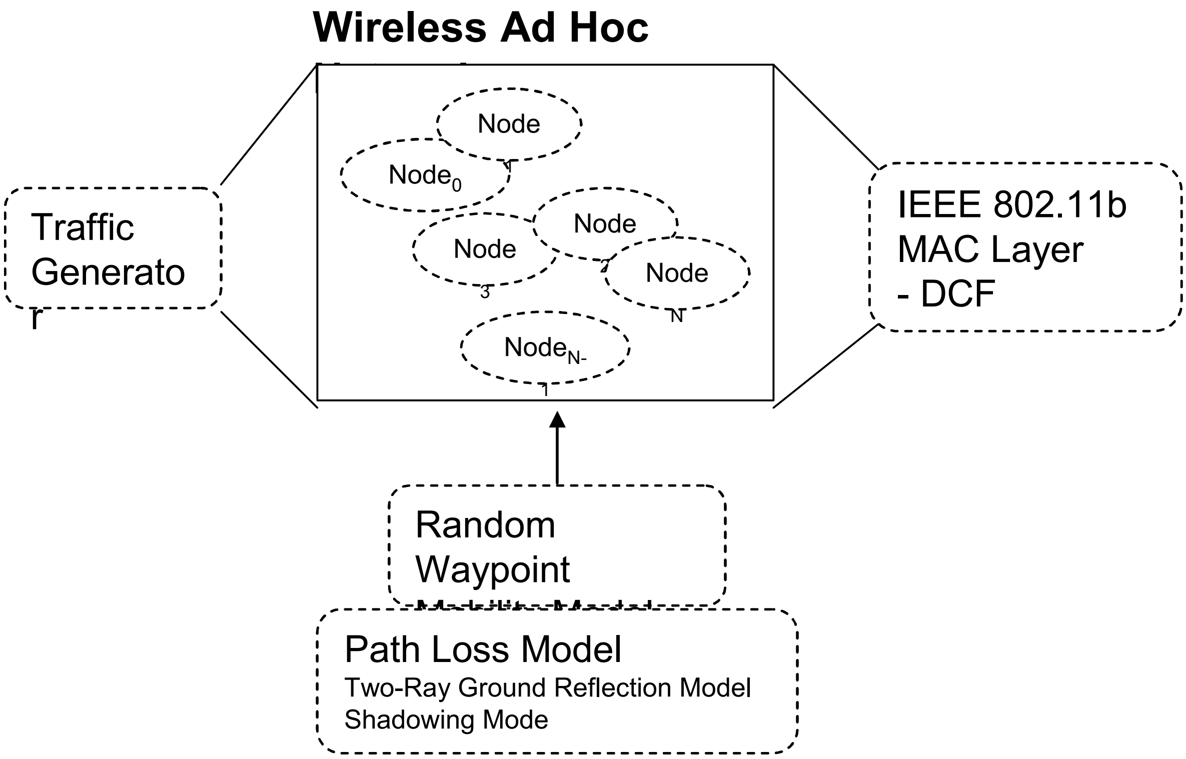
1. Routing
2. IP address assignment/reassignment
3. Service discovery

3.2.2 Experimentation Methodology

A simulation model of an ad-hoc network is being constructed that will analyse a multi-hop ad-hoc network and will be capable of recording the various delays in routing packets to their destination. The simulation experiment will simulate many such clusters for investigating the problem of IP address assignment as new nodes join the network, or as an existing network expands to come within the transmission range of another network.

3.2.3 Architecture and Design

Routing simulation architecture



The Ad Hoc network model is implemented in C++ using the Communication Network Class Library (CNCL). For network simulation the system is modelled so that the performance of the system can be expressed in terms of generating and transmitting events, with events being likened to data packets. The network is modelled using the following objects:

- Traffic Generator – this is a basic traffic model with every network node generating N (=1,10,100…) packets per second for a random destination with a negative exponential distribution. This packet traffic generator model is for evaluation purposes only and application based traffic models will be later incorporated.

- Node – models a mobile wireless device.
- DSR routing protocol.
- MAC Layer – IEEE 802.11b with DCF functionality.

3.2.4 Evaluation Methodology

Evaluation is based on statistical data analysis for DSR & Mac Layer evaluation & testing

- Packet throughput
- Packet roundtrip time
- Route discovery overhead associated with successful packet delivery
- Effectiveness of route caching techniques

3.2.5 Use within M-Zones Research Programme

The computer simulation environment will provide an experimentation platform for radio resource management issues in ad-hoc networks. It is expected that these ad-hoc networks will form the wireless communication infrastructure of smart environments in particular in the home and the work place. This testbed will facilitate experimentation with management concepts such as power awareness, routing, address and identity management as well as evaluation of performance of service discovery.

3.3 Ad-Hoc Network Services Testbed

3.3.1 Specification

A test bed implementation of ad-hoc networks is being constructed to investigate the operation of intelligent Jini based services within an ad-hoc network for smart environments. Clients and services should connect wirelessly to the network and utilise the services within the network. An Experimental test bed would include a collection of PC's, Laptop's, Printer's and PDA's. Using Jini, clients can also use the functions of Jini Services both near and far. Having knowledge of their IP address the client can directly access them over the network or using multicast requests locate all services within a short range. The test bed can be used to evaluate the operation of Jini over different network types such as 802.11b and Bluetooth.

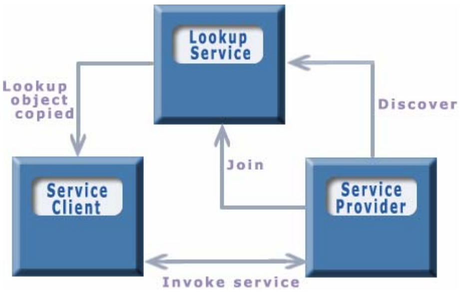
3.3.2 Experimentation Methodology

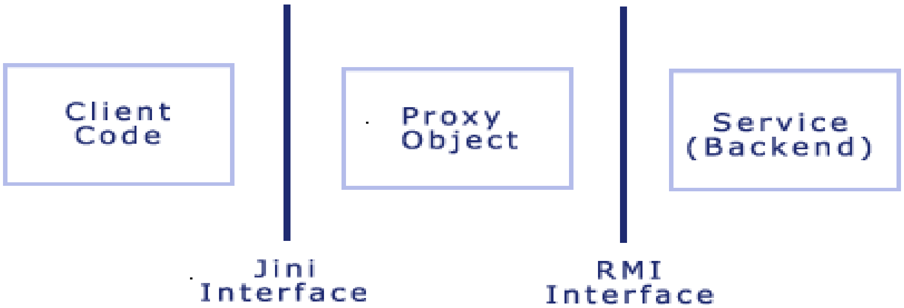
The experimentation methodology is based on the physical construction in hardware and software of a testbed for service management research. Equipment used in many real-world situations will be used which allows the realistic testing of concept and management solutions. Much of the infrastructure will be developed within software.

3.3.3 Architecture and Design

A Jini network consists of a network of many services, which can be combined into groupings known as federations. A Federation consists of look-up servers, clients and services. Each Federation must have one or more Look up servers, as this is where all of the client and service information is exchanged. If a Jini Service wishes to join a federation it must first locate a Lookup service where it can advertise itself. Thus if a service fails to find a lookup service it is considered useless within the Jini Federation. We have thus developed services, which first search for a lookup service. However if after a specified time they fail to locate a lookup service they can themselves act as a lookup service.

A Jini service will be implemented using a proxy on the client side and a service backend on the service provider side. There are a number of different of options one can use when using proxies. The Proxy could be used as the service so that the backend is very thin and the proxy very fat- performing all the work. The Opposite extreme is where all of the processing is done on the service side. The proxy just exists on the client side to take calls from the client, invoke the method in the service and return a result to the client. These proxies can be RMI or non-RMI. We have chosen an RMI thin Proxy, which simply provides a means of accessing the service backend through an interface. We have taken two services as examples. The first is a simple Print service where the proxy provides a means for the client, being unable to print, can perform the print command on the service. The second is a File Display application. This allows the client to view all the files present on the service and display these files on the client's display.





3.3.4 Evaluation Methodology

We are currently evaluating the time taken to perform the different processes involved in operating within a Jini Federation. The time taken to search and join a lookup service, advertise a service, download a proxy and complete service delivery. Current tests are over fixed and wireless networks using a PC and Laptop. It is hoped to extend testing over Bluetooth and to investigate operation using PDA's. We also wish to test the operation of the services using proxies of different sizes.

3.3.5 Use within M-Zones Research Programme

The two example services specified here demonstrate a means by which services can be delivered and used within M-Zones. A Jini Client can access printers within a federation without itself being configured to use a printer. It can use the functionality of other service without having the ability to itself perform the task. Likewise a client can access all or some of the files present on one or a number of services. They can then view these. Once the display window is closed the document or image no longer exists on the client, thus eliminating the need for storage space to hold extra files. They can be displayed but don't have to be stored. This is a useful service for devices within an M-Zone.

3.4 The AWS Smart Home

3.4.1 Specification

The M-Zones team within the AWS Group at CIT are currently working on the adaptation of a classroom space into a research space with the appearance of a one-bedroom apartment. This research space will contain sitting, kitchen/dining, sleeping and washing areas and will be used as a living smart home laboratory to evaluate M-Zones technology in as real an environment as possible.

3.4.2 Experimentation Methodology

The experimentation methodology is to provide as diverse a range of research problems as possible within a single “domestic space”. The “physical” methodology is to construct a space which will look like a modern apartment but with a hidden infrastructure allowing the deployment of both wired and wireless smart home technologies for delivery of information and entertainment services both for normal home living and for home office applications. The space will also be made an “aware” one through the deployment of a sensor network to identify and track users of the space, to manage the ambient environment, and to manage energy usage. A diverse range of services will be installed to facilitate research in hardware, networking and software for domestic smart spaces. The community of potential users considered will also be wide, including people with varying levels of technical ability and people with physical disabilities. The AWS Smart Home will be linked to both WIT and TCD to allow research on information transfer and mobility between smart spaces.

3.4.3 Architecture and Design

The design of the layout and fittings is currently underway. The physical appearance will be that of a modern home connected to state-of-the-art electronic services using commercially available equipment. The infrastructure will be such that new services and appliances relevant to the research objectives can be readily installed.

3.4.4 Evaluation Methodology

The space will allow a very wide range of research and tests to be carried out and the evaluation methodology will be different for each one.

3.4.5 Use within M-Zones Research Programme

The AWS Smart Home will support all aspects of CIT research within M-Zones: networking, software and hardware. It will particularly support the presentation of real world problems in a real environment in a way, which is impossible to duplicate conceptually or in simulations. It will therefore facilitate the transfer of research from the abstract to the concrete as well as providing a testing ground for the evaluation of new concepts. As mentioned above, it will also be used to examine linking of smart spaces and the mobility of users, information and services between them.

4 TCD Experimental Platforms

The following are the descriptions of the Experimental Platforms being developed within the M-Zones programme at TCD. Broadly speaking these descriptions focus on the experiments planned in the period up to the next deliverable, i.e. approximately 6 months.

The platforms covered are:

- *Nomadic Collaborative Smart Space Testbed:* This is a general service oriented testbed that will be assembled and tested with the aim of supporting future practical experiments and demonstration with the results from other, more focussed adaptive smart space experiments. It aims to provide a smart space to support group work and aims to be portable.
- *TSUNAMI Platform:* For experimenting on mechanisms for inferring user intent from gestures and spoken input.
- *Policy-based Management Service Development Testbed:* For experimenting with a community based approach for controlling and evolving organisation management using policies.
- *Semantic Service Development Testbed:* For experimenting with approaches to the specification of ontology-based service descriptions and supporting rapid prototyping of these.
- *Adaptive Service Composition Engine Evaluation Testbed:* For experimenting with a mechanism for mapping task descriptions to service composition using pre-existing service composition patterns.
- *Assurance Service Development Testbed:* For experimenting with a policy-based mechanism for managing aspects of a service composition and in particular assurance of quality of service requirements.
- *Context-aware Development Testbed:* For experimenting with the application of Topic Maps to the assembly and management of common context information.

4.1 Nomadic Collaborative Smart Space Testbed

4.1.1 Specification

This testbed aims to:

- Provide a practical service oriented testbed for collaborative smart space applications.
- Provide a portable demonstration platform for various adaptive service results.
- Support investigation into the practical problems of operating smart space infrastructure.
- To investigate the concept of a nomadic smart space.

4.1.2 Experimentation Methodology

This testbed is intended as a platform for a variety of experiments including:

- Dynamic composition of existing smart space services.
- The policy based management and access control of shared services.
- The deployment element of the semantic service development lifecycle.
- Initialisation and management of portable smart spaces.

Assembly of the platform itself will require the solving of several technical problems:

- *The selection of a suitable application server*. A J2EE one is favoured, but it must be able to support extension for the required service composition execution, monitoring and management functionality. Flexibility in binding to different distributed communication mechanisms is also needed. Standard Web Service bindings (i.e. HTTP/SOAP) may be used initially, but other service-bindings should be able to be easily implemented, e.g. to

Jini or JXTA, to allow better alignment with wireless and p2p communication mechanisms.

- *The development of decentralised service composition execution control.* This platform aims to support the integration of services within and external to a smart space in any number of service compositions. An efficient mechanism is therefore required for controlling the execution of such service compositions. Control of the execution of control and data flow in a service composition may be performed through centralised workflow engines, however such an approach seems inappropriate to smart spaces, where communication links may be intermittent and therefore the links to the workflow engine represent a single point of failure. A preferred approach would be a decentralised mechanism such as described in (Bentallah et al 2002) , so development of appropriate system support will be required.
- *The development of security mechanisms*: The wireless elements of the testbed present an obvious security problem, so some mechanism must be put in place for ensuring security of information over wireless links, e.g. a wireless VPN. In addition some mechanism is required for enforcing access control to services.
- *The development of atomic services*: To be a useful testbed for other service oriented experiments this platform will need to provide a rich set of services, so that a wide range of services can be used in service composition, service matching, service management and semantic interoperability experiments. To avoid skewing results by having services designed by the experimenters and to avoid the overhead of implementing new services, an external source of service functionality is sought. Two possible candidates are: office applications such as Open Office or Microsoft Office that offer APIs for interacting with applications and the rich set of linux commands, which benefit from well defined semantics (i.e. man pages) and are amenable to management (e.g. through environment variables, process priority, packet tagging). In both cases the existing functionality need to be wrapped in an appropriate service offering software.

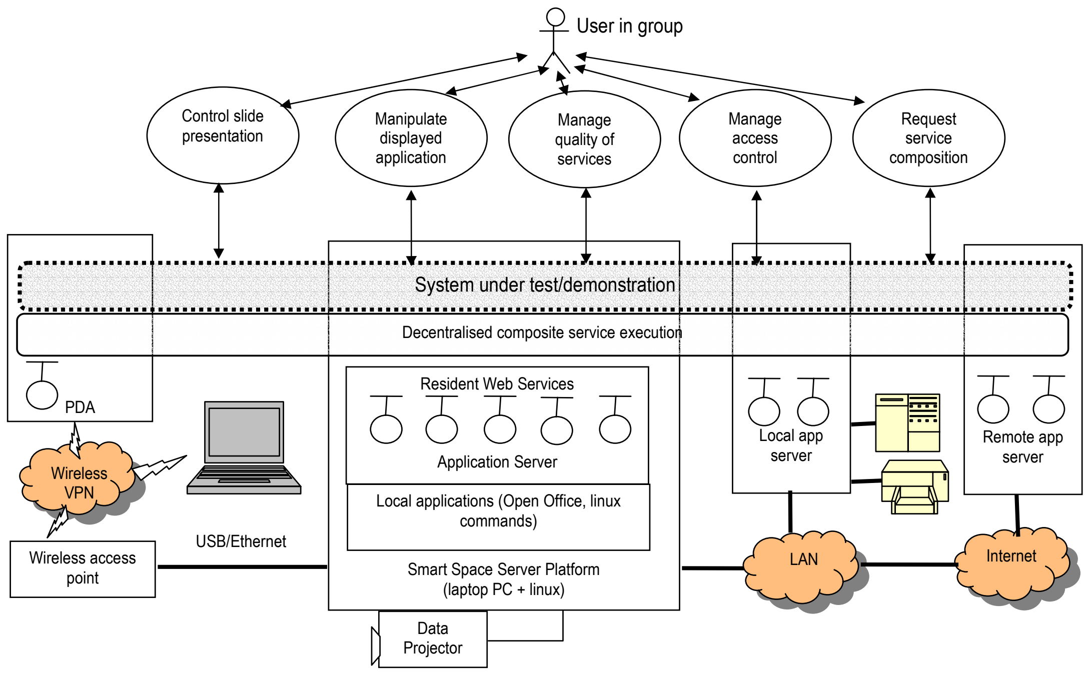
4.1.3 Architecture and Design

The platform supports applications and adaptive mechanisms for using web services on the smart space server, on personal devices, on server residing in the locale and on servers accessible over the Internet.

The platform will consist primarily of a portable smart space server accessible only via web services. This smart space server is intended for use in spaces where group activities such as meetings, lectures or tutorials are conducted. It is managed principally by the person chairing or hosting the gathering, though authorities may be delegated in some experiments. The server features:

- An integrated wireless access point supporting a local wireless VPN for communicating with personal devices held by group members.
- A connected data projector.
- A link to the local LAN and through that to the Internet for accessing other web services or for resident web services that need remote access.
- Resident web services that allow access of office applications and linux commands. These include services to interact with the application currently displayed on the data projector.

This will be integrated with other application servers and service offering and service clients on wireless devices through the decentralised composite service execution platform.



4.1.4 Evaluation Methodology

The evaluation of experiment that use this testbed are left to other experimental platform descriptions.

4.1.5 Use within M-Zones Research Programme

This platform offers a testbed for research under theme 1 on 'Adaptive Services' and theme 2 on 'Managing Dynamic Environments'.

Future developments of this platform may include:

- Using WLAN for connection to the wire LAN
- Using 3G, GPRS or some WLAN based connectivity service for Internet connectivity
- Using a non-VPN security mechanism for communication between personal devices and the smart space server.
- Using non-web bindings for service implementations, e.g. Jini or JXTA

4.2 TSUNAMI[1] Platform

4.2.1 Specification

The experiments for my work on smart space programming divide into three areas. Firstly, a real-world test with limited input axes and tasks. The inputs will most likely be a combination of simple voice commands and gesture tracking, while the tasks could be the manipulation of on-screen objects. These tests will seek to prove the usefulness and reliability of the inference mechanism

Once the fundamental concepts have been proven, a second set of tests in a virtual world can be conducted. This real versus virtual distinction is to allow unambiguous testing of the

[1] TSUNAMI – Tailored Support of Users Natural Activities with Mixed Initiative; see whitepaper

TSUNAMI inference mechanism in the real world, so that potential issues with the simulation are not confused with problems with TSUNAMI. The virtual test will allow for richer input streams and simulated use of declarative tools[2]. These tests will attempt to further refine and expand the inference system to show that it is capable of robust and scalable support of many users.

Lastly, if possible, a larger real-world test drawing on knowledge developed from the first two.

4.2.2 Experimentation Methodology

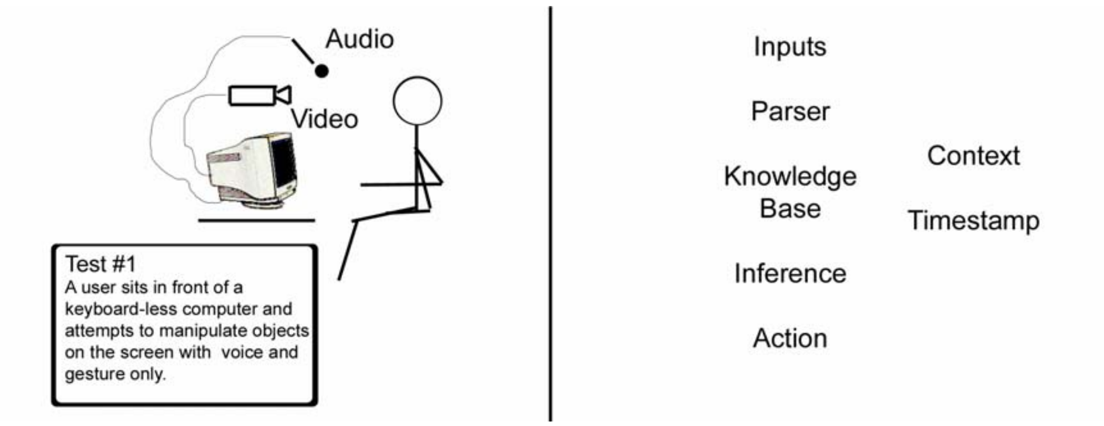
The first test would make use of voice and gesture tracking to enable simple inference by the system. A user could explicitly declare themselves to the system and their profile and inputs could be used to infer goals. Initially tests could be done with a single input axis, voice say, and the inputs could be parsed through recognition software such as Via Voice, and then added to a knowledge base. Representing gestures will pose its own challenges, and I am getting some guidance in this area from Trinity's Computer Vision & Robotics Group.

Entries in the base would also have metadata relating to context and a timestamp. This knowledge base would then inform the inference mechanism and help to derive goals. The activity in this initial test could be as simple as moving shapes on a screen or conducting a slideshow.

A similar, but more complex scenario could be used in the second test, with the potential for simulated interaction with the familiar[3] and ASCE[4]. The final tests would serve as a full-scale demonstration of capabilities and proof of concept.

4.2.3 Architecture and Design

The first two tests will run off a pc platform. The initial test will make use of audio and image input and will need the requisite equipment, see figure 1. It will also require voice recognition and motion tracking software. The second test will require a simulation package capable of creating a navigable, interactive virtual world. The last test, would make use of a suite of technologies from PDA's to intelligent whiteboards.



[2] Declarative Tools – conventional input devices, intelligent whiteboards, etc.

[3] Familiar – an electronic companion accompanying a user and storing their preference and profile data

[4] ASCE – Adaptive Service Composition Engine; responsible for dynamically composing tailored, lightweight, disposable services

4.2.4 Evaluation Methodology

The evaluation process will require volunteers who will attempt to use the system. These volunteers would range in skill from computer novices through to experienced users. Ideally, none of them would have any training on this platform. Evaluation will break down into three questions – first was the system successful at inferring the users' goal, two did it achieve the goal in a reasonable amount of time and three was the user satisfied. The first and third can be observed by a monitor and through soliciting feedback. The second question can be addressed by recording the time taken for successful hypotheses. Metrics will be developed by which results can be measured and success in testing will require passing these metrics.

4.2.5 Use within M-Zones Research Programme

This platform could provide a path into testing some of the other M-Zones components such as adaptive service composition or the integration of the familiar. It fits into M-Zones Theme 1, and will seek to test some of the ideas put forward in the whitepaper (O'Donnell et al 2003a). The first part of the test programme should be completed within the next six months, although the remainder, especially the last part, will take longer.

4.3 Policy based Management Service Development Testbed

4.3.1 Specification

The aim of this platform is to:

- Investigate the application of policy based management to access control and role based management in a smart space
- Investigate the use of a community based approach to modelling the actors in a smart space environment
- Investigate the integration of Policy definition languages and group decision making mechanisms in a policy based management system.
- Support the development of secure access control methods for use in other smart space service oriented experiments.

4.3.2 Experimentation Methodology

The experimentation methodology will be essentially 'build and learn'. The steps involved in this experimentation will be:

1. Assembling the development testbed described below.
2. Specifying a number of logical communities.
3. Applying policy based access control to a number of services in the smart space testbed environment, specified in WSDL
4. Applying policy based access control to a collective resource management system, modified to integrate it with the policy based management framework.
5. Generating graphical test interfaces for community based management of these services.
6. Setting test parameters, exercise the community based access control and community management system via the user interface and monitoring service behaviour and user feedback.
7. Modelling two real world communities through the community based policy based management system and attempting to describe them in terms of policy rules. The two

communities will be: a) an open source development team working on a subsection of the Linux system, b) The Irish Health system.

8. Running sets of simulations modelling common conflict-causing tasks in the two test communities.
9. Specifying alternative community structures and repeating the previous three steps.

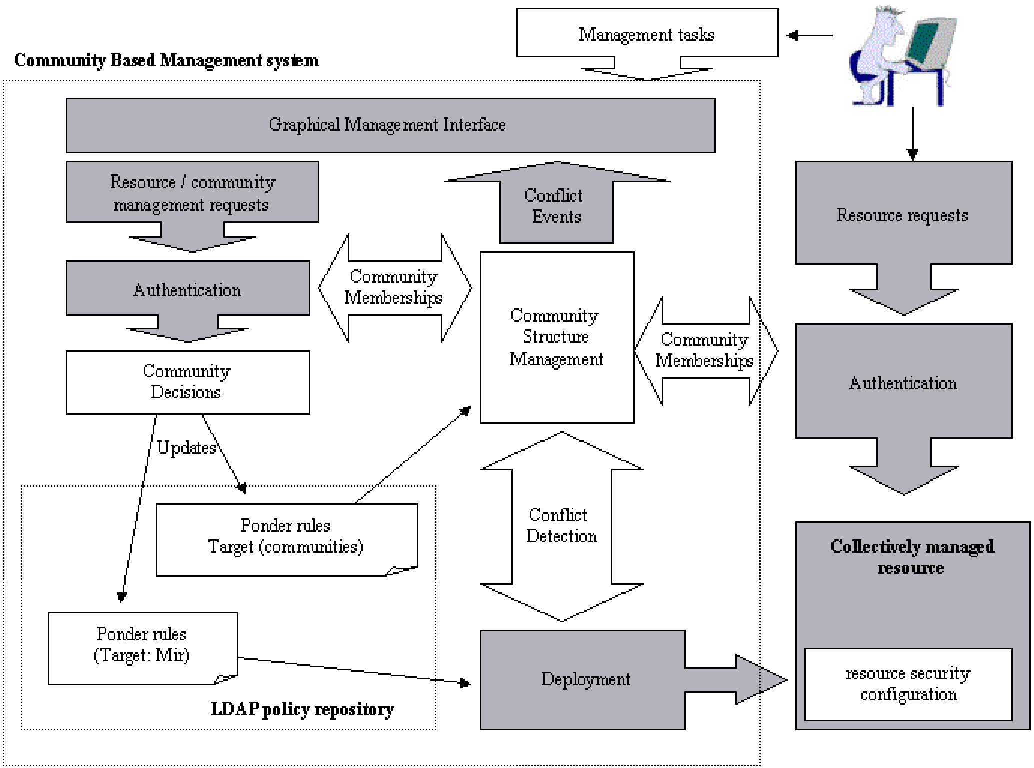
Based on experiences gained from the above step, it is envisaged that the following variables will be refined:

- The initial community structure
- The mapping from WSDL to sets of Ponder policy rules.
- The graphical user interface for describing the community structure

4.3.3 Architecture and Design

The platform will be structured as outlined in the figure below and will consist of the following components:

- A Perl-based community management system
- A Ponder based policy management framework
- A Java based user interface for demonstrating users and resources grouped into domains and communities for management
- A LDAP policy repository
- A collective resource management system, based on MIR & apache-tomcat.



4.3.4 Evaluation Methodology

Results will be evaluated by assessing the usefulness of the tools to the access control management of the services developed in the project. This will be conducted mostly on an informal basis, with the aim being to incrementally improve the usefulness of the tool and in the process gain insights into the pros and cons of using a community approach to distributing authority in a policy based management system

The model communities will be compared with their corresponding real world communities using various metrics to evaluate the efficiency of the solution in comparison to existing arrangements. Examples of metrics to be used: number of updates to managed resources, number of requests to administrators for problems, number of conflicts between community members, etc…

4.3.5 Use within M-Zones Research Programme

This platform primarily aims to addresses engineering issues covered by Theme 3 – "Engineering Open Smart Spaces". In particular it covers issues of managing access control. However, the platform also aims to allow evolution towards resource and network management through the same community management system. [1]

4.4 Semantic Service Development Testbed

4.4.1 Specification

The aim of this platform is to:

- Investigate the rapid prototyping of semantic services based on the DAML-S language.
- Investigate importing and rapid prototyping with externally defined ontologies
- Investigate the use of management information models as ontologies for semantic service definition, in particular the DMTF's CIM.
- Support the development of semantic services for use in other semantic service oriented experiments.

4.4.2 Experimentation Methodology

The experimentation methodology will be essentially 'build and learn'. The steps involved in this experimentation will be:

1. Assembling the development testbed described below.
2. Specifying a number of semantic services.
3. Generating graphical test interfaces for individual semantic services.
4. Setting test parameters, exercise the service via the user interface and monitoring service behaviour.
5. Specifying compositions of services and repeat the previous two steps.

Based on experiences gained from the above step, it is envisaged that the following variables will be refined:

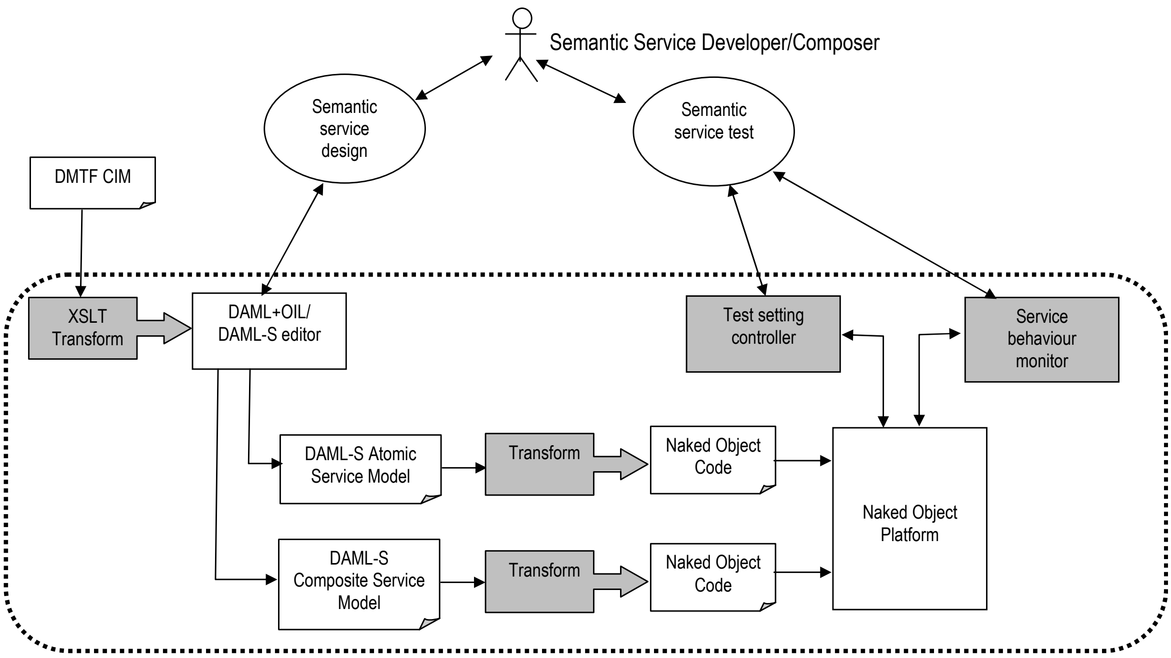
- The mapping from management information to ontologies.
- The mapping from DAML-S to and from Java object/interface specification.
- Refinement of test parameter control and service behaviour monitoring features.

4.4.3 Architecture and Design

The platform will be structure as outlined in the figure below and will consist of the following components:

- A Java-based rapid object prototyping framework based on the Naked Objects platform (Naked Objects)
- An XSLT transform for converting the CIM schema to an DAML+OIL or OWL ontology.
- A ontology editor suitable for editing DAML+OIL and DAML-S descriptions, e.g Protégé.
- A converter function for converting DAML-S atomic service descriptions into Java interface definitions and associated java object logic.
- A converter function for converting DAML-S composite service descriptions into Java interface definitions and associated java object logic.
- A facility for controlling test parameters.
- A facility for monitoring service behaviour.

The diagram shows the high level test cases the platform aims to support for the user, i.e. the semantic service developer/composer. The shaded elements are those that will be developed for the platform.



System Under Test: Semantic Service Development Platform

4.4.4 Evaluation Methodology

Results will be evaluated by assessing the usefulness of the tools to the semantic service develops in the project. This will be conducted mostly on an informal basis, with the aim being to incrementally improve the usefulness of the tool and in the process gain insights into the pros and cons of using DAML-S and existing management information models for semantic smart space service development.

4.4.5 Use within M-Zones Research Programme

This platform primarily aims to addresses engineering issues covered by Theme 3 – "Engineering Open Smart Spaces". In particular it covers issues of semantic service engineering raised in (O'Sullivan and Lewis 2003a). However, the platform also aims to allow specific semantic service descriptions to be developed and corresponding test stubs to be generated, supporting the need for example services descriptions as needed in for some experiments under Theme 1 – "Adaptive Services". This work builds on the state of the art analysis into service composition conducted for deliverable D1.1.

Depending of the success of the platform as a rapid prototyping tool, it may be used to develop deployable components that implement services for a future service testbed. This will involve generating components with open service interfaces such as WSDL or JXTA rather than test implementations of DAML-S services. Generating deployable service may also involve providing support for the integration of policy-based management for service components into the development process.

4.5 Adaptive Service Composition Engine Evaluation Testbed

4.5.1 Specification

A testbed in which services can be developed and run will be configured. The testbed will be completely simulated on a few network connected computers. In particular, two questions relevant (but not core) to my research will be examined. The first is examining how a service composition is described, monitored and executed. The second is how to provide hooks into an executing service composition, providing both feedback to and accepting input from a user. This will involve a considerable amount of integration with existing technology.

Once these infrastructural requirements have been resolved, the core questions of my research at this point need to be answered; to what extent can we expect a machine to decompose a task to a set of services without human input, and how best this input can be accepted, interpreted and stored.

4.5.2 Experimentation Methodology

Various methods to enable the resolution of inferred tasks down to functionality provided by individual services are examined according to their flexibility, robustness and quality. The suitable algorithm will gather a list of available services, described by their inputs, outputs and effects. It will also be supplied with a description of inputs, outputs and effects of the desired composed service. It will use a suitable searching algorithm (a number of tree searching techniques will be evaluated) to attempt to navigate from the supplied inputs through to the required outputs and effects.

4.5.3 Architecture and Design

The design consists of two main components, the available services and the composition engine. A more detailed description and diagram of the service composition can be found in (O'Donnell et al, 2003). Much of the design will become apparent during the course of the tests, indeed it is difficult to tie it down to any decision until the various options have been suitably evaluated. Integration with parallel research into quality of service and policy will also be examined.

4.5.4 Evaluation Methodology

Performance is not a core criteria here. Reliability, flexibility, and extensibility are the real focuses of the evaluation. Because these are not immediately measurable, but are rather deduced by the researcher from experience, it is difficult to provide any formula to define success or failure.

A suite of inferred tasks will be provided, along with a large pool of available services which between them, will be capable of completing the task. Each searching technique will be examined in terms of both the quality of the final output of the executed composition (is it what was originally desired) and the efficiency of the composition (it shouldn't create a composition which takes useless but non-damaging steps).

4.5.5 Use within M-Zones Research Programme

The platform will provide a dynamic method of software (and knowledge) reuse. In a smart space, software must be flexible to meet the user's needs, rather than expecting the user to bridge the gap between their understanding and the software's functionality. This platform will be included in Theme 1: Adapting to User Needs. It will build upon existing research into user-centric adaptive software and service composition.

4.6 Assurance Service Development Test-bed

4.6.1 Specification

The aim of this platform is to:

- Investigate the rapid prototyping of service composition;
- Investigate the development of a quality of service assurance mechanism for composite services;
- Investigate the integration of a policy-based management architecture (e.g. dtmf) and language (e.g. ponder) to the composite services;
- Investigation of Policy refinement mechanisms;
- Investigation of expressing the service's non-functional parameters;

4.6.2 Experimental Methodology

The methodology used will be a real world test-bed with the possibility of using simulation for some parts of the experiment.

The idea is to implement the design step by step starting from a system and adding 'features' one at the time. After adding each feature, the system will be tested to make sure it is working in accordance to spec.

4.6.3 Architecture and Design

For the moment, the architecture will have the following with its candidate technologies:

- Java based composite service using EJB or Jboss;
- Policy based management using DMTF model and Ponder language;
- Quality of Service parameters will be expressed in SLA or DAML-S;
- Component's parameters/interface will be expressed in DAML-S;

4.6.4 Evaluation Methodology

Results will be evaluated by appraising the validity and performance of each component built for the system. This approach allows us to note and take evasive action at an early stage if an unseen problem is encountered with the architecture while giving us a better understanding of the pros and cons of our architecture.

4.6.5 Use within M-Zones Research Programme

This platform aims to address the issues of management of composite services and the quality assurance for those services, both monitor and control.

4.7 Context-aware Development Testbed

4.7.1 Specification

This platform aims to:

- Investigate the nature of context information, particularly that information which should be made available to a Context Information System (CIS) to enable smart space applications.
- Investigate methods for interoperability between (and ideally automatic discovery and transformation of data from) sources of context information such as sensors.
- Extend the usefulness of context systems in different smart spaces by making context information available for transfer between them.
- Investigate the internal structure of a CIS, concentrating on forms of knowledge representation.
- Construct APIs useful for context-aware applications to make use of context information.

4.7.2 Experimentation Methodology

This testbed will assess the usefulness of a Context Information System (CIS) to a context-aware application. This experimentation will probably take some (not necessarily all) of the following forms:

- Computer simulations of smart space scenarios, modelling typical situations in software and feeding inputs from these situations into the CIS. These inputs will be transformed by the CIS into its own representation of context, and made available to context-aware applications through an API. The simulation would also contain models of these applications, and attempt to assess the quality of the information supplied to them.
- Supplying context information to (and receiving it from) other smart space entities such as service composition engines, policy servers and gesture recognition engines. These entities will be very important participators in a context system, and may find information supplied by the CIS interesting.
- Real-life application of a CIS could take place through the addition of context-awareness to existing applications, particularly those relevant to wireless clients such as e-mail access. The hardware used in such an experiment could be as simple as a wireless access point and a handful of PDAs. The challenge in such an experiment would be to obtain enough real-world data from the environment to construct a useful representation of context.

4.7.3 Architecture and Design

Initial architecture designs hope to have a distributed network of CIS servers, each taking inputs from various sources. These sources would include:

- Physical smart space sensors, measuring information such as temperature, humidity, user location, etc.
- Network elements such as routers, switches, etc. with management interfaces that expose data about their internal state.

- Smart space entities such as user familiars, service composition engines, gesture recognition, etc.
- Other CIS servers distributed around the network.

This information would be combined within the CIS to construct a knowledge representation that can be queried to make this knowledge available to context-aware clients, and also to other CIS servers. This representation has yet to be decided on, but should be flexible enough to support the addition of new forms of context as inputs to the system without being reworked.

An extensible API will also be constructed to allow consumers of context to query the constructed representation of context, and get meaningful answers within a reasonable period of time. Although this API would most likely have to be fixed at design time, it also should not restrict the addition of new forms of context to the system at a later stage, in a similar fashion to the context representation.

Security mechanisms must be put in place to prohibit unauthorised access to context information, possibly making use of work in the area of policy-based management and role-based access.

4.7.4 Evaluation Methodology

The evaluation criteria will most likely be application-specific, apart from some criteria that will be relevant to all applications, such as overall request time. Sets of criteria will be identified and documented for each of the modelled scenarios. With these criteria, it will be possible to construct software models of the applications in question and the devices that they will run on, and model their interaction with the CIS.

As well as hardware devices such as user workstations and PDAs, this process will model interaction with other smart space systems for the purposes of evaluation. It may also be possible to interface with these systems directly as they mature during the course of the project. Keeping constant contact with other participators in the M-Zones project will help to identify requirements that they have at an early stage. Thus, evaluation of the performance of the CIS will be an ongoing process, concentrated on in the WP4 stage of the project.

Experimentation into the performance of the CIS may be able to interoperate with the experimentation platforms of other smart space entities to sanity-check results from the CIS. The construction of a simple API for access to context information will be essential for this to be possible.

4.7.5 Use within M-Zones Research Programme

A Context Information System as outlined here could be made use of by almost any part of a smart space system, to varying degrees. Gathering requirements for context information from context-aware applications and smart space infrastructure components will help to ensure that the context information that is made available will be as useful as possible.

Thus, use of a CIS will most likely fit in with most parts of the M-Zones programme, with the exception of low-level hardware without the capability of interfacing with such a system.

5 References

Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H.H. (2002), 'Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services', Proceedings of the 18th International

Conference on Data Engineering (ICDE'02), ISBN: 0-7695-1531-2, March 2002, pp 297 – 308

Carey, K., Cullen, B., Lewis, D., Wade, V. (2003a), "Assurance Management in Smart Spaces", M-Zones draft whitepaper, May 2003

Feeney, K. (2003a), "Beyond the role model: Organisational modelling in policy based management systems", M-Zones draft whitepaper, May 2003

Naked Objects, www.nakedobjects.org

O'Donnell, T, Higels, S., Brady, A., Wade, W. (2003a), "Towards a Natural Interface to Adaptive Service Composition", M-Zones draft whitepaper, 29th April 2003

O'Sullivan, D., Lewis, D. (2003a), "Semantically Driven Service Interoperability for Pervasive Computing", M-Zones draft whitepaper, 25th April 2003

Power, R. (2003a), "Topic Maps for Context Management", M-Zones draft whitepaper, 29th April 2003