

**M-Zones Deliverable No. 1**  
**State of Art Surveys**  
**Release 2: May 2003<sup>1</sup>**

**Cork Institute of Technology**  
**Trinity College Dublin**  
**Waterford Institute of Technology**

**Editor: Declan O'Sullivan, Trinity College Dublin**

---

<sup>1</sup> Release One was delivered in March 2003

## TABLE OF CONTENTS

<i>Introduction</i>	2
<i>"Network Connectivity and Configuration" Theme</i>	3
<i>State of the Art: Admission Control and Mobility Management in Heterogeneous Wireless Networks</i>	4
<i>State of the Art: Ad Hoc Networking</i>	16
<i>"Managing a Smart Space" Theme</i>	27
<i>State of Art Survey: Intra-domain and Inter-domain Management of Smart Space Environments</i>	28
<i>Adaptive Technologies</i>	46
<i>State of the Art: Adaptive Hypermedia</i>	47
<i>State of the Art: Policy Techniques for Adaptive Management of Smart Spaces</i>	58
<i>Software Infrastructure</i>	67
<i>State of the Art: Context Management</i>	69
<i>State of the Art: Service Composition</i>	88
<i>State of the Art: Middleware in Smart Space Management</i>	98

## Introduction

*A Smart Space is a physical space rich in devices and services that is capable of interacting with people [aka users], the physical environment and services originated outside the Smart Space. The aim of the Smart Space is to orchestrate the use of integrated physical and computing environment to bring tangible benefits to people in support of their tasks.*

The above definition of a Smart Space represents a challenging vision for the M-Zones project. To realise this vision a number of key research challenges need to be met from a broad spectrum of areas (e.g. ad hoc networking, dynamic service composition etc.). In order to initiate our research, a number of key areas were chosen for review and a state of the art survey was undertaken for each area. These surveys have helped to inform and orient the research work, and are documented within this deliverable. The surveys have been grouped according to the research themes proposed by the project: "Network Connectivity and Configuration" and "Managing a Smart Space".

Network Connectivity and Configuration Theme surveys:

- Admission Control and Mobility Management in Heterogeneous Networks
- Ad hoc networking

Managing a Smart Space Theme surveys:

- Inter/Intra Domain Management
- Adaptive Hypermedia
- Policy Techniques for Adaptive Management
- Context Management
- Service Composition
- Middleware Infrastructure

Each survey provides an introduction, an overview of the problem domain, an analysis of the state of the art, research directions and details of references used.

## **"Network Connectivity and Configuration" Theme**

The M-Zones programme has identified wireless networking as the main focus for network connectivity within the research programme, because Smart Space services offered to users are accessed at the access network level and it is widely anticipated that this will be predominantly wireless in the future. From the perspective of wireless access, the view of the late 1980s and early 1990s was that the telecommunications research community would be able to develop a Universal Mobile Telecommunication System that would be able to provide a unified radio access systems catering for all service needs. This has not become a reality but instead, driven by market forces, a wide range of mobile and wireless communications technologies have emerged. This includes the Global Systems for Mobile Communications (GSM) and its evolution in form of the General Packet Radio Service (GRPS), and EGDE (Enhanced Data Rates for GSM Evolution). It includes Wide-Band CDMA and the 3G mobile network technology and in the local area, wireless LAN systems based on IEEE802.11 standards, ETSI HiperLAN, Bluetooth, HomeRF, Digital Enhanced Cordless Telecommunications (DECT) and other systems. All of these wireless network technologies use different methods to form networks ranging from the planned, cellular network approach to ad-hoc networking to a hybrid form provided by some WLANs standards such as IEEE802.11, HiperLAN, and DECT.

This fragmented wireless access networking landscape poses significant problems for the delivery of communication services to users in Smart Spaces, as the range of data rates provided by the individual systems ranges from a few tens of kilobytes to tens of megabytes in the case of IEEE802.11a/g and the mobility support ranges from wide area mobility in the case of GSM and UMTS to virtually no mobility in the case of Bluetooth.

In order to deliver services in a smart space environment, it is critical to manage network configuration and connectivity from the network and the terminal side in such a way that the optimum access technology in terms of delivering a particular service is chosen. This means from the network perspective the optimum configuration that achieves the agreed QoS, that achieves maximum capacity, or that achieves maximum revenue for the operator. From a user perspective this means context-awareness of access in terms of location, time of day, business/leisure situation to minimise cost, maximise bandwidth, always guarantee a particular QoS, etc. The first paper in this state of the art review addresses the issue on admission control and mobility management in such a heterogeneous access network environment. It reviews the current state of the art in handover control, network selection and access control, and mobility management protocols based on Mobile IP and cellular IP. It suggests routes for future research utilising management concepts such as policy-based management. The second paper reviews aspects of ad-hoc networking, in particular with respect to addressing in wireless ad-hoc networks but provides also a limited review of routing in ad-hoc networks. Addressing is a critical issue in any wireless networks in particular when users access resources through different radio access technologies as this may require the use of multiple addresses. Implications are in the area of security, routing, charging, and QoS delivery. The paper suggests future research directions for addressing in ad-hoc networks.

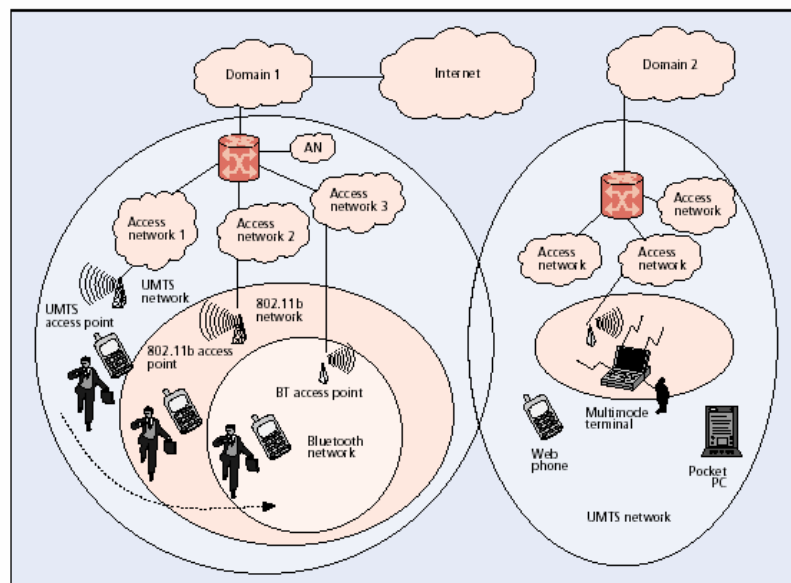
## **State of the Art: Admission Control and Mobility Management in Heterogeneous Wireless Networks**

Ken Murray, Dirk Pesch  
Adaptive Wireless Systems Group  
Cork Institute of Technology

### **1. Introduction**

Future mobile networks will not consist simply of one radio access technology such as WCDMA (wideband code division multiple access) or EDGE (enhanced data rate for GSM evolution), but will contain many different technologies. Seamless intersystem roaming across heterogeneous networks, as depicted in Figure 1, will be one of the main features in future generation mobile networks such as those envisaged in an integrated smart space environment. The motivation for heterogeneous networks arises from the fact that no one technology or service can provide ubiquitous coverage and continuous high QoS (quality of service) levels across multiple smart spaces. It will therefore be necessary for a mobile terminal to employ various points of attachment to maintain connectivity to a corresponding node at all times. Many wide area wireless network technologies are emerging. WCDMA is designed to meet the future requirements of third generation wireless communication services with data rates up to 2Mbps. UMTS (Universal Mobile Telecommunications System) will be based on this radio access technology. Both packet and circuit switched services can be freely mixed, with variable bandwidth and delivered simultaneously to the same user with specific quality levels. GPRS (General Packet Radio System) is a packet data service within GSM allowing bit rates from 9 to more than 150kbps. The user will be charged for the amount of data that is transferred and not for the connection time. Satellite networks promise global coverage and total ubiquitous computing but with lower QoS constraints than its cellular counterparts, while WLAN provides high-speed data service (up to 11Mb/s with 802.11b and 54Mb/s with 802.11a/g) over a geographically small area. All these network technologies differ in bandwidth, latency, power consumption and cost. Mobile terminals in a smart space environment will seamlessly roam between these access networks so as to maintain minimum QoS contracts for different applications and support user preferences. Supporting this seamless mobility is seen as one of the key issues in resource management for heterogeneous wireless networks (Shiao-Li Tsao, 2002). With intersystem mobility, users will benefit from the different coverage and capacity characteristics of each network throughout the interconnected smart spaces. Low tier WLAN offers high bandwidth but with less mobility than a UMTS network which offers medium bandwidth but with greater mobility. Services carried by future generation mobile networks will demand varying QoS for both real-time and non real-time applications. It is only with efficient mobility management and sophisticated admission control algorithms that these QoS constraints can be met and maintained (Nokia CRRM, White Paper, 2001). In a network with a mixture of resources (a heterogeneous network) offering a mixture of different services, it will be vital to provide the optimum radio bearer for each service.

This paper presents a state of the art survey in resource management in heterogeneous networks, in particular, emphasis is given to admission control and mobility management between different access schemes both of which are critical in a heterogeneous network environment as they balance the load across networks while maintaining sufficient QoS for varying traffic types.



**Figure 1. Seamless vertical handover in a 4G environment (A. Misra 2002)**

## 2. Overview

Giving access to users as they roam between networks or within a single network is governed by an admission control scheme. Admission control enables a wireless network to carry the largest amount of traffic for a given amount of spectrum. It ensures that the QoS perceived by each user is above the minimum guaranteed. Admission control and mobility management strategies facilitate load balancing between heterogeneous networks - users can be forced to handover to another network to make way for users with more demanding bandwidth requirements and can thus prioritise users. It may be possible using an admission control algorithm to admit a user to multiple networks simultaneously and use multiple connections to deliver services to the user and thus achieve a higher QoS than that offered from a single network. Much work has been done on call admission control in homogeneous networks such as UMTS. Before admitting a new user, admission control needs to check that the admittance will not sacrifice the planned coverage area or the quality of the existing connections. In UMTS, admission control accepts or rejects a request to establish a radio access bearer in the radio access network based on the interference levels, as the capacity in UMTS is interference limited. The introduction of a new user increases the interference level to existing users as all connections share the same radio channel. SIR (Signal to Interference Ratio) values are periodically measured to determine the level of interference at each base station. The SIR is the ratio of the received signal strength to the total interference from all interfering mobiles.

The admission control algorithm is executed when a bearer is set up or modified. The admission control functionality can have a decentralised approach where information regarding neighbour cell loads can be obtained from an entity controlling those cells. The admission control algorithm estimates the load increase that the establishment of the bearer would cause in the radio network. This has to be estimated for both the uplink and downlink for asymmetrical traffic. The requesting bearer gains access only if both the uplink and downlink admission control admit it, otherwise it is rejected because of the

excessive interference that it would produce in the network. Proposed admission control schemes for WCDMA cellular systems will be discussed in a later section.

The admission control procedure across heterogeneous networks is much more complicated than in a homogeneous network such as that described above. If multiple networks are available to a user at any one time, then choosing the most optimal network for a particular service delivery and choosing the correct time to execute a vertical handover<sup>1</sup> to improve the QoS for all users are important factors. A mobility management system can be used to control the migration of users from one system to another. The user, network or both can govern the mobility management and admission control procedures. Giving total control to the user can result in network instability as users compete for network resources, while a network controlled system will ignore user preferences and QoS requirements. The most optimal mobility management and admission control scheme should encompass both aspects – mobile assisted call admission control and mobility management.

The mobility management system will have many parameters with which to make a vertical handover decision, such parameters should include:

- Signal Strength Measurements
- Bit Error Rates
- Perceived QoS and the QoS requirements for the current application
- Network Coverage
- Cost
- Battery Power requirements to implement the handover – powering up new interface
- User Preference – user wants to be connected to the cheapest network available regardless of offered QoS

The above parameter values come from both the network and the user and thus emphasises the need for the mobile assisted approach. The mobility management algorithm will process this information and decide which is the most optimal network from those available and when is the correct time to initiate a handover. Schemes based on Policy (Helen J. Wang, 1999), Fuzzy Logic (P.M.L.Chan 2001a, P.M.L.Chan 2001b) and Neural Networks (J. Makela 2000) have been proposed and will be discussed in the next section.

The chosen network will admit the user if resources are available or may force a current user to handover to another network to make way for the incoming connection. Checking resource availability in GSM/GPRS is a straightforward procedure as network access is based on FDMA/TDMA. If the requested timeslots are available the service can be delivered at the requested data rate. However, capacity in UMTS is interference limited. UMTS utilises WCDMA on the air interface, users are separated using different spreading codes with all users sharing the same bandwidth. Admitting another user will increase the overall interference level for all users in the reference and neighbour cells and therefore reduces the average SIR at the receiving base station. Users requiring high data rates will demand a high SIR level. It is important therefore to accurately control access in a UMTS network. In a heterogeneous network environment, the mobility management system can move users from a UMTS network to make way for a high priority user in the event of high congestion. The next section will discuss state of the art proposals for mobility management and call admission control algorithms in heterogeneous networks using various techniques. The proposals are mainly concerned with handover

---

<sup>1</sup> Vertical handover is the common term given to handovers between access networks

initiation and network selection. The proposals do not address the availability and reservation of resources in the destination network or the network controlled movement of users between networks to ensure the availability of capacity.

### 3. Analysis

This section will detail state of the art proposals for mobility management and admission control schemes in heterogeneous networks. Mobility management is primarily concerned with handover initiation and network selection. Admission control is inherently part of the handover execution stage and governs access to the chosen network. The state of the art in each of these areas will now be presented.

#### 3.1 Handover Initiation

Handover Initiation is the process of monitoring the current network connection, recognising the need for handover and subsequently initiating it. The criteria used will reflect the condition of the current network connection such as signal strength, the coverage area, bit error rate and perceived QoS. As outlined in (Reynolds, 2000), there are three main reasons to initiate a handover.

To service a user request. For example, a user currently participating in a multimedia call may request their QoS to be modified in such a way that the serving cell, code, technology, network is unable to service the request.

To service a system request. For example, to maintain an existing call or to maintain network policy regarding user access rights, guaranteed QoS to users, optimal network selection to serve user request etc.

To service a service provider request. For example, a service provider's preferred network over which to deliver a service may be time dependent.

The handover initiation algorithm needs to process many parameters and decide whether a handover to another system is required. In (P.M.L.Chan, 2001a) a handover initiation algorithm using Fuzzy Logic concepts is presented. The algorithm is separated into three different stages. In the first stage, data from the system is fed into a fuzzifier, to be converted into fuzzy sets. A fuzzy set is a set without a crisp, clearly defined boundary and therefore, has a varying degree of membership. The system data comprises of values defining QoS perceived by the user, network coverage, bit error rate and average signal strength measurements which are mapped into a membership value of a fuzzy set. In the second stage, a set of IF-THEN fuzzy rules is applied to the system. These rules are conditional statements that specify how the fuzzy system works. The following example taken from (P.M.L.Chan, 2001a) illustrates the concept.

IF signal strength is strong, AND QoS is good, AND Bit Error Rate is medium, AND Network Coverage is medium, THEN handover = NO

Other possible outcomes are, Yes, Possibly Yes, and Possibly No. A table is generated that shows the outcomes for all possible values of the input criteria. Each outcome is assigned a membership value based on the input criteria. The final stage of the algorithm is the defuzzification process where all outputs are aggregated to produce a single number representing the handover factor. Based on the final handover factor, a handover will be initiated or not.

Deciding the correct time to initiate a handover is an important part of mobility management in a heterogeneous network. For example, users may want to minimize the cost of being connected to a cellular data service and therefore want to maintain WLAN connectivity as long as possible. In (J. Makela 2000), a predictive inter technology handover algorithm is presented. Neural network based pattern classification using signal strength measurements are used for path identification. As a user moves away from a WLAN access point, the system recognises the migration path from trained samples and therefore knows the most optimal handover initiation time. This classification type handover algorithm will need prior knowledge of the radio environment so that the neural networks can be trained before system deployment. The author's argue, however, that this training can be conducted gradually while the system is on line.

A technique for handover initiation using neural networks is also presented in (Kaveh Pahlavan, 2000). A three-layer back propagation neural network used for pattern recognition (S. Haykin, 1999) is trained using received signal strength measurements and locations where handoffs should be made. In this way, the system requires knowledge of the received signal strength patterns at such locations. A simulation scenario is presented using four identical base stations (BS) in a micro cellular environment and a mobile host (MH), which is moving from the neighbourhood of BS<sub>1</sub> toward BS<sub>2</sub> along a direct path. It is assumed that all BSs can provide the same service to the MH. The neural network takes a number of power samples from each BS and using pattern recognition, selects the BS, which is most suitable, while minimizing handoff delay and ping-pong<sup>2</sup> effect. The output of the system is a control signal that is zero as long as the MH is closer to BS<sub>1</sub> and one whenever the MH is closer to BS<sub>2</sub>. As in (J. Makela 2000), the system requires prior knowledge of the radio environment and needs much configuration before deployment.

### **3.2 Network Selection**

The aim of the network selection stage is to select a network connection that can satisfy the requirements of the network provider and also the user, such as low cost, good signal strength, optimal bandwidth, low network latency and high reliability. In (P.M.L.Chan, 2001b) a fuzzy logic multiple objective decision making algorithm is presented. There are two stages to the handover decision. In the first stage a fuzzy ranking procedure is used to compare the performance of the different networks for a particular handover decision criterion. For example, when cost is considered, the network with the lowest cost will have the highest ranking for the cost criteria. The weighting of the criteria is then developed by obtaining a ratio scale for the criteria based upon a paired comparison of each criterion. In the second stage of the algorithm these weightings are applied to each criterion and the highest criterion value selected.

The chosen network will be the one corresponding to the highest ranking for that criterion. To evaluate the performance of the algorithm, three access segments are considered, GPRS, UMTS and a satellite system.

A policy controlled handover scheme between heterogeneous networks is presented in (Helen J. Wang, 1999). This scheme allows users to express policies on what is the best wireless system at any moment and makes tradeoffs among network characteristics and dynamics such as cost, performance, and power consumption. A network condition estimation scheme is used that periodically reports the available bandwidth at each network to the mobile host. A performance agent collects this information on current bandwidth usage at base stations and announces this information to its coverage area. Based on this

---

<sup>2</sup> Excessive handovers between two base stations is commonly referred to as the ping-pong effect

information, policies are designed that achieve load balancing across different networks. The policy approach presented in this work uses a cost function to evaluate the performance offered by each network. The cost of using a network at a certain time is a function of several parameters, the bandwidth it can offer,  $B$ , the power consumption of using the network access device,  $P$  and the cost  $C$  of each network. Each parameter has an associated weight factor. The bandwidth parameter estimates the current network condition, while power consumption and cost are fixed budgets whose weightings must change to reflect battery power available and how much money the user has currently spent. If a user wants to be connected to the cheapest network at all times, then  $w_c$  should be set to 1 and 0 for the other weight values. When comparing two networks, their cost functions are evaluated and the one with the lowest cost value is selected. Periodically, the system re-calculates the cost function of each reachable network based on up to date parameters. A stability period is included to ensure a handoff is worthwhile for each mobile. The stability period is a function of  $T_{\text{makeup}}$ , which is defined as the time needed to make up the loss of money or data depending on the current policy due to handoff latency. If a user is likely to be in the range of a better network for  $T_{\text{makeup}} + \text{handoff latency}$ , then it is worthwhile to handoff. Experimentation is carried out with four networks: IBM Infrared LAN, Lucent WaveLAN, the Metricom Ricochet network and GSM.

### 3.3 Handover Execution

The objective of the handover execution stage is to change cell, code technology, or network conforming to the details resolved during the decision phase. As outlined in (Reynolds, 2000), handover execution across heterogeneous networks involves the following issues.

1. Connection Changes. Changing radio links often means changing radio access nodes. New connections need to be set up and superfluous connections released.
2. Switching and bridging. If data is to be transmitted on two connections (packet duplication) or if data coming on two connections is to be combined on one connection, a bridge connection is required. Bridge connections are used, for instance, to prevent loss of data. Once this feature is no longer needed the bridge is released.
3. Combining and multicasting. In the case where macro diversity is supported by the layer network involved, connections are added to and released from multi-casting and combining points. In this case, adding a connection does not imply releasing another.
4. Re-routing. Handover may imply re-routing of connections through the fixed network, even outside the access network that is currently involved.
5. Control point transfer. If a user moves from one domain to another, it could be required to transfer control.
6. Security functions. Handover often requires the transfer of security keys and authentication.

To support seamless mobility in a heterogeneous environment, the mobility management entity will be required to locate roaming terminals for call delivery and maintain connections with terminals that change their point of attachment. Fast handoff with minimal packet loss is an important characteristic in any mobility management architecture.

IP will be the common network layer protocol in future mobile networks as outlined by A. Sanmateu et al (2002):

“... IP technology has emerged as a natural means of initiating network convergence and the all IP paradigm has become the implicit assumption for most studies on the next generation architecture design”.

With this view of an all IP core network, Mobile IP is the common approach for macro mobility management between access segments (Shiao-Li Tsao 2002, A. Sanmateu 2002). Cellular IP and hierarchical network structure using Mobile IP with packet buffering and redirection have been proposed for micro mobility within an access network (A. Misra 2002, Andras G. Valko 1999, Jon Chiung Shien Wu 2001, Chen Lin Tan 1999).

### 3.3.1 Macro Mobility

The movement of a mobile host between access networks is known as macro mobility. Mobile IP has been widely proposed in the implementation of macro mobility. In an IP network, the IP address is usually representative of the location of the IP node. As the mobile node changes its point of attachment, the association between the mobile's identity and location is lost. Mobile IP solves this problem by registering the address of a foreign agent (FA) in which the mobile node is roaming at the home agent (HA) in the mobile's home network. The address of the FA is known as the Care of Address (CoA). Any packets received at the HA addressed for the mobile node are “tunnelled” to the FA and delivered to the mobile node. As mobiles move to a new area covered by a new FA, the new CoA has to be re-registered at the HA. Figure 2 shows the message exchange for UMTS/WLAN handover using Mobile IP. Mobile IP introduces latency in the handover procedure and is not appropriate to handle seamless handoffs for real time traffic. To reduce the handover latency, an enhancement to Mobile IP is proposed, in (A. Sanmateu 2002). In this the radio channel signal quality is used to predict a handover, therefore the HA registration procedure can take place prior to the handover and thus reduce handover latency. Soft handover is used to maintain a seamless handover between segments.

In (Shiao-Li Tsao 2002), a gateway approach to macro mobility management between WLAN and UMTS is presented. In this architecture a new logical node connects the two networks together. For inter-working services control and data packets are routed through the gateway. The two networks operate independently and Mobile IP is not necessary. An emulator approach is also presented where the WLAN is used as an access point in the UMTS network. In this sense the WLAN access point emulates a UMTS base station (node-B) or UMTS RNC (Radio Network Controller / packet switching entity in UMTS network).

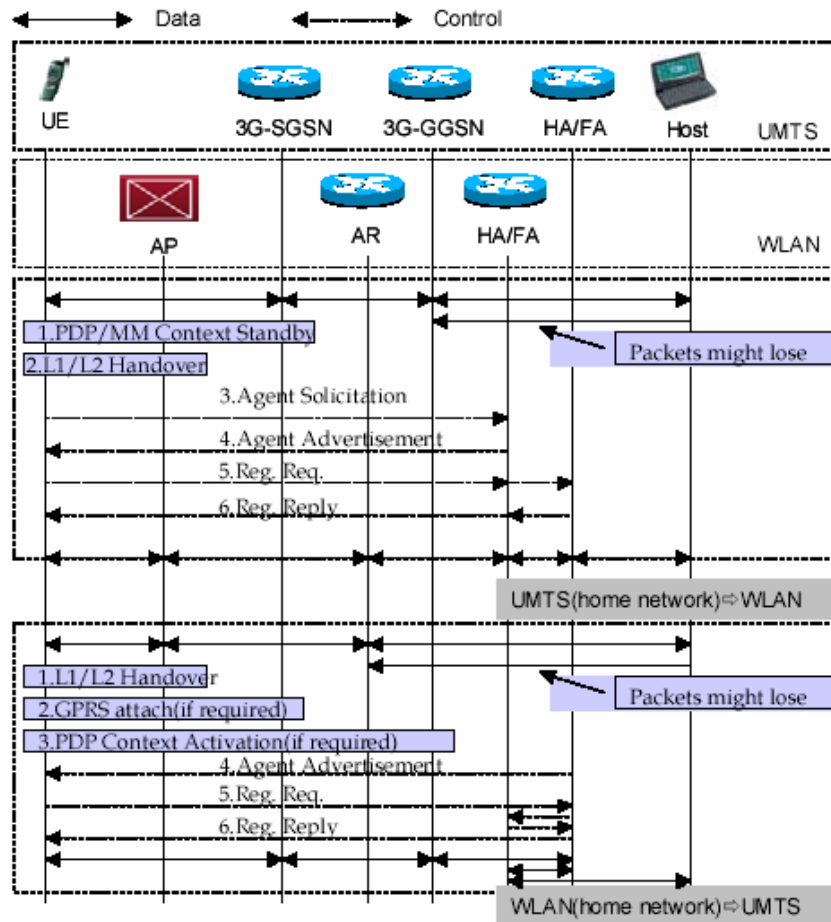


Figure 2. UMTS/WLAN Handover using Mobile IP (S. Li Tsao 2002)

### 3.3.2 Micro Mobility

Movement between points of attachment within a network is known as micro mobility. Fast handover is critical for micro mobility as the handoff rate is very much higher than that with macro mobility and large-scale packet loss and latency are unacceptable.

In (A. Misra 2002), the Intra Domain Mobility Management Protocol (IDMP) is introduced. In the IDMP fast handoff scheme, a mobility agent is situated between the core network and the subnet agents. Before a mobile moves to a new subnet agent, it sends a move imminent message to the mobility agent. The mobility agent then broadcasts all subsequent packets to each subnet agent where they are buffered. When the mobile arrives at the new subnet agent the buffered packets are delivered to the mobile. This can happen before the registration update thus reducing the handover latency and lost packets during handover.

Cellular IP is presented in (Andras G. Valko 1999) as an approach to reduce handover latency and improve the location management for roaming mobile terminals. Cellular IP maintains distributed cache for location management and routing purposes. Distributed paging cache coarsely maintains the position of 'idle' mobile hosts in a service area. Cellular IP uses this paging cache to quickly and efficiently pinpoint idle mobile hosts that wish to engage in active communications. This approach is

beneficial because it can accommodate a large number of users attached to the network without overloading the location management system. Distributed routing cache maintains the position of active mobile hosts in the service area and dynamically refreshes the routing state in response to the handoff of active mobile hosts. The authors argue that by separating local and wide area mobility, the performance of existing mobile host protocols (e.g. Mobile IP) can be significantly improved.

An intelligent handover scheme is presented in (Jon Chiung Shien Wu 2001). The architecture builds on the Mobile IP concept by enabling interaction between layer 2 and layer 3 during the handover process. The solution consists of three extensions.

#### **Packet Buffering**

In this scheme, the old FA buffers received packets for a mobile host when it hands over to a new FA. When the handover is complete the old FA forwards the packets to the new FA, thereby reducing the lost packet rate.

#### **Neighbour List Update message**

Here the mobile host has access to a neighbour list of candidates the mobile host can roam into from its current location. This enables quicker handoff to a neighbouring network and quicker registration at layer 3.

#### **Layer 2-handoff notification to layer 3**

This extension enables layer 3 to know about a handover occurring at layer 2 and doesn't have to rely on timer expiration.

A hierarchy of mobility agents is presented in (Chen Lin Tan 1999) based on the Mobile IP concept to restrict the handoff processing overheads within the vicinity of the mobile node, and uses multicast as the packet forwarding mechanism to deliver packets to multiple base stations within the vicinity of the mobile node to achieve fast handoff performance. Mobiles roaming within a network register their new point of attachment with a DFA (Domain Foreign Agent) within the network. This DFA maintains the bindings for each mobile node within the network. It is only when the mobile moves to a new network (inter network mobility), the home agent has to be informed. To reduce the packet loss when moving between access points, each candidate access point receives the packets for the mobile node and begins to buffer them. When the mobile completes the handover, the packets can be sent from the buffer to the mobile. The buffers in the other access points are then cleared.

### **3.3.3 Admission Control**

Much research has been done on admission control schemes in homogeneous networks with overlaid micro/macro cell structures and interference limited cellular networks such as WCDMA UMTS. Admission control schemes across heterogeneous networks based on radio channel characteristics, resource availability, QoS constraints and user policy still remains an open issue. This section will therefore focus on admission control schemes proposed for a single network, which may be incorporated into an admission control algorithm for heterogeneous networks.

An admission control policy (ACP) for a two tier cellular system with multiple types of service requests is presented in (F. Santucci, 2000). Each service request has different bandwidth requirements and priority. Two types of ACP are investigated – non prioritised and prioritised ACP. In the non-prioritised ACP scheme, all types of calls are treated equally, the scheme is based on a first come first served rule. A call is denied access only if no channel is available. In the prioritised ACP scheme, the policy is based on dropping lower bandwidth calls to serve handover or new call requests of a larger

bandwidth class, thus higher bandwidth calls have a higher priority. This prioritising policy could be useful in a heterogeneous environment where connections with a low bandwidth requirement can be reallocated to a network optimised for that particular data rate and service provisioning and thus leave high speed connections free for users requiring high QoS.

Much literature has been dedicated to Signal to Interference ratio (SIR) based call admission control algorithms for CDMA cellular systems. In (Zhao Liu, 1994), the concept of residual capacity is introduced as the additional number of calls a base station can accept such that the system wide outage probability will be guaranteed to remain below a certain level. The residual capacity is dynamically updated at each cell according to the reverse link SIR measurements at each base station. The residual capacity at each base station is defined as

$$R_k = \begin{cases} \min R_k^{(j)} & \text{if } \min R_k^{(j)} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where  $R_k^{(j)}$  is given by

$$R_k^{(j)} = \begin{cases} \left\lfloor \frac{1}{SIR_{th}} - \frac{1}{SIR_k} \right\rfloor & \text{for } j = k \\ \left\lfloor \frac{1}{\beta} \left( \frac{1}{SIR_{th}} - \frac{1}{SIR_j} \right) \right\rfloor & \text{for } j \neq k \end{cases}$$

$SIR_{th}$  and  $SIR_k$  are the SIR threshold at the BS input receiver and the reverse link SIR for BS  $k$  respectively.  $\beta$  denotes the estimate of the interference coupling between adjacent cells. For each call request at cell  $k$ , the BS checks the value of the residual capacity,  $R_k$ , if  $R_k > 0$ , the new call is accepted.

It is envisaged that this SIR based admission control policy could be part of the initial stage of network access in a UMTS network. Other parameters such as required QoS, user preference, other current connections etc. will have to be included in the network access decision process.

A much different approach to admission control in WCDMA using neural networks and fuzzy logic is presented in (Chung-Ju Chang, 2000). An intelligent call admission controller adopts fuzzy and neural network techniques to make admission decision for a new call request by considering the QoS measures of all service types, predicted next step call interference and estimated new call interference. The fuzzy interference estimator estimates the equivalent interference,  $C$  of a new call from its claimed traffic parameters: peak rate  $R_p$ , mean rate  $R_m$ , peak rate duration  $T_p$  and its outage probability requirement  $P_{otg}$ . A set of fuzzy rules is used to obtain the interference estimate based on the new call parameters. A pipeline recurrent neural network (PRNN) is used for interference prediction. The PRNN takes the mean interference at the present time instant  $n$ ,  $\bar{I}_k(n)$ , as an input variable to predict the interference mean at the next time instant  $(n+1)$ ,  $\bar{I}_k(n+1)$ . A fuzzy call admission processor takes  $\bar{I}_k(n+1)$ ,  $C$  and the system outage probabilities of voice and data services denoted  $P_{otg1}(n)$  and  $P_{otg2}(n)$  as input variables to determine the acceptance of a new call request. The smaller the value of  $C$ ,  $\bar{I}_k(n+1)$ ,  $P_{otg1}(n)$  and  $P_{otg2}(n)$  the higher is the probability that a new call will be accepted.

The call admission control schemes presented herein are designed for a single network. Call admission in a heterogeneous network environment remains an open issue. Such schemes should control the access of users between available networks based on current loads, predicted traffic profiles (Ken Murray, 2001) and the optimality of a network connection to a particular service request.

#### 4. Research Directions

This paper has presented a state of the art survey in mobility management and admission control for heterogeneous wireless networks. Many proposals exist for handover initiation, network selection and handover execution. Admission Control Policies between heterogeneous networks however remains an open issue. Adopting robust and adaptive admission control algorithms will play an important role in the mobility of users through a smart space environment where services can be offered over multiple access networks, which offer varying QoS levels, coverage and capacity.

Future direction will be toward the development of adaptive admission control algorithms for wireless heterogeneous networks. The algorithms must consider user preferences, network load conditions – both current and predicted, the QoS requirement of running applications and the priority of users within the heterogeneous network. The main goal of the algorithm is to create a system that can dynamically adjust the load of users between the available access networks so as to maintain satisfactory QoS levels in a proactive manner. It is envisaged that much of work presented in this paper on handover initiation and network selection can be incorporated into a heterogeneous admission control algorithm.

#### 5. References

- A. Sanmateu (2002), “Seamless mobility across IP networks using mobile IP”, *Computer Networks* 40, 2002, Available: [www.elsevier.com/locate/comnet](http://www.elsevier.com/locate/comnet) (Accessed: 2002, Dec)
- A. Misra (2002), “IDMP – Based Fast Handoffs and Paging in IP-Based 4G Mobile Networks”, *IEEE Communications Magazine*, March 2002
- Andras G. Valko (1999), “Cellular IP: A New Approach to Internet Host Mobility”, *Ericsson Research*, Available: [www.comet.ctr.columbia.edu/cellularip/pub/ccr99.pdf](http://www.comet.ctr.columbia.edu/cellularip/pub/ccr99.pdf) (Accessed: 2002, Dec)
- Chen Lin Tan (1999), “A Fast Handoff Scheme for Wireless Networks”, *WoWMoM '99*, Seattle, WA
- Chung-Ju Chang (2000), “Intelligent Call Admission Control for Differentiated QoS Provisioning in Wideband CDMA cellular Systems”, *IEEE Vehicular Technology Conference*, FALL 2000
- F. Santucci (2000), “Admission Control in Wireless Systems with Heterogeneous Traffic and Overlaid Cell Structure”, *IEEE Vehicular Technology Conference*, FALL 2000
- Helen J. Wang (1999), “Policy – Enabled Handoffs Across Heterogeneous Wireless Networks”, *WMCSA '99*, New Orleans, Louisiana

J. Makela (2000), "Handoff Decision in Multi-Service Networks", *PIMRC 2000*, Sep 2000, London, UK

Jon Chiung Shien Wu (2001), "Intelligent Handoff for Mobile Wireless Internet", *Mobile Networks and Applications*, Vol. 6, 2001, Available: [www.elsevier.com/locate/comnet](http://www.elsevier.com/locate/comnet) (Accessed: 2002, Dec)

Kaveh Pahlavan (2000), "Handoff in Hybrid Mobile Data Networks", *IEEE Personal Communications Magazine*, April 2000

Ken Murray (2001), "Neural Network based Adaptive Radio Resource Management for GSM and IS136 Evolution", *IEEE Vehicular Technology Conference*, FALL 2001

Nokia Common Radio Resource Management (2001), White Paper, Available: [http://www.nokia.com/networks/systems\\_and\\_solutions/files/white\\_papers/3g\\_wp\\_allip\\_crmm.pdf](http://www.nokia.com/networks/systems_and_solutions/files/white_papers/3g_wp_allip_crmm.pdf) (Accessed: 2002, Nov)

P.M.L.Chan (2001a), "An intelligent Handover Strategy for a Multi-Segment Broadband Network", *PIMRC 2001*, Sep 2001, San Diego, California

P.M.L.Chan (2001b), "Mobility Management Incorporating Fuzzy Logic for a Heterogeneous IP Environment", *IEEE Communications Magazine*, Dec 2001

Reynolds (2000), "Mobility Management for the Support of Handover within a Heterogeneous Mobile Environment", *3G Mobile Communications Technologies Conference*, 2000

S. Haykin (1999), "Neural Networks, A Comprehensive Foundation", 2<sup>nd</sup> edition, Prentic Hall.

Shiao-Li Tsao (2002), "Design and Evaluation of UMTS-WLAN Interworking Strategies", *IEEE Vehicular Technology Conference*, FALL 2002

Zhao Liu (1994), "SIR-Based Call Admission Control for DS-CDMA Cellular Systems", *IEEE Journal on Selected Areas in Communications*, May 1994

## State of the Art: Ad Hoc Networking

John Paul O Grady/ Aidan McDonald  
Adaptive Wireless Systems Group  
Cork Institute of Technology

### 1 Introduction

Ad Hoc networks are multi-hop wireless networks where nodes may be mobile. These types of networks are used in situations where temporary network connectivity is needed. Ad hoc networks are formed on a dynamic basis, i.e. a number of users may wish to exchange information and services between each other on an ad hoc basis, in order to do this they will need to form an Ad Hoc network. An example of this may be found in a disaster relief situation. Here an Ad Hoc network could enable emergency services to co-ordinate emergency services more effectively or enable medics in the field to retrieve patient history from hospital databases (assuming that one or more of the nodes in the Ad Hoc network has connectivity to the Internet).

Smart spaces are defined as environments that allow people to perform tasks efficiently by offering unprecedented levels of access to information and assistance from computers. Ad Hoc networks will play a significant part in these environments, allowing people to exchange information and services; for example, people at a meeting could create an Ad Hoc network using their PDA's or Laptops and exchange information relevant to the meeting. Indeed there are endless examples of where their use could be found.

### 2 Overview

The two areas of Ad Hoc networking that we mainly researched were IP routing and dynamic configuration of IP addresses in Ad Hoc networks. Routing in Ad Hoc environment is different compared to normal wired networks. This is mainly due to two factors

- The bandwidth restriction due to the use of wireless connections
- Rapid change in network topology due to node movements

In our analysis of IP routing we present the advantages and disadvantages of the various types of IP routing protocols proposed for Ad Hoc networks.

Most of the research conducted in the area of Ad Hoc networking to date has focused on solving the routing problem, however the problem of dynamic configuration of IP addresses has yet to be addressed.

Two methods can be used to configure an IP addresses, manual configuration and dynamic configuration. Since an Ad Hoc network is highly dynamic by nature, manual configuration can't be used, as it would take away from the dynamic nature of the network. This means that dynamic configuration is required; in wired networking dynamic configuration is achieved using Dynamic Host Configuration Protocol (DHCP). DHCP however can't be used in Ad Hoc networks, in our analysis of

the area we present the reasons why as well as investigating the various protocols proposed for dynamic configuration of IP addresses in Ad Hoc networks.

## 3 Analysis

### 3.1 IP Routing in Ad Hoc Networks

An Ad Hoc network (also called a Mobile Ad Hoc Network MANET) consists of wireless hosts that move around, i.e. they have no permanent physical location. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes before the exchange of IP data packets. Below is a brief overview of IP routing in an Ad Hoc environment. There are many papers dealing with routing in Ad Hoc networks. If a more detailed review is required we have a summary of Ad Hoc routing protocols written by Susan Rea who is a PhD candidate student and a member of the Adaptive Wireless Systems Research Group at CIT.

The routing protocols in Ad Hoc wireless networks are generally categorised as

#### 3.1.1 Proactive

These protocols require each node to maintain one or more tables to store up to date routing information and to propagate updates throughout the network. These protocols try and maintain valid routes to all communication mobile nodes all the time, which means before a route is actually needed. Periodic route updates are exchanged in order to synchronise the tables.

Some examples of table driven ad hoc routing protocols include Dynamic Destination Sequenced Distance-Vector Routing Protocol (DSDV) [1], Optimized Link State Routing Protocol (OLSR) [2] and Fisheye State Routing Protocol (FSR) [3]. These protocols differ in the number of routing related tables and how changes are broadcasted in the network structure.

The problem with these protocols is the overhead; the protocols propagate and maintain routing information, regardless of whether or not it is needed.

#### 3.1.2 Reactive

These protocols create routes only when desired by a source node, therefore a route discovery process is required within the network. Once a route has been established, it is maintained by a route maintenance procedure until either the destination becomes inaccessible or until the route isn't needed any longer.

Some examples of source initiated ad hoc routing protocols include the Dynamic Source Routing Protocol (DSR) [4], Ad Hoc On Demand Distance Vector Routing Protocol (AODV) [5], and Temporally-Ordered Routing Algorithm (TORA) [6]. No periodic updates are required for these protocols but routing information is only available when needed.

#### 3.1.3 Hybrid

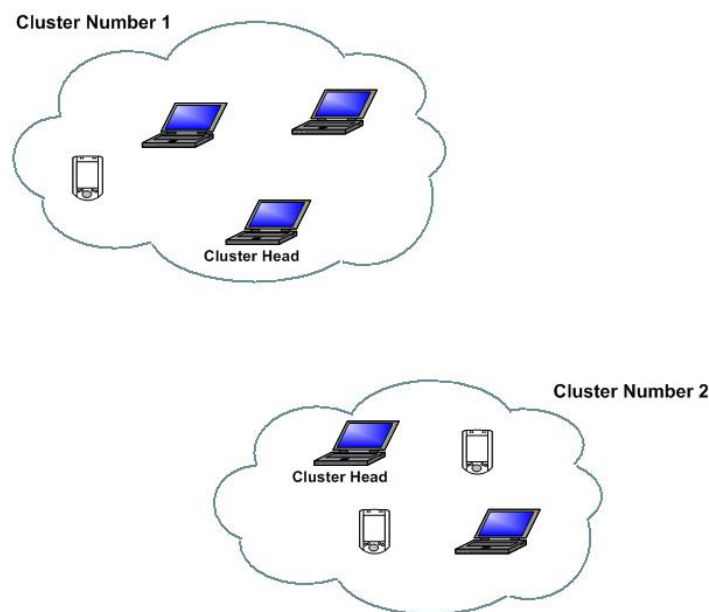
These protocols try to incorporate various aspects of proactive and reactive routing protocols. They are generally used to provide hierarchical routing; routing in general can be either flat or hierarchical

In a flat approach, the nodes communicate directly with each other. The problem with this is that it does not scale well, it also does not allow for route aggregation of updates

In a hierarchical approach, the nodes are grouped into clusters, within each cluster there is a cluster head, this acts as a gateway to other clusters, it serves as a sort of default route.

The advantage of a hierarchical structure is that within a cluster, an on demand routing protocol could be used which is more efficient in small-scale networks. For inter cluster communication then a table driven protocol could be used which, would allow the network to scale better. An example of such a hybrid routing protocol is the Zone Routing Protocol (ZRP) [7].

In figure 1.0 below we have two clusters, with one laptop representing the cluster head in each cluster.



**Figure 1.0**

### 3.1.4 Other Types of Routing Protocols

There are many other types of ad hoc routing protocols; one for example LANMAR [8] uses location info, obtained using the Global Positioning System (GPS). By knowing the precise location of a node you can limit the search to a smaller “request zone” of the network.

## 3.2 *Dynamic Configuration of IP Addresses*

In order to communicate nodes need IP addresses. Since Ad Hoc networks lack any centralised administration these addresses can't be manually configured, and so must be configured dynamically. In a wired network dynamic configuration is achieved using the Dynamic Host Configuration Protocol (DHCP) [9], however this requires the presence of a centralised DHCP server, which maintains the configuration information of all nodes in the network. Since an Ad Hoc network is devoid of any fixed infrastructure such as a central server, this approach can't be used.

There are a number of issues to consider:

- As regards the auto-configuration, the process is simpler in IPv6 compared to IPv4. It is easier to create globally unique addresses in IPv6; because IPv6 addresses are 128-bit addresses compared to IPv4's 32-bit. The MAC (48-bit) address or some other unique identifier for example could be embedded into the address to ensure the address is unique.
- In IPv6 there is also a process called [10] "IPv6 Stateless Address Auto Configuration" this is a process used to create unique link local addresses in the absence of a DHCP server. This process has to be modified for use in ad hoc networks, since ad hoc addresses need network wide uniqueness. A number of papers "Wireless Multi-hop Internet access: Gateway Discovery, Routing, and Addressing" [11], "IPv6 Auto Configuration in Large Scale Mobile Ad Hoc Networks" [12], "Stateless Address Auto Configuration in Mobile Ad hoc Networks using Site local Addresses" [13] propose a modification of the above technique as a method for dynamic IP address allocation. The problem with these solutions is that they are only applicable to IPv6 and so can't be used in IPv4; this in turn limits the use of these proposals.
- The use of Mobile IP "IP Mobility Support for IPv4" [14], "Mobility Support in IPv6" [15] is also proposed for use in ad hoc networks. In Mobile IP a node has two addresses, its home address and its care of address, when a node is away from its "home address" it obtains a "care of address" and registers that address with its home address. When packets are sent to the nodes home address they are forwarded to its "care of address". A foreign agent is used to assign the care of addresses.
- In order to connect to the Internet, there needs to be a gateway node that has access to the Internet. The other nodes need to discover this node if they want to access the Internet. There are a number of possible ways to discover the gateway, a multicast address could be used, or the gateway could advertise itself. This process could be done proactively or reactively depending on the routing protocol in use. One proposed solution to this problem is outlined in [16] "Global Connectivity for IPv6 Mobile Ad Hoc Networks". Two methods for Internet gateway discovery are proposed here, one method periodically disseminates gateway advertisements to all nodes in the MANET (Mobile Ad Hoc Network); the other utilizes solicitation and advertisement signalling between the MANET node and the gateway. This solution however is only applicable to IPv6, as it utilizes the [10] "IPv6 Stateless Address Auto Configuration" process.

The problem in regard to Internet connectivity for IPv4, is that most of the proposals for dynamic address allocation assumes the use of private addresses, due to the difficulty in obtaining global addresses. There is an issue here in regard to connectivity to the Internet, as some sort of network address translation (NAT) process will be required, this is a process that converts a private address into a unique global address. In the wired environment NAT is achieved using "Traditional Network Translation (NAT)" [17], this is a process that converts a non-unique IP address to a unique IP address. A solution is proposed in "Connectivity for IPv4 Mobile Ad Hoc Networks" [18] however it assumes that each node in the MANET (Mobile Ad Hoc Network) is using Mobile IP and as a result already has a globally unique home address. It assumes that a Foreign Agent assigns "care of addresses" to nodes as they arrive into the network and acts as a gateway for them when they want to connect to the Internet.

- When a node leaves the network, it needs to give up its allocated IP address. A node may leave the network gracefully or it may leave very suddenly, the nodes must recognise that a node has left the network and recover its IP address. Different proposals use different techniques to recognise that a node has left the network. We will outline these techniques in detail later.
- Another problem is the merger of networks. Say you have two networks, both using the same private address (e.g. 10.0.0.0). If these two networks merge, a process must detect this, as there is the possibility that a node in each of the two networks may share the same IP address. If this problem is not addressed routings errors can occur.

If two networks merge, each with different network addresses (e.g. 10.0.0.0 and 157.190.0.0), there will not be a problem in regard to having duplicate addresses. This is because different address spaces are used in each network, and so normal Ad Hoc routing will allow the nodes to communicate with each other. However still the merger of the networks may need to be identified, for example in-service discovery it may be beneficial to know that another network is nearby.

The partition is different, here the network may splits in two, and the allocation scheme must cope with this. It must for example recover any IP addresses that are not being used currently in the network.

### 3.2.1 Dynamic Allocation Proposals

When a node is assigned an address, it needs to find out whether or not the address it chooses is unique within its network. In order to determine this, a process known as duplicate address detection (DAD) is performed, it is basically a process that determines whether or not a chosen IP address is unique within a chosen network. An overview of different DAD techniques is presented in [19]. Generally the different proposals differ in the technique they use to perform DAD. The proposals fall into two different categories.

#### ***Hierarchical Approach***

In a hierarchical approach, a clustering technique is used with one node (cluster head) assuming the responsibility for the allocation of addresses to new nodes as they arrive, basically when a new node arrives, he registers with the cluster head, who then allocates a new addresses and coordinates a duplicate address (DAD) process in order to determine whether or not the address chosen is unique within the ad hoc network. The following paper is based on this approach

- Sanet Nesargi, Ravi Prakash. 2002 “MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network” in Proceedings of Infocom 2002. [20]

This paper presents a hierarchical approach to the IP address allocation problem. This paper proposes the use of IPv4 private addresses although IPv6 address could also be used. It does however only consider a stand alone MANET (Mobile Ad Hoc Network) which has no connection to the Internet. Below is an outline of the basic idea.

A new node entering the network, hereafter called the “requester”, chooses a reachable node as the “initiator”, which performs address allocation on its behalf. All other nodes know the route

to the initiator and can forward their responses to it. The initiator chooses an address it perceives as unallocated and attempts to acquire permission from all other nodes in the network to assign the address to the requester. Nodes perceiving this address as unallocated mark the requested address as allocation in progress and reply in affirmative to the initiator. This allocation is made permanent by a second message, which is sent by the initiator if the initiator receives an affirmative response from all nodes in the network. Therefore the IP address allocation is similar to a two-phase commit.

Nodes, which no longer wish to be part of the system, relinquish their address by broadcasting a message to the effect before leaving the network. If a node abruptly leaves the network, i.e. goes down without relinquishing its address, it would fail to respond to the address allocation request by some initiator the next time a requester enters the network. In this case the address of the departed node is cleaned up by the initiator awaiting a reply from the departed node.

This paper also deals with network partitions and mergers. Partitioning of the network is easy to handle, all the nodes in one partition conclude that all the nodes in other partitions have departed abruptly and so reclaim their addresses.

Merging works differently, here a unique partition ID is used. When two nodes come into contact and exchange their partition ID's, they realise that they are different partitions and so merge. To do this they exchange partition ID's as well as the table, which shows all IP addresses that have been allocated. If it is found that nodes in the partitions have the same IP address, then one node gives up its IP address and requests another one.

One issue with this paper is that networks merge even if they have no reason to do so. Also the problem of security has not been addressed.

### ***Flat Topology Approach***

In a flat topology approach there is no cluster head, which assumes responsibility for the allocation. Here when a node joins the network and wants an IP address, it chooses an address at random and then performs a duplicate address procedure in order to determine whether or not that address is unique. The papers below present solutions based on this approach

- Perkins et al. Nov 2001. Internet Draft "IP Address Auto-configuration for Ad Hoc Networks, Technical Report. Internet Engineering Task Force, MANET Working Group. [21]

This is the proposal from the IETF MANET (Mobile Ad Hoc Network) working group. The process differs slightly depending on whether you are using IPv4 or IPv6

The process is based on a proactive routing protocol e.g. Ad Hoc On Demand Distance Vector Routing (AODV) [5] and uses a flat structure. In a proactive routing protocol routing is done on an on demand basis, in order to route to a destination a AREQ message is sent out looking for the destination. When the destination is found an AREP message is sent back indicating that the destination is reachable.

A node performing the auto-configuration process picks two addresses, a temporary address and the actual address to use. The former is used only once in the uniqueness check to minimise the possibility for it to be non unique. The unique check is based on sending an address request (AREQ) and expecting an address reply (AREP) back in case the address is not unique. In case no AREP is received, the uniqueness check is passed.

The process differs slightly when using IPv4 or IPv6

#### *IPv4*

In this case, a node attempts to select a random address on the network 169.254/16. For IPv4, the messages are Internet Control Message Protocol (ICMP) packets.

#### *IPv6*

For IPv6, on the other hand, the AREQ is a modified neighbour solicitation and AREP is a modified neighbour advertisement as specified in the “Neighbour Discovery Protocol for IPv6”[22]

The process for the auto-configuration of IPv6 addresses is based on the “IPv6 Stateless Address Auto-configuration” [10], but with a number of changes. The IPv6 Stateless Address Auto-configuration process specifies a process that can be used to create link local unique addresses in the absence of a DHCP server. Ad hoc networks can't use link local address due to the fact that the addresses are valid over a multiple hop distance, not only to the immediate neighbours, changes need to be made to the process to ensure the addresses are unique within the network.

#### *Internet Connectivity*

This process allows for a node to obtain a globally unique IPv6 Address, however obtaining a globally unique IPv4 address is out of its scope. In order to connect to the Internet a gateway node needs to be available. If a gateway node is available a node should be able to connect to the Internet through the “Global Connectivity for IPv6 Mobile Ad Hoc Networks” [16] process.

#### *Problems*

The major problem with this proposal is that it does not consider the possibility of merging and partitioning.

- Mansoor Mohsin, Ravi Prakash. 2002 “IP Address Assignment in a Mobile Ad Hoc Network” in Proceedings of Milicom 2002. [23]

This paper uses the concept of binary split in order to perform dynamic configuration. It uses a flat structure; every node can assign an IP address to a new node without consulting any other node in the network.

The paper only considers a stand alone MANET (Mobile Ad Hoc Network), i.e. the network does not have access to the Internet. Below is an outline of the basic idea.

In the beginning, there is only one node in the network that has the entire pool of addresses. When an un-configured node A, wishes to join the network, it requests the nearest configured node, B, for an IP address. Node B assigns the requesting node A an IP address from its pool of IP addresses. It also divides the set of IP addresses into two and gives one half to the requesting node A (keeping the other half for itself)

A node can leave the network either gracefully or abruptly. When node A leaves a network gracefully, it gives its pool of IP addresses to any node B nearby. Node B then has the responsibility for handling this set of addresses. On the other hand, when node A leaves the network abruptly it leads to IP address leak (because there is some IP address that is neither assigned to any node nor available for assignment to an un-configured node). This situation is handled through synchronization. Nodes synchronise from time to time to keep track of the IP addresses assigned and detect any leaks in the available pool of IP addresses.

This paper also addresses the problem of network partitioning and merging through the use of a partition ID. When a new network is formed a new partition ID is created and this ID is passed to new nodes as they join the network. When a node detects that it has been partitioned from the main network through the synchronisation process it assigns itself a new ID.

When two partitions merge there may be a problem as two different nodes in the two partitions may share the same IP address. This paper presents an algorithm which they believe will solve this problem, its details are similar to the “Manetconf: configuration of hosts in mobile ad hoc networks” [20].

One issue with this paper is that networks merge even if they have no reason to do so. Also the problem of security has not been addressed.

- Subir Das, Anthony McAuley, Archan Misra. 2001 “Auto-configuration, registration, and Mobility Management for Pervasive Computing” in IEEE Personal Communications August 2001, pp. 24-31. [24]

Focusing only on the auto-configuration part of this paper only, the paper proposes the Dynamic Registration and Configuration Protocol (DRCP), which tries to extend Dynamic Host Configuration Protocol (DHCP) [9] to a stateless auto-configuration protocol for wired and wireless networks. Basically it presents a distributed DHCP architecture. Each node represents a DRCP client and server and owns an IPv4 address pool. The Dynamic Address Allocation Protocol (DAAP) is responsible for the distribution of the address pools. Each node requesting a pool obtains half of the pool of a neighbouring node. This may lead to a lot of unassigned addresses in the already scarce IPv4 private address space, and subsequently to scalability problems.

The major problem with this proposal is that the problems of network merging, partitioning or Internet connectivity are not considered.

## 4 Future Directions

Our future research will focus on the dynamic IP address assignment problem associated with Ad Hoc Networking; there are a number of problem areas, which could be further researched.

- **Merging of Networks**  
In most of the papers we have seen, the partitioning and merging of networks has not been addressed in detail. One potential area of research here is in the service discovery, if two networks are to merge together there will presumably be a reason to do so, for example one network may offer printing facilities to another network, issues of service discovery are important here. Other potential areas here include methods for detecting partitioning and merging of networks.
- **Security in the auto configuration process** has not been addressed, denial of service attacks are one possible security flaw, one node for example may request all the potential IP addresses available.
- **Internet Connectivity**  
This problem is closely related to the routing problem, and the problem differs depending on whether you are using IPv4 or IPv6.
- **The applicability of Mobile IP in Ad Hoc networks?**
- **Routing**  
Could the IP address assignment process be optimised for different IP address assignment protocols? For example if a hierarchical routing protocol is used, which utilizes clusters and cluster heads. Would it be more efficient for a hierarchical IP address assignment protocol to use the clusters and cluster heads identified by the routing protocol or to create its own. Also could routing information be used in the address assignment protocol, for example, if a node finds that it cant route information to a particular node, can it assume that that node has left the network? This kind of information could be useful for the IP address assignment protocol as it may allow nodes to identify the departure of a node more quickly.
- **IPv4 vs. IPv6**  
Should the IP address assignment solution be independent of IP version in use, i.e. will one solution work for IPv4 and IPv6 or will two different solutions be needed?

## 5 References

- [1] P. Bhagwat ,C. E. Perkins, 1994 “Highly Dynamic Destination –Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, Proceedings of ACM SIGCOMM’94.
- [2] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, 2002 “Optimized Link State Routing Protocol”, Work in Progress, Internet Draft, MANET Working Group
- [3] M. Gerla, X. Hong, G. Pei, 2001 “Fisheye State Routing Protocol (FSR) for Ad Hoc Networks”, Work in Progress, Internet Draft, MANET Working Group
- [4] Y-C. Hu, J. G. Jetcheva , D. B. Johnson, D. A. Maltz, 2000 “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)” Work in Progress, Internet Draft, MANET Working Group
- [5] C. E. Perkins, E. M. Belding-Royer, S. R. Das, 2001 “Ad Hoc On Demand Distance Vector (AODV) Routing, Work in Progress, Internet Draft, MANET Working Group
- [6] S. Coran V. Park, 2001 “Temporally-Ordered Routing Algorithm (TORA) Version 1 Fundamental Specification”, Work in Progress, Internet Draft, MANET Working Group
- [7] Z. Haas, M. Pearlman, 1997 “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, Work in Progress, Internet Draft, MANET Working Group
- [8] I. Cardei, 2002 “LANMAR: Landmark Routing for Ad Hoc Wireless Networks”, Wireless Networking Seminar, University of Minnesota, February 2002.
- [9] R. Droms 1997 “Dynamic Host Configuration Protocol”. Network Working Group, (Draft Standard) 2131.
- [10] S. Thomson and T.Narten 1998 “IPv6 Stateless Address Auto Configuration” (Draft Standard) 2462. Internet Engineering Task Force.
- [11] Christian Bettstetter and Jin Xi. 2002 “Wireless Multihop Internet access: Gateway Discovery, Routing, and Addressing” in Proceeding of International Conference on Third Generation Wireless and Beyond (2002)
- [12] Kilian Weniger, Martina Zitterbart. 2002 “IPv6 Auto Configuration in Large Scale Mobile Ad Hoc Networks” in Proceedings of European Wireless 2002
- [13] Jung-Soo Park, Yong-Jin Kim, Sung-Woo Park. 2001 “Stateless Address Auto Configuration in Mobile Ad hoc Networks using Site local Addresses”. Internet Draft.
- [14] C. Perkins 2002 “IP Mobility Support for IPv4”, Network Working Group, (Draft Standard) 3344

- [15] J. Arkko, D. Johnson, C. Perkins 2002 “Mobility Support in IPv6”, IETF Internet Draft (Work in Progress)
- [16] C. Perkins, J.Malinen, A. Nilsson, A. Tuominen, R.Wakikawa. 2002 “Connectivity for IPv6 Mobile Ad Hoc Networks”. IETF Internet Draft (Work in Progress)
- [17] K. Egevang, P.Srisuresh “Traditional Network Translation (NAT)”, Network Working Group, RFC 3022
- [18] E. Belding-Royer, Y.Sun. 2001, “Connectivity for IPv4 Mobile Ad Hoc Networks”. IETF Internet Draft (Work in Progress)
- [19] Nitin N. Vaidya. 2002 “Weak Duplicate Address Detection in Mobile Ad hoc Networks” in Proceedings of Mobicom 2002
- [20] Sanet Nesargi, Ravi Prakash. 2002 “MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network” in Proceedings of Infocom 2002.
- [21] Perkins et al. (Nov 2001) Internet Draft “IP Address Auto-configuration for Ad Hoc Networks, Technical Report. Internet Engineering Task Force, MANET Working Group.
- [22] T.Narten, E. Nordmark and W. Simpson. 1998 “Neighbour Discovery Protocol for IPv6”. (Draft Standard) 2461. Internet Engineering Task Force.
- [23] Mansoor Mohsin, Ravi Prakash. (2002) “IP Address Assignment in a Mobile Ad Hoc Network” in Proceedings of Milicom 2002.
- [24] Subir Das, Anthony McAuley, Archan Misra. 2001 “Auto-configuration, registration, and Mobility Management for Pervasive Computing” in IEEE Personal Communications August 2001, pp. 24-31

## "Managing a Smart Space" Theme

*One of the difficulties of managing a general smart space is that it will typically be put to different uses by different people at different times. The demands which applications place on the infrastructure vary considerably depending on the exact user population and their on-going tasks. Furthermore it is well known that there is no canonical way in which to view a set of spaces optimally - different sets of applications will have a different view of the same resources. These characteristics mitigate against static management. - M-Zones proposal*

The M-Zones approach of overlaying a particular set of smart spaces is highly dynamic, mandating highly flexible and adaptive architectures and technology solutions.

Taking this into consideration, a survey was undertaken analysing the applicability of current network management architectures and major surveys were undertaken into software infrastructure solutions that provide for service flexibility at runtime, including:

- (i) Context Management. A key enabler of service flexibility is the management of context information. This involves the gathering, interpretation, storage and dissemination of context information dynamically and in real-time.
- (ii) Service Composition. Service composition addresses techniques for rapidly integrating existing services into richer, composite services.
- (iii) Middleware. Technologies surveyed include those emerging from the IT domain (e.g. Jini, Intelligent Agents etc.) and from the telecoms domain (e.g. Parlay)

In addition, two particular areas of research were surveyed in order to investigate the applicability of existing approaches that provide adaptation at runtime:

- (iv) Adaptive Hypermedia. These techniques surveyed have implications for personalisation in many facets of Smart Space design
- (v) Policy Techniques. By enabling decisions to be made closer to where events are detected within Smart Spaces, policy techniques potentially allow a less centralized and more flexible management architecture.

## **State of Art Survey: Intra-domain and Inter-domain Management of Smart Space Environments**

Robert O'Connor,  
Telecommunications Software Systems Group,  
Waterford Institute of Technology.

### **Abstract**

This paper surveys network management architectures and their applicability to Smart Space environments. It highlights lessons that can be learned from SNMP, X.700, TMN and CIM with respect to developing a multi-domain Smart Space architecture based on the following criteria: architectural information components; communications protocols and formats; management services.

**Keywords** Ubiquitous computing; Pervasive computing; Smart Spaces; Network Management; Intra-Domain; Inter-Domain; TMN; SNMP; OSI X.700; CIM; Management of Smart Space Environments.

## **1. Introduction**

Ubiquitous Computing; Pervasive Computing; Smart Spaces – these are all terms used to describe the strongly emerging trend in Information Technology towards highly dynamic, heterogeneous, computing environments. Areas utilising ubiquitous technology usually exhibit the following characteristics:

- contain numerous, casually accessible and often invisible computing devices
- utilise mobile and imbedded environmental infrastructure
- connect to an increasingly ubiquitous network structure

The basic definition of a Smart Space is a physical space rich in devices and software services that is capable of interacting with Users, the physical environment and external networked services. As computing power increases, hardware size decreases and programming techniques become more accessible, the potential for mobile computing is growing. Within Smart Spaces everything has the possibility of being a computer. Pervasive Computing is about creating such technologies and infrastructures and developing the devices and services that will deliver this ubiquitous computing experience.

The research topic, “Intra-domain and Inter-domain Management of Smart Space Environments” has the potential to yield many benefits that could contribute greatly to the field of Ubiquitous Computing. There are many projects across the globe engaging in research into certain specialised areas of Smart Space technology. However, very little effort is being expended on the management structures that will allow differing component technologies to successfully inter-operate with one another. Intra-domain management is concerned with the internal management of a Smart Space, while Inter-domain management is concerned with external communications and operability across multiple Smart Spaces.

This paper summarises computer network management architectures and the more traditional telecommunications network management techniques and highlight aspects of these architectures that

may prove useful when attempting to describe future network architectures for emerging ubiquitous environments. The successful deployment of truly ubiquitous services is dependent on these heterogeneous Smart Space environments communicating, collaborating and interacting with one another, hence it is vital that intra and inter zone architectures are studied, researched and developed.

This paper has the following structure: Firstly, network management architectures that are of interest to the M-Zones project are given a brief overview. Then, each architecture is analysed based on the following criteria: architectural information components; communications protocols and formats; and the services each management structure provides. After this, the management architectures are discussed in terms of the direct relevance each bears to the M-Zones project. Research projects with similar goals to M-Zones are also identified and briefly evaluated in this section. Finally, conclusions are drawn and future work outlined.

## 2. Overview

The Simple Network Management Protocol (SNMP) (Case, Fedor, Schoffstall & Davin, 1990) was originally designed for the management of networks based on Internet Protocols (IP). The basic principle of SNMP is simplicity (KISS - keep it small and simple). This results in small, simple, and mostly cheap Agent software applicable for the devices of IP-based networks, such as modems, bridges, hubs, routers, printers, etc. SNMP defines a framework for the management of IP based data communication network devices. With the different versions of SNMP (v1, v2, v3), the functions added have become increasingly more complex. SNMPv3 resolved this problem by using modularity to allow the evolution of portions of SNMP without requiring a redesign of the general architecture.

The goals of SNMP are:

- the complexity and number of management functions in the management agent should be kept to a minimum (i.e. keep the agent as light as possible)
- the functional paradigm for network control and monitoring should be sufficiently flexible and extensible to accommodate unanticipated additional network functions
- the management architecture should be independent of the architecture of the gateway or hosts

The X.700 OSI Management Environment (ITUT-X700, 1992) includes both the capability for managers to gather information and to exercise control, and the capability to maintain an awareness of and report on the status of resources in the managed network environment. This responsibility may be manifested in terms of autonomous management of the open system and co-operation with other open systems; through the exchange of information; and through the performance of co-ordinated management activities. Systems management provides mechanisms for the monitoring, control and co-ordination of resources and protocol standards for communicating information relevant to those resources. In order to describe management operation on resources, resources are viewed as managed objects with defined properties. Information required for systems management purposes in any open system may be provided through local input; may result from input from other open systems through systems management communication; or may be a result of lower layer network protocol (Zimmermann, H, 1980) exchanges.

Telecommunications Management Network (TMN) (ITUT-M3000, 2000) provides an architecture to transport, store and process information used to support the management of telecommunications networks and services. It is based on the X.700 OSI Network Management Model as described earlier.

It also employs other OSI application service frameworks such as X.500 Directory Service (ITUT-X500, 1993), but these are beyond the scope of this document.

The Distributed Management Task Force (DMTF) is an industry association dedicated to promoting and developing standards for distributed environments, systems management and interoperability. The Common Information Model (CIM) (DMTFa, 1999) is an object-oriented information model that provides a framework through which management data can be abstractly modelled.

### **3. Analysis**

The management architectures are now discussed in terms of architectural information components; communications protocols and formats; and the services that each provides.

#### **3.1 Information Architecture Components**

##### **3.1.1 SNMP**

With SNMP, the guiding principle is simplicity. Management information is stored in a Management Information Base (MIB) in the form of Managed Objects (MO). An SNMP MO is not an object in the object-oriented sense, but instead represents a variable object. Instances of MOs can be accessed with the protocol elements of SNMP to traverse tables and to send simple traps (messages) to notify managers about events that have occurred at the controlled devices. Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, which have management agents responsible for performing the network management functions requested by the network management stations. SNMP is used to communicate management information between the network management stations and the agents in the network elements.

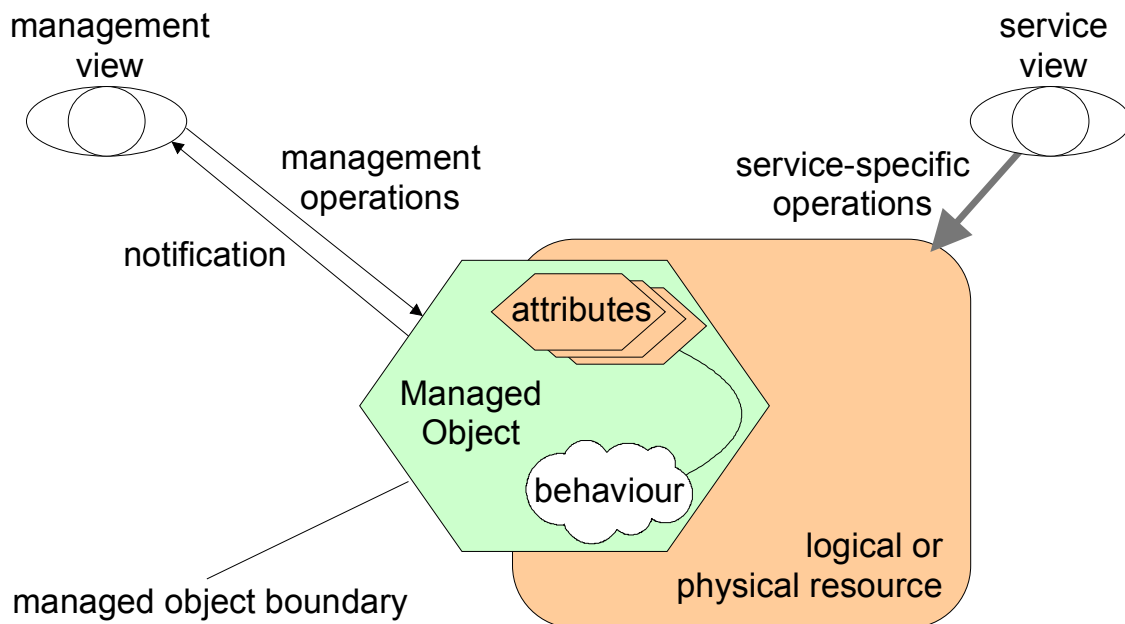
(Case, Fedor, Schoffstall & Davin, 1990)

##### **3.1.2 OSI**

Management of a communications environment is an information processing application. Because the environment being managed is distributed, the individual components of the management activities are themselves distributed. Management applications perform the management activities in a distributed manner by establishing associations between systems management application entities.

(ITUT-X700, 1992)

An MO is the OSI Management view of a resource that is subject to management, such as a layer entity, a connection, a Directory Service Agent or an item of physical communications equipment. Thus, an MO is the abstraction of such a resource that represents its properties as seen by and for the purposes of management. An essential part of the definition of an MO is the relationship between these properties and the operational behaviour of the resource. This relationship is not modelled in a general way.



**Fig 3.1.2.1 Managed Object**

Managed Objects can be specific to an individual layer of the OSI Network Model (ITUT-X200, 1994), in which case they are known as (N)-Layer MOs. MOs that are relevant to more than one layer, to a specific systems management function (management support object) or to the system as a whole are known as Systems Managed Objects.

The distinction between the MO as visible to management and the resource that it represents may be described by saying that the attributes, operations and notifications are visible to management as the Managed Object Boundary, whereas the internal functions of the resource are otherwise hidden from management. The concept of a managed object boundary is an abstract idea and may have no implications during implementation.

(Zimmermann, H, 1980) (ITUT-X720, 1992)

A Managed Object class is defined as a collection of packages, which are comprised of:

- The **attributes** visible at the MO's boundary
- The **operations** that can be applied to the MO
- The **behaviour** exhibited by the MO
- The **notifications** that can be emitted by the MO

An MO is an instance of an MO class. An MO exists (from a management perspective) if it has a qualified name and supports the operations and notifications defined for its class. Objects may logically exist that possess MO class properties, however unless they obey the above criteria, they do not exist from a management point of view. MO classes are arranged in a class hierarchical structure of superclasses and subclasses. This structure obeys logical inheritance laws whereby subclasses inherit all characteristics of their superclass. The OSI model also supports multiple inheritance, where MO classes may directly inherit from more than one superclass.

(ITUT-X720, 1992)

A group of Managed Objects that are comprised in a distributed system is said to be a Management Information Block (MIB). The programs that perform management operations on these MIBs are known as Management Information Services (MIS). An MIS program taking on the role of an Agent is the part of a distributed application that manages the MOs in its local environment. An Agent performs management operations on MOs as it is instructed from the Manager. An Agent will also forward notifications from MOs to the Manager.

An MIS program taking the role of a Manager is the part of a distributed application which has responsibility for one or more management activities. The Manager issues operations and receives notifications from Agents. An MIS program taking the role of Manager is not limited to applications engaging solely in systems management; other application requiring access to management information may take on this role.

Roles are not permanently assigned to MIS-programs. Some MIS programs can be restricted to taking on only an Agent role or only a Manager role, while others can take the Agent role in one interaction and the Manager role in another.

(ITUT-X700, 1992)

### 3.1.3 TMN

TMN management is performed through the use of Operations Systems (OS). A TMN provides management functions and communications between interconnected OSs and between OSs and the various parts of the telecommunications network. A TMN may also provide management functions and communications to other TMNs or TMN-like entities in order to fully support the management of wide range of telecommunications networks. A telecommunications network consists of many types of analogue and digital telecommunications devices and associated support equipment. When network management is applied, these devices are generically referred to as Network Elements (NEs). A TMN-like network is a network that is not based on the TMN model, but can interwork and communicate with a TMN.

A TMN is conceptually a separate network that interfaces with a telecommunications network at several different points to exchange information and to control its operations. A TMN may also use parts of the telecommunications network to provide its own communications and thus, there is a requirement for a certain amount of self-management to be performed on a TMN.

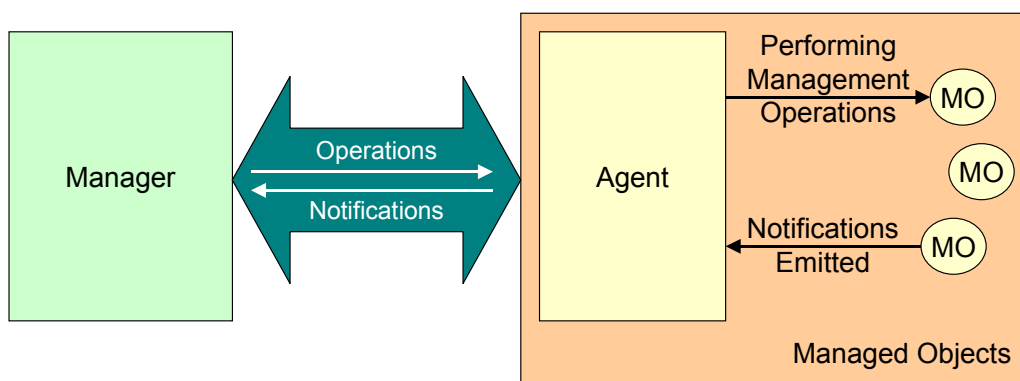
The **Information Architecture** describes the nature of information that needs to be exchanged between the functional building blocks of a TMN. Management of a telecommunications environment is an information processing application. Because such environments are distributed, the network management is also a distributed application. This involves the exchange of management information between management processes to monitor and control various physical and logical network resources. The management information is considered from two perspectives:

- The **Management Information Model** is an abstraction of management aspects of network resources and related supporting activities. This model determines the scope of the information that may be exchanged, concentrated at the application level and involves a variety of management application functions such as storing, retrieving and processing information. These functions are referred to as “TMN Function Blocks”.
- The **Management Information Exchange** concerns itself with lower levels of the OSI network layer model (Zimmermann, H, 1980) such as the communications layer and provides

functionality that allows physical components to attach themselves to the telecommunications network.

Because TMN is based on the X.700 model, a similar architecture comprised of Managers, Agents and Managed Objects is observed. Depending on the management function being performed at the time, management entities can take on either the Manager or Agent role. When a component assumes the Manager role, it takes responsibility for issuing management operation directives and receiving notifications. A component in the Agent role directly manages the associated Managed Object (MO) and receives and responds to directives issued by a Manager. It will also reflect a view of these MOs to a Manager and emit notifications concerning their behaviour.

Typically, “many-to-many” (M-N) relationships will exist between Managers and Agents in the sense that one Manager may be involved in information exchanges with several Agents, and one Agent may be involved in information exchanges with many Managers.



**Fig 3.1.3.1 Information Architecture (ITUT-X701, 1997)**

Management systems exchange information modelled in terms of Managed Objects (MOs). Similar to the OSI definition, MOs are conceptual views of actual resources (both physical and logical) that are being managed or support certain management functions. Object-oriented principles apply to the information modelling of MOs, but should not have any impact on the internal implementation of the telecommunications management system.

#### 3.1.4 CIM

The Common Information Model (CIM) applies basic structuring and conceptualisation techniques from the object-oriented paradigm to describe the management of systems and networks. A uniform modelling approach and the inherent properties of object-oriented descriptions support the co-operative development of management schema.

CIM provides a schema with respect to classification and association. This allows the managed environment to be described in a common framework. This framework is comprised of several layers:

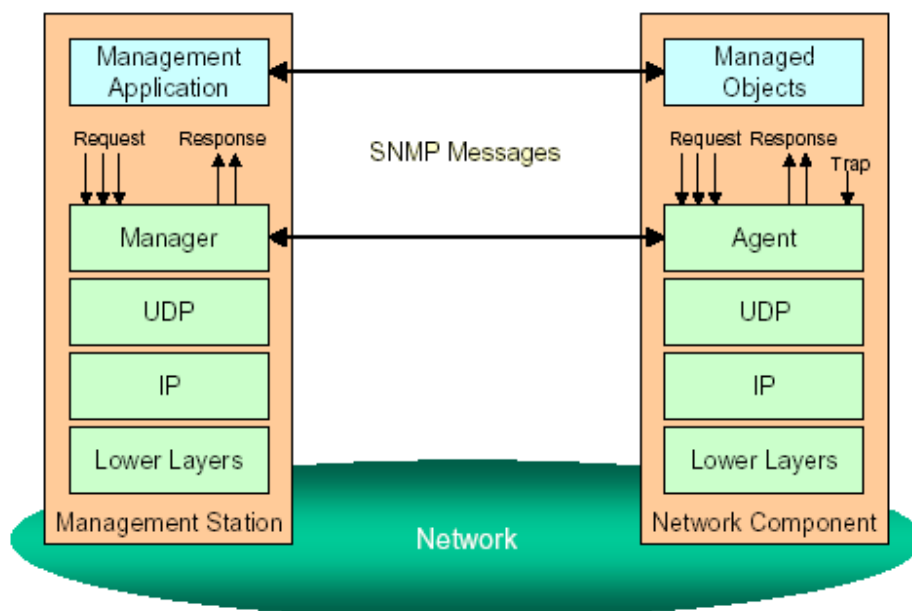
- The **Core Model** represents information that is applicable to all areas of management.
- The **Common Model** represents information that is common to particular areas of management.
- The **Extension Schema** represents information extensions that are unique to specific technologies or environments.

In a similar fashion to SNMP, OSI and TMN, the Manager-Agent dynamic exists, but the main focus of the architecture is built around the Core Model, the Common Model and the Extension Schema. (DMTFa, 1999)

### 3.2 Communication Services, Protocols and Formats

#### 3.2.1 SNMP

The Simple Network Management Protocol (SNMP) was (originally) designed for the management of networks based on Internet Protocols (IP). In a similar fashion to Internet communications, it operates on a connectionless basis. Management operations are performed through a series of retrieve (“get”) and alter (“set”) functions. The number of these commands continues to grow, however the semantics of the syntax is quite complex and does not interoperate well with non-SNMP systems.



**Figure 3.2.1.1 SNMP Communications Architecture**

#### 3.2.2 OSI

Since X.700 is an abstract notation, it is not bound to any particular technology when dealing with communications. However, there are strict guidelines in the specification as to how communication is to take place. Both connection and connectionless oriented transmissions can take place between MOs, provided both MOs support the protocols. The OSI Management Architecture also provides a management syntax that can be used describe the messages between MOs.

Communications in X.700 is provided by the Common Management Information Service (CMIS). CMIS provides the means for the exchange of information in management operations and notifications for management purposes in a common and standardised manner. CMIS uses the Common Management Information Protocol (CMIP) to transfer messages between MOs. Two types of information transfer can take place: Management Notification Service and Management Operation Service.

The behaviour of the communicating entities is dependent upon the specification of the MOs at which the notifications/operations are directed and is outside the descriptive scope of the CMIS. However, certain standard operations (such as GET and SET operations) that are used frequently within the scope of systems management are defined by CMIS.

(ITUT-X710, 1997)

### 3.2.3 TMN

TMN systems support the concept of interworking. That is to say that Management System A can communicate with Management System B. In order to interwork, communicating management systems must share a common view or understanding of certain management information, known as Shared Management Knowledge (SMK). This SMK provides the means by which disparate management systems can interface with one another. The logical concept of SMK can exist independently of the actual physical implementation. This is particularly the case for hierarchical management where a logical layered approach is taken.

It is necessary for a TMN system to support a wide variety of management areas, which cover the planning, installation, operations, administration, maintenance and provisioning of telecommunications networks and services. A TMN system should have the following functional abilities:

- to exchange management information between the telecommunications environment and the TMN environment
- to exchange management information between TMN environments
- to convert management information from one format to another so that information within the TMN has a consistent nature
- to transfer management information between locations within the TMN environment
- to analyse and react appropriately to management information
- to manipulate management information into a form which is useful and/or meaningful to the management information user
- to ensure secure access to management information by authorised management information users

(ITUT-M3010, 2000)

### 3.2.4 CIM

CIM provides a set of legal statement types that can be used to describe management operations in the information model or management schema. CIM is structured in a way such that the managed environment can be viewed as a collection of interrelated systems. Each of these systems is comprised of a number of elements, which can be described using the CIM syntax.

Management data is collected, stored and analysed using a common XML format. By using this common standard, 3<sup>rd</sup> parties can develop value-added functions that may be easily integrated into the managed system.

(DMTFa, 1999) (DMTFb, 1999)

### 3.3 Core Services and Application Services

#### 3.3.1 SNMP

An SNMP entity (Manager, Agent) is comprised of an Engine and one or more Applications. The SNMP Engine is made up of the following subsystems:

1. The **Dispatcher** is a key component in the SNMP engine. It dispatches tasks to the multiple version-specific Message Processing Models and sends PDUs (Protocol Data Units) to various applications. There is only one Dispatcher in an SNMP engine.
2. The **Message Processing Subsystem** is responsible for preparing messages to send and extracting data from received messages. An Engine can contain multiple Message Processing Models.
3. The **Security Subsystem** provides security services such as authentication and messages privacy. An Engine can contain multiple Security Subsystems.
4. The **Access Control Subsystem** provides authorisation services by means of one or more Access Control Models

(Harrington, Presuhn & Wijnen, 1999)

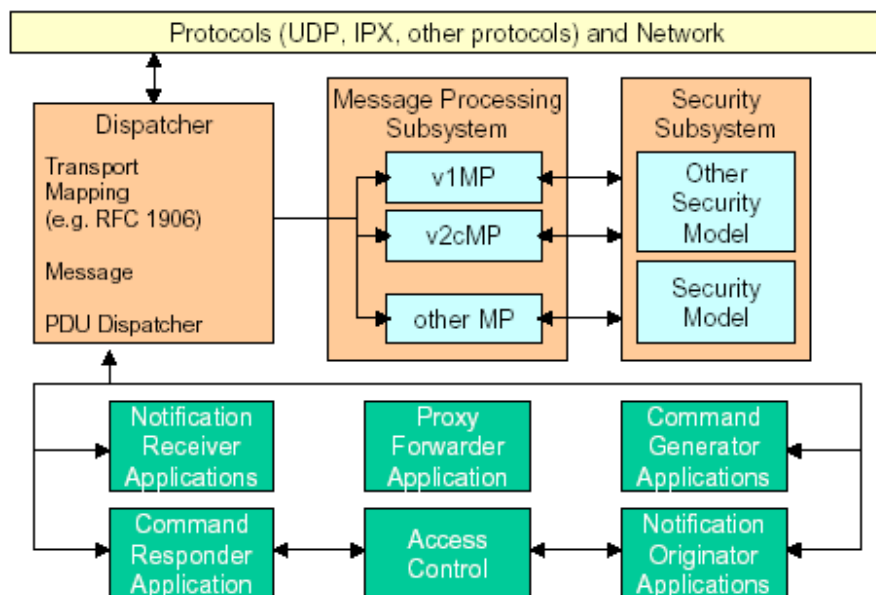


Figure 3.3.1.1 SNMP Engine

In an SNMP deployment, Managers and Agents comprised of the identified subsystems and applications communicate and collaborate with one another to manage network traffic, resources, etc.

#### 3.3.2 OSI

The role of OSI management is categorised into a number of functional areas: Fault Managements; Configuration Management, Accounting Management; Performance Management and Security Management. These are often collectively referred to as FCAPS functionality.

**Fault Management** is concerned with fault detection, isolation and the correction of abnormal operations in the managed network environment. Faults can cause systems to fail to meet their

operational objectives and usually manifest themselves as errors. Fault management includes functions that maintain and monitor error logs, accept and respond to error notifications, identify faults, carry out diagnostic tests and correct faults (where possible).

**Configuration Management** identifies, exercises control over and exchanges data with systems for initialising, providing and terminating interconnecting services. It includes functions to control normal operations, associate names with MOs and sets of MOs, start and stop MOs, collect information on demand about current conditions and to change configurations of services.

**Accounting Management** enables charges to be incurred for the use of resources with the managed network environment and for resource costs to be identified. Accounting Management includes functions to inform users of costs incurred, resources consumed, enable accounting limits to be set and tariff schedules to be associated with the use of resources and enable costs to be combined where multiple resources are utilised to achieve a given communication objective.

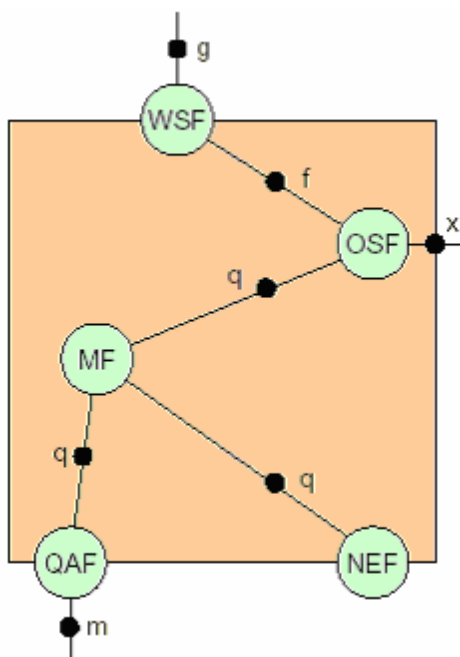
**Performance Management** enables the behaviour of resources in the managed network environment and the effectiveness of communication activities to be evaluated. It includes functions to gather statistical information, maintain and examine logs of system histories, determine system performance and alter system operation modes to conduct performance management activities.

**Security Management** supports the application of security policies by means of functions which include the creation, deletion and control of security services and mechanisms, the distribution of security information and the reporting of security events.  
(ITUT-X701, 1992)

### 3.3.3 TMN

A TMN system is comprised of a set of functional building blocks; Operations Systems Function (OSF), WorkStation Function (WSF), Mediation Function (MF), Network Element Function (NEF) and Q Adaptor Function (QAF). Operations take place between these blocks to achieve the functional goals of a TMN system. A Management Application Function (MAF) represents part of the functionality of one or more TMN management services. Each functional block has a set of MAFs associated with it:

1. **Mediation Function.** The MF-MAF is used in the support of Manager and Agent roles. MF-MAFs are optional and are used to provide supportive functionality in OSF. Examples of such functions are temporary storage, filtering, thresholding, etc.
2. **Operations Systems Function.** These management application functions are the essential and underlying aspects of management functionality blocks. The range from simple to complex functions, such as:
  - Support of Manager and Agent roles in access to managed object information
  - Adding value to raw information (e.g. data concentration, statistics, performance analysis, etc)
  - Reaction to incoming information (e.g. automatic reconfiguration, fault tracking, etc)
3. **Network Element Function.** These management application functions are present in the NEF primarily to support the Agent role.
4. **Q Adaptor Function.** These management application functions are present in the QAF primarily to support the Manager and Agent roles.



**Fig 3.3.2.1 Reference Points (ITUT-M3010, 2000)**

In order to illustrate connections between function blocks, the concept of **Reference Points** is introduced. Reference points define the interface characteristics for information exchange between function blocks. Each of these is a high level abstraction of communication and they do not dictate implementation protocols.

- The **q reference point** illustrates the logical part of the information exchange between function blocks, as defined by the information model mutually supported by the functions.
- The **f reference point** is used to denote information exchange between WSF and OSF blocks and/or WSF and MF blocks.
- The **x reference point** is used to illustrate communication between the OSF function blocks in different TMNs.
- The **g reference point** is used to denote communicate between human users and the WSF. It is not considered to be part of the TMN, even though it conveys TMN information.
- The **m reference point** is located between the QAF and other non-TMN managed entities or managed entities that do not conform to TMN standards.

To decrease complexity when defining a logical telecommunications management architecture, TMN is considered to be separated into logical layers. This **Logical Layered Architecture (LLA)** is a concept for the structuring of management functionality into four layers, where each layer restricts management activities within its boundaries in accordance with clear definitions, based on business, service, network and element management. The **Business Management Layer (BML)** has responsibility for the proprietary functionality of enterprise. The BML is included in the TMN architecture to facilitate the specification of capability that it requires of the other logically lower layers. The **Service Management Layer (SML)** is concerned with the services that are provided to customers. Some of the main functions of this layer are service order handling, complaint handling and invoicing. The **Network Management Layer (NML)** has the responsibility for the management of a network as supported by the Element Management Layer. Functions addressing the management of a wide geographical area are

said to be contained in this layer. It has complete visibility of the whole network and a technology independent view is provided for the SML. The **Element Management Layer** (EML) manages the Network Elements (NE) encompassed by the TMN system. It provides an abstract representation of NEs for the NML. Below the EML, the Network Elements are said to exist, but they are not considered a separate layer themselves.  
(ITUT-M3010, 2000)

### 3.3.4 CIM

Since CIM is an abstract notation, there are no directly implemented services available. However by following the notation guidelines, some useful structures can emerge.

A Directory Enabled Network (DEN) provides the building blocks for mapping users to network services, and business criteria to network services. Applications and services are able to leverage network infrastructure on the user's behalf to assist with service creation, maintenance and management. The central information repository of a DEN is directory where the intersecting relationships of users, applications and network services are defined. Networked applications are managed by associating users and applications with a set of resource policies. The DEN is used by the DTMF to describe cross-domain solutions.

The Web-Based Enterprise Management (WBEM) is an initiative to provide web access to enterprise management information and systems. Since the goal of the DTMF is to tie industry organisations and standards together, the WBEM is now part of the overall DTMF strategy. The DMTF have defined a mapping of CIM operations onto Hyper Text Transfer Protocol (HTTP) to allow WBEM operations to be performed in an open and standardised manner.  
(DMTFa, 1999) (Carey & O'Reilly, 2002)

## 3.4 *Relevance to M-Zones*

A good analogy of Smart Spaces is the idea of a series of unconnected islands. Each of these islands represents a smart space, operating independently of one another. An architecture is required to allow these unconnected islands to interoperate in a cohesive and useful fashion. Smart Spaces using well-designed management architecture(s) would allow domains to exchange information, perform remote tasks and provide a greater level of functionality than Smart Spaces operating with no overall functional structure. Each smart space is based primarily upon its own network, which will most likely be IP based. Networks based on IP principals are the industry standard when referring to computer networks. Telecommunications networks are also moving in this direction, with both GPRS and UMTS being IP based. Smart Spaces are building upon computer and telecommunications technologies, so it would be logical to assume that they will also be IP based. It is important to study existing IP network management techniques and identify the aspects of them that may be applied when defining this architecture for interconnecting disparate zones.

From SNMP the most important lesson is simplicity. It is relatively easy to implement an SNMP compliant network and very few modifications need to be made to resources to become part of the managed network. However, SNMP places most of the burden on the Manager component and keeps the Agent relatively "light". In a multi-zone scenario, this may not be a scaleable solution, since a management component may serve many agents over a large area. In the event of a fault and the manager were unable to communicate with the agent, the local managed zone may cease to function. If

the agent were sufficiently autonomous, it would be able to continue most of its activities while the manager was offline. Any messages for the manager could be stored in a buffer and sent at a later time.

The connection-less aspects of SNMP are also quite interesting. Current Smart Spaces technologies are based mostly around wireless networking technologies, such as 802.11, Bluetooth and Infrared. Due to the large number of devices potentially present in a Smart Space and the transient nature of many of these elements, it is logical to assume that Smart Spaces will continue to predominantly use wireless communications methods. In a busy Smart Space with devices frequently entering and leaving, network connections will be established and broken at a much higher rate than traditional wired networks. In this way, it will be difficult to maintain connection-oriented communication protocols.

An M-Zones architecture will certainly be a more complex network management definition than SNMP, due to the increased physical network complexity, greater management functional scope and broader service definitions. However the underlying ideas of robustness and network element autonomy are important issues to consider when defining this network management system.

The OSI X.700 network management architecture presents some characteristics that may be useful when describing an ubiquitous networking architecture. The most notable are FCAPS functionality and the object-oriented hierarchical structures. The FCAPS model provides a clear set of functional objectives that must be achieved. Although FCAPS functionality may not map directly to pervasive environments, the principals of fault, configuration, accounting, performance and security management would be useful guidelines when outlining pervasive architecture(s).

X.700 presents two hierarchical structures for consideration. The object model class structure offers an abstract view of the network in an uninstantiated or ideal form. This is useful when describing a defined architecture as a separate entity to the actual physical network and network elements. The object model name-binding scheme presents a model of the actual instantiated objects and offers a representation of an actual working system. These two models will differ in appearance, but certain hierarchical similarities will exist. It is useful to compare the abstract ideal system with an actual working system. A similar approach would be useful when describing a Smart Space network management architecture, to highlight areas in which the abstract model differs from the implemented system.

The inheritance concept could also be useful when addressing the idea of an evolving network. As technologies advance, management architectures must adapt to suit more radical changes. A modular, object-oriented approach would allow certain components to be “upgraded” without requiring any great change to the overall architecture. One drawback of the OSI network management architecture is the lack of real world implementations. This would indicate that OSI is perhaps too complicated or flawed to exist in the real world? Or perhaps IP networks have thus far not required an architecture that offers as much functionality? Whatever the case, X.700 appears to offer some useful guidelines and raise important issues which will need to be addressed when developing a multi-domain management architecture.

The most striking aspect of TMN is the huge complexity of the architecture, as shown in Sections 3.1.3, 3.2.3 and 3.3.3. A trend of increasing complexity can be clearly seen when comparing and contrasting SNMP, OSI X.700 and TMN with one another. However, TMN also brings with it a very well defined and tested architecture, which can be somewhat lacking when discussing computer

network management architectures. TMN is based upon the X.700 model as described earlier, hence many of the characteristics highlighted as being of importance to the M-Zones project are also applicable here. The Manager, Agent and Managed Object dynamic can be clearly seen again, cementing the view that this model is of particular importance. Considering Managers, Agents and Managed Objects, there is an immediate network management abstraction with respect to ubiquitous devices.

One of the most important aspects of TMN that has direct relevance to M-Zones is the way in which it deals with the concept of interworking and inter-domain communication. The SMK (Shared Management Knowledge) concept described in Section 3.2.3 is certainly worth investigating. Defining communication interfaces for every possible inter-zone communication scenario may be an impractical solution when dealing with Smart Space systems, however this model warrants further study. Of equal importance is the Reference Point concept, specifically the x and m points which deal with information exchanges outside of the TMN environment. When dealing with Smart Spaces, a set of interfaces to networks that are not necessarily Smart Spaces themselves will be required and lessons can be learned from TMN approach to communication with other TMN and non-TMN environments.

The layering model as outlined by the LLM is an abstraction that may prove useful when attempting to define logical pervasive management applications. If applications can be defined with an abstract vision, management issues may be highlighted, than by merely defining applications in terms of network elements, managers, agents etc. Also the concept that the layer definitions may or may not dictate physical implementation strategies is worth noting. Physical implementation techniques are often incompatible with logical modelling and a separation of the two could be quite useful, especially with respect to emerging and untested ubiquitous devices and standards. Many other aspects of TMN do not concern themselves with implementation issues, which reinforces this view. When defining a multi-domain management architecture for Smart Spaces, it will be important to assume an abstract position. Focussing on implementation-specific issues may cause more general management issues to remain unstudied. Also adopting this broader view may allow for many abstract architectures to be considered, as opposed to one all-encompassing network management system. It may be the case that different Smart Spaces will require different degrees of management architecture, based on requirements. E.g. A Smart Space in a café that offers wireless Internet access to its customers will not require as much management as a Smart Space in a shop that allows customers to pay for goods directly from their bank account (e.g. Smart Space debit 'card'). Logically, the shop would require greater levels of authentication and security in its Smart Space management system than the café. Will they both use one system; different versions of one system; or totally different systems? This is an important question to ask when beginning to develop management architecture(s) for Smart Spaces.

TMN highlights many issues that need to be considered when describing Smart Space network management architecture. Any architecture(s) produced by M-Zones will not be as rigid as TMN, because they will need to be suitably flexible to adapt to different Smart Spaces. Again, this raises the issue as to whether there will be many architectures or one all-encompassing system. Further research is required in this area. Another important issue to highlight, is that it is any architecture(s) developed by the M-Zones projects must be able to interface and communicate with other managed networks (e.g. TMN, SNMP) to provide a genuinely ubiquitous service. Knowledge of these systems will be required when detailing these interfacing components.

The CIM as proposed by the DMTF highlights some issues that deserve consideration. The fact that CIM is an abstract notation reduces the real-life implementation lessons that can be learned. However, due to the constantly evolving industry environment, basing an architecture on any one technology would not be prudent. An architecture specifically designed for a single technology needs to evolve with this very technology or it will become obsolete when the technology becomes obsolete. This approach is not desirable for Smart Spaces management architecture(s) as they must operate in a heterogeneous and dynamic technical environment. However, as mentioned with respect to both X.700 and TMN, creating an architecture model that is not implementation specific (but has implementation guidelines) may prove to be effective.

CIM can be used a mediation technology between multiple networks. The WBEM initiative bears a lot of relevance to the practical stages of the M-Zones project. WBEM applications use standard Internet technologies and protocols to provide implementations of management systems. The fact that these technologies are well defined and tested could be useful when implementing an M-Zones architecture. Web-based applications would be ideal during the implementation stages of the project due to the fact that many are freely available and well supported.

The Extension Schema also provides an interesting aspect. This allows 3<sup>rd</sup> party organisations to add components as they see fit. Allowing a Smart Space architecture to be extensible might assist in industry adoption and encourage other developers to create services that interact with it. The use of XML as a communications standard within the CIM is also worth noting. XML is a standard mark-up language that has gained a lot of popularity and is widely used in many systems. This highlights XML as a possible candidate for a messaging standard within any Smart Space architectures that the M-Zones project may produce.

### **3.5 Similar Work**

The eBiquity research group from the University of Maryland Baltimore County undertook the task of providing an infrastructure and communication protocol for wireless services. The architecture developed by the **Centaurus** project consists of a number of components that facilitate communication between portable devices and the Centaurus System (CS) within a confined space. The Communications Manager (CM) handles all communications between the client and the rest of the Centaurus system. The Service Manager (SM) acts as a mediator between clients and services and is responsible for client and server registration, service leasing and service discovery. The communication medium used by the Centaurus project is a derivative of XML, which they named Centaurus Communication Markup Language (CCML). They also developed Centaurus Communication Protocol (CComm), which was used to communicate with mobile clients and services. The Centaurus project ended in 2001 and was superceded by Centaurus 2 (Undercoffer, Cedilnik, Perich & Joshi, 2001) and Vigil (Kagal, Undercoffer, Perich, Finin & Yesha, 2001). However, both of these projects have concentrated on security aspects of pervasive computing. (Kagal, Korolev, Chen, Joshi & Finin, 2001)

The most interesting aspect of Centaurus from an M-Zones point of view, is their use of XML when developing a their communication language. XML is a very simple text-based standard that most platforms can understand. It is lightweight and few resources are required to read/write XML documents. When dealing with many ubiquitous entities of differing computational ability, it is important to ensure that entities of similar complexity can communicate as easily with each other as entities with a simpler design and vice versa. XML presents a strong argument for itself when deciding

on standards for communication. It is very well defined and tested and there is a lot of software freely available that supports it. The Centaurus project is limited in its relevance to M-Zones since it only concerned itself with managing services in a single smart space. However, it highlights the need for management at the local level. Inter-domain management is just a step above managing a single domain and any M-Zones multi-domain architecture will have to interoperate with local Smart Space management systems. This reinforces the idea that M-Zones should concern itself with both intra and inter zone management.

#### **4. Future Directions**

There are many potential rewards as a result of developing management structures that successfully allow components to inter-operate with each other across multiple domains. The explosive growth of the World Wide Web and its corresponding increase in popularity and subsequent development of Internet technologies illustrates just how powerful large scale networks can be. Smart Spaces present an opportunity to expand the reach of computer networks into areas previously thought to be inaccessible. New services, models and usage practices may develop as a result of this, furthering the impact of Information Technology in people's lives.

The ultimate goal of this research topic is to produce a new type of management architecture. It is difficult to imagine that Smart Spaces could be adequately managed by any of the management technologies described. However, the initial step to be taken when attempting to create a new management structure is to study existing architectures. The most immediate problem when surveying these management architectures is identifying areas that bear relevance to the field in question and isolating the subsections that are of particular interest. Lessons learned from studying SNMP, X.700, TMN and CIM (and to a lesser extent the Centaurus project) will prove invaluable to the M-Zones project when attempting to develop its own Smart Space network architecture(s). From SNMP, simplicity is the key recognisable factor. An architecture that becomes bloated will most likely become too complicated to be useful. The connectionless communication methods used by SNMP are also worth noting because of the higher frequency of broken connections in wireless networks. The OSI X.700 management architecture presents a level of abstraction not seen in SNMP. A management notation method that is separate from implementation may prove effective when dealing with pervasive environments. Unlike traditional computing networks, which remain relatively stable in terms of connected devices, Smart Spaces will be ever-changing with devices constantly entering and leaving the environment. This would be very difficult to model without a certain degree of abstraction. The FCAPS functionality described by X.700 is also a good guideline that may be used when attempting to develop functional aspects of Smart Space architecture(s). From TMN, there are lessons to be learned regarding inter-working and inter-domain communications. Smart Space management architecture(s) may not be quite as rigid as TMN, but they will certainly share characteristics, especially in this respect. The most important aspect of note from the DMTF's CIM is the use of a common messaging standard. This highlights the issue of the need for a standard messaging protocol in Smart Space environments. Since devices will have varying degrees of complexity and processing abilities, simplicity will again be a key issue. The WBEM initiative illustrates some initial steps that may be taken when the M-Zones project moves into the practical experimentation phase.

The next step in this field of research is to study some ubiquitous technologies and catalogue areas that are of particular importance to management. As a result of this, some experiments will be conducted with Smart Spaces with a vision to producing some preliminary management architectures. Over the

lifecycle of the project, various views, iterations and versions of architectures will be presented. It is the goal of the project to provide multi-domain network management systems that can assist in the delivery of ubiquitous services and provide the user with a seamless Smart Space experience.

## 5. References

Carey K., O'Reilly F., 2002, "Heterogeneous Tools for Heterogeneous Network Management with WBEM", DMTF Developers Conference 2002, San Jose. Adaptive Wireless Systems Group, Department of Electronic Engineering, Cork Institute of Technology, Cork, Ireland.

Case J., Fedor M., Schoffstall M. and Davin J., May 1990, "A Simple Network Management Protocol", RFC 1157.

Case J., Mundy R., Partain D. and Stewart B., April 1999, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570.

Case J., Harrington D., Presuhn R. and Wijnen B., April 1999, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572.

DMTF, Distributed Management Task Force, <http://www.dmtf.org>

DMTF, June 14, 1999: Common Information Model (CIM) Specification. DMTF, Version 2.2.

DMTF, July 20, 1999: Specification for the Representation of CIM in XML. DMTF Version 2.0

Harrington D., Presuhn R., and Wijnen B., April 1999, "An Architecture for Describing SNMP Management Frameworks", RFC 2571.

ITU-T Recommendation M.3000, February 2000: Telecommunications management network Overview of TMN Recommendations.

ITU-T Recommendation M.301, February 2000: Telecommunications management network – Principles for a telecommunications management network.

ITU-T Recommendation X.200, July 1994: Information Technology – Open Systems Interconnection Basic Reference Model: The Basic Model. International Telecommunication Unit, Geneva, Switzerland.

ITU-T Recommendation X.200, 1993: Open Systems Interconnection – Model and Notation – Specification of Abstract Syntax Notation One (ASN.1). International Telecommunication Unit, Geneva.

ITU-T Recommendation X.210, November 1993: Information Technology – Open Systems Interconnection – Basic Reference Model: Conventions for the Definition of OSI Services. International Telecommunication Unit, Geneva.

ITU-T Recommendation X.500, 1993: Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services. International Telecommunication Unit, Geneva.

ITU-T Recommendation X.501, August, 1997: Information Technology – Open Systems Interconnection – The Directory: Models. International Telecommunication Unit, Geneva.

ITU-T Recommendation X.700, September 1992: Management Framework for Open Systems Interconnection (OSI) for CCITT Applications. International Telecommunication Unit, Geneva.

ITU-T Recommendation X.701, 1997: Information technology – Open Systems Interconnection – Systems management overview.

ITU-T Recommendation X.710, 1997: Information technology – Open Systems Interconnection – Common management information service.

ITU-T (CCITT) Recommendation X.720, 1992: Information technology – Open Systems Interconnection – Structure of management information: Management information model.

ITU-T (CCITT) Recommendation X.721, 1992: Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.

ITU-T (CCITT) Recommendation X.722, 1992: Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.

Kagal L., Korolev V., Chen H., Joshi A., Finin T., April 2001. “Project Centaurus: A Framework for Indoor Services Mobile Services.” In Proceedings of International Workshop on Smart Appliances and Wearable Computing (IWSAWC) at The 21<sup>st</sup> International Conference on Distributed Computing Systems 2001. Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore MD, USA.

Kagal L., Undercoffer J., Perich F., Finin T., Yesha Y., October 2001. “Vigil: Providing Trust for Enhanced Security in Pervasive Systems.” Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore MD, USA.

Undercoffer J., Cedilnik A., Perich F., Joshi A., June 2001 “Centaurus 2: A secure infrastructure for multi domain service discovery and utilization.” Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore MD, USA.

Zimmermann, H, 1980. OSI Reference Model -- The ISO Model of Architecture for Open Systems Interconnection. IEEE Transactions on Communications.

## **Adaptive Technologies**

Smart Spaces offer a vision of computing systems that are embedded in the fabric of everyday life. Smart Space systems will therefore become an increasingly invisible part of our everyday lives, providing us with seamless access to information and communications regardless of our location. Human interaction with smart spaces will become a more natural affair, with user needs being inferred using inputs from multi-modal sensors and context information. In addition, if people are able to interact with any smart space environment they visit, we must allow the owners of the local resources to limit their usage according to their security, resource allocation or charging policies.

Smart Space environments must adapt dynamically therefore to a combination of user needs, the policies of resource owners, the current usage context and the capabilities of surrounding services.

There are many areas of research that have been seeking to develop adaptive techniques. Two areas in particular have been identified by the Mzones team as potentially suited to contributing towards the solutions needed within Smart Spaces. For these two areas the state of the art has been explored and the results are provided here. First the state of the art in adaptive hypermedia in eLearning domain is presented. This area was surveyed as it is our belief that the research undertaken to date in this domain could contribute to the personalisation solutions required in many facets of Smart Space design. Secondly, the state of the art in policy techniques is provided. Policies represent an important existing approach to providing runtime flexibility in the operation of management components and systems and therefore are a strong candidate

## State of the Art: Adaptive Hypermedia

Owen Conlan  
Knowledge and Data Engineering Group  
Department of Computer Science  
Trinity College Dublin

### 1. Introduction

Adaptive Hypermedia (AH) is one of the most promising application areas for user modelling and user-adapted interaction techniques [Brusilovsky, 94]. AH systems can be useful in any situation where the system may be used by people with different goals and knowledge and where the hyperspace is reasonably large. Users with different goals and knowledge may be interested in different pieces of information presented on a Hypermedia page and/or may use different links to navigate to those pages. AH tries to overcome this problem by using knowledge about a particular user, represented in the user model, to adapt the information and links being presented to the given user.

From the perspective of M-Zones, Adaptive Hypermedia and the techniques researched in the broader adaptive eLearning domain have implications for personalisation in many facets of Smart Space design. These facets include –

- Content delivery (personalisation of content to users requirements, adapting to the users device and surroundings)
- Personalised views of a Smart Space (analogous to views in Databases; Smart Space visualisation – spatial or service based – may be adapted to the users preferences)
- Personalised/Adaptive service composition (adaptive navigation and presentation techniques utilised in AH may influence to customised combination of services to fulfil the users requirements)
- Adaptive management of services (as services are delivered across diverse smart spaces they may require resource management based on the unique characteristics of the Smart Space).

### 2. Overview

The underlying principles of software and information adaptivity may be divided into three discrete layers – the objectives of adaptivity, the techniques used on those objects and the axes across which those techniques or methods are employed. The primary field in which these objectives, axes and methods have been examined is in the eLearning domain, principally in the research area of adaptive hypermedia

#### Adaptive Hypermedia

Adaptive Hypermedia (AH) attempts to alleviate some of the difficulties encountered in Hypermedia systems by adapting the system individually for each user. The system collates information about each user into a user profile and this model is used to make assumptions about how best to change the system to benefit an individual user. The system may infer user objectives and help the user to discover the scope of information available or delineate a relevant path to get to the information required [De La Passardiere, Dufresne 92].

Adaptive systems infer the requirements of a user and modify the system accordingly. This introduces the problem of balancing control between the user and the system and the issue of the extent to which a user should be made aware of system made changes i.e. the transparency of the adaptivity. The correctness of assumptions made by the system cannot be guaranteed. This argument implies that users should be able to control the system adaptivity. Adaptable changes are those which originate from and are controlled by the user. Adaptive changes originate from and are controlled by the system [Fink et al 96].

The system adaptivity may be hidden entirely from the user so that the user is unaware of changes made by the system on her behalf. Alternatively the adaptivity may be negotiated with the user, allowing the user to accept or reject modifications suggested by the system. The modifications may be visible to the user but the user may not be able to change them. For example a link which is visible as a link, but dimmed and inaccessible to the user.

Users should have some control over the adaptivity but should not have to control it continuously [Espinoza, Hook 95]. System designers must attempt to strike a balance between the control allowed to the user and the ease of use of the system. It is imperative that users should not be surprised, disoriented or displeased by the changes made by the system [De La Passardiere, Dufresne 92]. When the usability of the interface is in opposition to the potential effectiveness of the system, the designer must attempt to provide adequate balance.

So what features of the system are modifiable? The system may customise the link structure or format which is offered - this is known as adaptive navigation. Similarly the system may vary the content displayed - this is known as adaptive presentation. The system may adapt the modality of the content or the prominence of links or content. Orientation aids and search facilities may be included, omitted or highlighted depending on the information contained in the user model and the rules used to apply the changes [Kay, Kummerfeld 95][Fink et al 96].

### **Adaptive Hypermedia in Educational Systems**

Much of the focus in adaptive hypermedia for educational courseware has attempted to alleviate the difficulties of content comprehension (cognitive overload) and orientation (so-called '*lost in hyperspace*' [Laurillard, 93]). Adaptive presentation techniques which effect changes to both the selection of different media depending on a users preferences and adaptation of the content based on an individual's user model are beginning to show success. Also the use of adaptive navigation which effect changes to the link structure between elements of the hypermedia courseware based on an individual user's (mental) model, has proven effective since learners using such systems have demonstrated faster learning, more goal-oriented attitude and take fewer steps to complete a course.

To achieve the maximum effectiveness from the use of non-adaptive Hypermedia in an educational context there are some features of learners that are particularly significant. These include preknowledge, cognitive style, maturity, general ability, confidence and motivation. These features influence the ability of students to accept effectively the additional mental load caused by the need to monitor and self-evaluate as well as learn [Specht, 98].

Although increasing learner control is thought to increase the learner's motivation and engagement, results in performance using adaptively controlled environments have been superior to systems within which the user is left to their own devices [Specht, 98]. Studies have shown that users of educational

Adaptive Hypermedia systems are faster, more goal-orientated and take fewer steps to complete the course. It is claimed that Adaptive Hypermedia learners are less likely to repeat the study of content they have already covered [Eklund, Brusilovsky, 98].

### **Adaptive Axes**

There are several adaptive axes that may be usefully employed in software and information adaptivity. These axes include adaptive navigation, structural and historical adaptation and adaptive presentation.

#### **Adaptive Navigation**

Adaptive Navigation attempts to guide the student through the system by customising the link structure or format according to a user model. The form of adaptive navigation will determine the level of guidance and freedom granted to the student within the system. Hypermedia experienced learners are known to be more likely to navigate in a non-linear way. Similarly learners who are familiar with the subject matter are more likely to navigate non-linearly and therefore reap the benefits of Hypermedia learning [Eklund, 95].

#### **Structural Adaptation**

Structural Adaptation attempts to give the student a spatial representation of the Hyperspace environment. This representation is based on the user model and is hoped to provide the student with a sense of position within the environment and a sense of the size of the environment itself. Overview maps, local maps, fisheyes, filters and indexes are all structural aids which the system may adapt for the student.

#### **Historical Adaptation**

Historical Adaptation attempts to give a time context to the student by adapting representations of the student's path through the system. History trails, footprints which are made by the system, landmarks which are made by the student and progression cues may be customised by the system for the student.

#### **Adaptive Presentation**

Adaptive Presentation is the customisation of course content to match learning characteristics specified by the user model. The granularity may vary from word replacement to the substituting of pages or the application of different media. Content may be customised to contain additional information, pre-requisite information or comparative explanations.

This form of adaptivity may be implemented by fragmenting the constituent content components into discrete words, phrases or paragraphs. These components of pagelets constitute a discrete unit of information about a concept. The pagelet is displayed if the user model conforms to required conditions for the display of that pagelet. For example, if a student has not covered a pre-requisite concept for a given page the relevant pagelet may be included.

With this approach different pagelets may be displayed for different students. An example would be a technical term or acronym with which the student is unfamiliar. The system may substitute the unfamiliar content until the student can be introduced to the technical term or acronym.

If the courseware is constructed dynamically each student may potentially see an individually tailored course that is different to the course displayed for all other users.

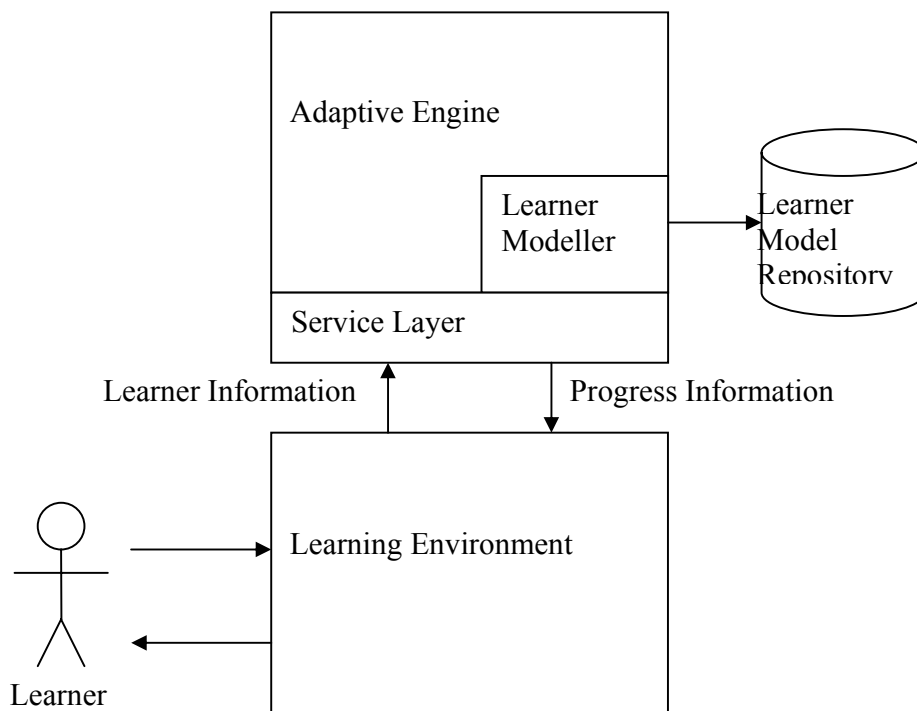
### **3. Analysis**

The techniques utilised in adaptive software and the methods used to realise these techniques in different systems (described above) have implications for how adaptive information services may be implemented within an M-Zone. Core to many of the adaptive information services (Interbook, PLS, AHA) is some form of concept model. Each of these systems takes a bespoke approach to developing the concept models and the vocabularies used to describe those models. In an smart space environment, where automatic sourcing of content, service combination and common user models are envisaged, it is desirable that the vocabulary, and by implication the ontology structuring that vocabulary, should be common between, or at least discoverable by, the different services. Another alternative to this approach is that the services may attempt to approximate mappings between subsets of the vocabularies.

#### **3.1 Service versus Stand-Alone System**

The majority of Adaptive Hypermedia Systems are designed to exist as stand alone systems (Interbook, AHA). KnowledgeTree [Brusilovsky, 02] and PLS [Conlan et al, 02] however are two AHSs that take different approaches to exposing their adaptive content as services. Indeed neither KnowledgeTree or PLS should be viewed as traditional AHSs as they are designed to source content and functionality (such as Learner Management, Collaborative Tools, Testing Services) externally, not encapsulating all functionality into a monolithic core. However, they take quite different approaches to this sourcing.

PLS utilises a standards-based API (based on ADL SCORM 1.1) to interface with other compliant systems. In this way it can integrate with a Learner Management System (LMS) and pass user and assessment information back and forth between the systems. The PLS service is based on the notion that an adaptive content provider should be a service provider rather than a repository for extraction of content. Communication between PLS and a learning portal (or LMS) is achieved by enhancing the SCORM Runtime Communication API as used in SCORM v1.1.



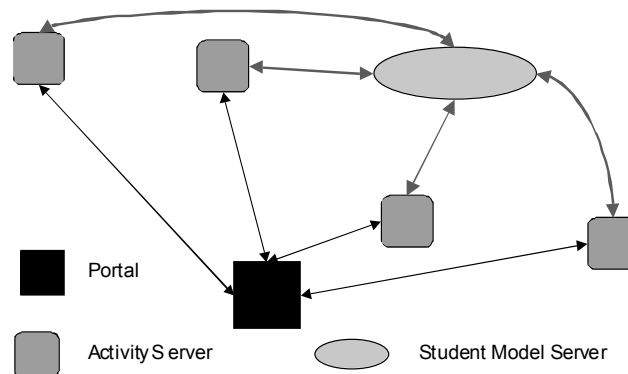
**Figure 1. Learning Portal and Adaptive Service Interface**

This requires a modification to the HTML frame layout for the PLS to enable calls to API functions residing on the LMS from PLS content. The actual API calls used are the same as those used in SCORM v1.1 as the API is designed to get and set values that are separately defined by an external data model. The remote PLS calls the Content Interworking API to access the data model on the Learning Portal (or LMS).

The learning content (visible on the learners screen) and JavaScript API (via a hidden browser frame) are delivered to the learner's browser. An API function, (which is in the hidden frame) is called from the content frame e.g. `LMSGetValue("cmi.core.lesson_status")`). The hidden API frame then communicates the request to the Learning Portal (or LMS). The Learning Portal returns the value (in this case of `cmi.core.lesson_status`) to the API Frame. The API function returns the value to content frame from which it may be passed back to the Adaptive Hypermedia Service (Figure 1).

Using these services, the deep complexity of the various meta data models (content, narrative, user) are simplified. The exported information model of the learner (and her performance) is made available via the API. The modified SCORM v1.1 interface facilitates integration with IMS and SCORM Compliant LMSs with only very minor adjustment of the information model passed between the Learning Portal (LMS) and the Adaptive Content Service. There is no change to the actual API function signatures [Conlan et al, 01].

Both PLS and the LMS maintain their own user information repositories, however. This approach was adopted in PLS as much of the user information and the way it is characterised is specific to the forms of adaptivity it is applying. In many regards the user modelling is one of the main distinguishing features of PLS. Some aspects of PLSs user model may be shared given a common vocabulary and framework, however others may still remain proprietary.



**Figure 2: Main components of the KnowledgeTree distributed architecture.**

The goal of KnowledgeTree [Brusilovsky, 02] is to replace the current monolithic course management systems (CMS) such as Blackboard or WebCT with a community of communicating servers. The architecture anticipates the presence of at least three kinds of servers: activity servers, learning portals, and student model servers (Figure 2). A *learning portal* plays a role similar to modern CMS. It allows a teacher to design a course and manages the student interaction with the course. The difference with CMS is that the learning content (activities) resides not in the portal, but in multiple distributed *activity servers*. An activity server plays a role similar to an educational repository in the sense that it hosts some (usually specialized) learning content. Unlike repositories that are essentially pools for storing learning materials that can be copied and inserted into courses, an activity server is responsible for both storing, and delivering *learning activities*. A portal has an ability to query activity servers for relevant activities and launch remote activities selected by students. An activity server is able to inform portals about available activities and provide a complete support for a student working with one of its activities. *Student model server* collects data about student performance from each portal and each activity server that work with a student. In exchange, it provides information about the student that can be used by adaptive activity servers to personalize their communication with the student. The presence of multiple adaptive activities requires a centralized user modelling architecture.

KnowledgeTree considers the standards-based model as not appropriate for adaptive distributed content and argues for a 3-component model (portal – content – student model server). PLS is structured to work within existing courseware management systems (CMS) that are completely static and thus consider adaptive services to be the main providers of adaptivity. It is assumed that that adaptive selection and structuring of content can only be done by a service. In contrast, KnowledgeTree allows for different kinds of portals – some can be as static as existing CMS, but some can be adaptive. In this vision, an adaptive portal can provide different adaptive support such as, for example, as adaptively selecting the best of existing static or adaptive content and adaptively arranging it for the student.

### 3.2 Adaptive Navigation Techniques

From a HCI perspective both AHA and Interbook use visual aids to indicate that different material is available to the user. In AHA [De Bra, Calvi, 98] [De Bra, Calvi, 97], the user model is constructed of user preferences indicated by the user directly, user knowledge initialised by stereotype and an overlay

model which consists of boolean variables - true if a concept is known and false if the concept is not known. Adaptive navigation is implemented by links with three possible states - desired, undesired and uninteresting. The standard colours of WWW browsers are used - blue links are desired, purple links are uninteresting which implies the information has been visited and does not represent new information to be learned and dark grey links indicate undesired information for which prerequisites have not been covered.

Adaptive navigation in Interbook is implemented by link annotation using checkmarks and coloured balls. Annotation refers to adding information to a link so that the user has more of an idea of where the link will lead and whether it complies with the current objective of the user. Link annotation allows the user to be advised as to the degree of relevance the system applies to a link on the basis of the user model, and the user may then choose their own path. A link can have a number of different states the values of which may be displayed to the user by colour, icons, or font formats.

World Wide Web browsers currently use link states with two values - visited links (the default for which is purple) and unvisited links (the default for which is blue). Adaptive Hypermedia Systems can extend this idea to show links with three states to signify concepts that are learned, well-learned or unknown. Links with up to six states have been implemented in Adaptive Hypermedia systems: visited, unvisited, current, suggested [Eklund, Sawers 98]. The link may be changed to a light colour to suggest that the link is dimmed - this gives the effect of hiding without restricting the user. Annotation gives the user a degree of freedom and supports stable ordering and the formation of correct mental maps [Eklund et al 97].

In Interbook green ball and bold text designates the state of the link as ready and recommended which implies that all prerequisites are well-learned or at least learned. A red ball and italic text designates a link that connects to content that is not ready to be learned which suggests that some prerequisites are not yet learned. A white ball is used when the link connects to information that is not new implying that all outcome concepts are learned or well-learned. Checkmarks by a link show that a link has already been visited. If the link is small then the content behind the link is known implying that learning has started. A medium checkmark designates content that is learned, while a large checkmark indicates content that is well-learned.

The mechanisms used in AHA and Interbook for implementing adaptive navigation do not restrict the learners ability to follow a link, they indicate the suitability of the information for the learner. This principal may be employed in other facets of Smart Spaces, helping to guide the user without limiting their options. Other navigation support techniques include –

*Relevance* - Link adaptivity may require the system to decide on the relevance of certain sections of the course content to a particular user at a particular time. This decision is reached based on the information in the user model. As an example the decision may be based on the current objective or goal the system has inferred for the user. If a link connects to information which is not required to meet the current goal, the link may be marked as irrelevant. Similarly, the concept to which a link is connected may require knowledge of concepts that the user has not yet covered. These links may be marked as irrelevant at this time.

*Direct Guidance* is provided by the system deciding where the user should link to next and presenting the user with this option. This is also called curriculum sequencing as the system enforces a path

through the course. This path is customised for that particular user but the advantages of Hypermedia are lost when the user cannot organise their own learning through the Hyperspace. When link annotation and direct guidance are offered together, users who are not confident of their ability to work through the course independently are more likely to click on the next button and accept direct guidance [Eklund, Brusilovsky 98].

*Link Ordering* is when the system sorts a list of links according to their relevance to the user. The system filters the links on the basis of the user model and presents the list with the most relevant links displayed at the top. This type of link adaptivity is often used for indexes or table of contents. A user who is inexperienced with the content of the course or with Hyperspace generally can be disoriented by a link order which is unstable.

*Link Hiding* restricts the navigational choice offered to a user. The system decides what links are not relevant to the user and changes the format to that of regular text so that the link is not displayed as a link. The link may be removed completely so that the user cannot access it even accidentally [De Bra, Calvi 98]. Link hiding can reduce the cognitive load on the user and conceal the complexity of a course while supporting the stable ordering of links. However, the usability of link hiding is questionable for a number of reasons. Users do not like to be restricted. There is a danger that the user will form an incorrect mental map of the Hyperspace. A sense of completion of the course will be difficult to attain when the user cannot be confident that all the links have been displayed.

### **3.3 Adaptive Presentation Techniques**

When considering the presentation of information to the user both AHA and PLS utilise fine-grained pieces of content. In AHA adaptive presentation is implemented through conditional inclusion of fragments depending on the user model. Alternative presentation and hiding of text is also used depending on the inferred knowledge of the user. The system will switch to verbose mode for a novice user who has just started to use the system. A concept explanation is included in the current page if it is a prerequisite concept for that page which has not been covered. Similarly if a technical term has not been covered a substitute will be used. For example the word page is used instead of the term node until the description of a node is encountered by the user and marked as encountered in the user model.

PLS also uses page fragments, called pagelets. The metadata driven multi-model approach, upon which PLS is based, has a minimum of three models – Content, Narrative and Learner. The Content Model contains metadata descriptions of the actual small size learning objects (pagelets). The Narrative Model only refers to concepts which may be selected as part of a course. There is no direct reference between the narrative model and actual content. This is the primary point in which PLS and AHA differ. In AHA the logic governing the adaptivity is coupled with the content. Indeed, as the logic is primarily implemented in JavaScript the content and logic are an integral unit. With PLS, however, the mapping between narrative and content is performed at run-time by the service engine, which reconciles the metadata imperatives of the narrative model with the metadata of the content model maintaining a separation of logic and content. The narrative and content models are linked via a shared (or mapped) metadata vocabulary. This vocabulary used in PLS is an extension of the IEEE LOM and ADL SCORM. From a Smart Space perspective it is attractive to retain this separation as the service that delivers the expertise (or narrative in PLS terminology) may not be the same service from which the content is sourced. With the separation of the AHS into three discrete models – Learner, Content and

Narrative – PLS facilitates the maximum reuse of expertise (narrative model) and learning material (content model). Multiple candidates may be present in the system to instantiate the models. For example, there may be several narratives used to teach a single course. An appropriate narrative is selected based on the learner's preferred learning style. Similarly, appropriate content may be selected based on their preferred visual style.

### **3.4 Representing Expert Knowledge and Experience**

The problems faced by the application of ITSs in the real world is one of the main driving forces behind the development of authoring environments for construction of ITS. Again the reuse of learning objects across multiple ITS systems is limited if not at times impossible. Likewise the instructional strategies developed by the domain expert were even less reusable. For successful development the system has to have some mapping between the high level pedagogy described by the author and the low level intelligence to be interpreted by the ITS during delivery. REDEEM (Reusable Educational Design Environment and Engineering Methodology) focuses on authoring pedagogy rather than on helping instructors create domain material. It offers a compromise between the static but simple CBT and the dynamic, albeit complex, ITS. REDEEM and PLS are some of the new generation ITSs and authoring tools aimed at creating ITSs in ways that require less effort, require less training and knowledge, provide help for authors to articulate their knowledge, support good practice or enable rapid prototyping [Murray, 99]. REDEEM's main goals are to: a) allow classroom teachers and trainers to construct ITSs in a reasonable timeframe; b) support reuse of existing material; c) focus on the authoring of pedagogy; and d) exploit the symbiotic relationship between psychology and reusable ITSs – using research to inform the design of ITSs, which can then be used to test the theories embedded within it, which in turn can inform developing theories of instruction and learning [Major et al, 97].

While not strictly an AHS REDEEM provides a means to creating ITSs that embody the teacher's experience and expertise, in a similar way to PLS.

## **4. Research Directions**

Some research directions that have been identified with relation to software adaptivity are:

One of the key areas of interest both in the Smart Spaces and with respect to eLearning specifically is the research area of service integration. Traditional LMSs are monolithic systems that attempt to provide all functionality at the core. The disadvantage of this approach is that often users (in this case learners) do not always want/need all of the functionality or the functionality provided does not fit their requirements. In this case allowing the user/learner to construct their own learning environment from a selection of services may be desirable.

Related to the issue of service integration is the research area of terminal adaptivity, i.e. adapting the content delivered to the platform it is being delivered on. The separation of content and rendering is part of the solution to this issue. Appropriate models of the target platforms, however, do not exist and will be a necessary component in the adaptation process.

On going research challenges include the appropriate representation of the learner/user in adaptive systems along with the other metadata models that impact on the adaptation process. With the potential for many distributed services gathering information about a user what appropriate mechanisms can be employed to ensure that the user is accurately represented and that as little duplication as possible occurs.

The relationship between expertise encapsulation and realisation of actual information requirements at runtime is also a valuable research direction. It would utilise any work done on vocabulary/ontology, but also help differentiate the roles in content management (content creation, knowledge sequencing, content/knowledge reconciliation, content delivery).

## 5. References

[Brusilovsky, 96] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), Spec. Iss. on Adaptive Hypertext and Hypermedia, User Modeling and User-Adapted Interaction 6 (2-3), 87-129.

[Brusilovsky, 02] Brusilovsky, P. and Nijhavan, H. (2002) A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, pp. 154-161.

[Conlan et al, 01] Conlan, O., Hockemeyer, C., Lefrere, P., Wade, V., & Albert, D. (2001). Extending educational metadata schemas to describe adaptive learning resources. In Hugh Davies, Yellowlees Douglas, & David G. Durand (eds.), Hypertext 01: Proceedings of the twelfth ACM Conference on Hypertext and Hypermedia, pp. 161-162, New York: Association of Computing Machinery (ACM).

[Conlan et al, 02] Conlan, O., Wade, V., Bruen, C., and Gargan, M.: Multi-model, metadata-driven approach to adaptive hypermedia services for personalized eLearning. In: De Bra, P., Brusilovsky, P. and Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002) Proceedings, Málaga, Spain (2002) 100-111.

[De Bra, Calvi 97] De Bra, P., Calvi, L. : Creating adaptive hyperdocuments for and on the web. - In: WebNet'97: proceedings of the WebNet'97 Conference, s.l., AACE Press, 1997, p. 149-155.

[De Bra, Calvi 98] De Bra P., Calvi L. : AHA: a generic adaptive hypermedia system. - In: Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, Pa, 1998, s.l., 1998, p. 5-12.

[De La Passardiere, Dufresne 92] de La Passardiere, B., Dufresne, D.: Adaptive Navigational Tools for Educational Hypermedia. ICCAL 1992: 555-567

[Eklund et al 97] Eklund, J., Brusilovsky, P., Schwarz, E. : A Study of Adaptive Link Annotation in Educational Hypermedia. <http://www.wis.win.tue.nl/ah/> 1997.

[Eklund, Brusilovsky, 98] Eklund, J., Brusilovsky, P., "The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence", <http://www.wis.win.tue.nl/ah/>, 1998.

- [Eklund, Sawers 98] Eklund, J., Sawers, J. : Customizing Web-Based Course Delivery in WEST with Adaptive Navigation Support. <http://wwwis.win.tue.nl/ah/> 1998.
- [Espinoza, Hook 95] Espinoza, F., Hook, K. : An Interactive WWW Interface to an Adaptive Information System. In proceedings of the User Modelling for Information Filtering on the World Wide Web, a mini-workshop at the Fifth International Conference on User Modelling. <http://wwwis.win.tue.nl/ah/> 1995.
- [Fink et al 96] Fink, J., Kobsa, A., Nill, A. : User-Oriented Adaptivity and Adaptability in the AVANTI project. <http://wwwis.win.tue.nl/ah/> 1996.
- [Kay, Kummerfeld 95] Kay, J., Kummerfeld, R. : User model based filtering and customisation of web pages. <http://wwwis.win.tue.nl/ah/>
- [Laurillard, 93] Laurillard, D., "Rethinking University Teaching: A Framework for the Effective Use of Educational Technology", Routledge & Kegan, P., 1993.
- [Murray, 99] Murray, T. Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.
- [Major et al, 97] Major, N., Ainsworth, S. & Wood, D. (1997). REDEEM: Exploiting Symbiosis Between Psychology and Authoring Environments. *International Journal of Artificial Intelligence in Education*, 8, 317-340.
- [Specht, 98] Specht, M., "Empirical Evaluation of Adaptive Annotation in Hypermedia" in *Proceedings of EdMedia*, 1998.

## State of the Art: Policy Techniques for Adaptive Management of Smart Spaces

Kevin Carey, Kevin Feeney, Dave Lewis  
Knowledge and Data Engineering Group  
Trinity College Dublin

### 1. Introduction

Policies represent an important existing approach to providing runtime flexibility in the operation of management components and systems. Smart Space will require management systems that are highly adaptive and which can accept modification to their behaviour at runtime from a number of appropriately authorised roles. Policies will therefore be a key mechanism in providing the runtime configuration of component behaviour needed in any adaptive smart space management system. This paper aims to give a brief overview of the state of the art in the structure of policy languages and their application, the latter with a focus on access control and management of quality of service.

### 2. Overview

A policy is a rule that can be used to change the behaviour of a system. They can be considered as declarations of business rules that an organisation wishes to apply to the operation of its systems. In general policies are expressed in terms of an *event* which triggers the evaluation of a policy rule, a set of *conditions* that must be met for a policy rule to be enacted and a set of *actions* that are performed upon such enactment. A policy management system is tasked with interpreting policies to enact behaviour on a set of devices. Policy events are mapped to the requests made on those devices or specific state change events, conditions are mapped to specific device states and actions to specific device operations. As policies are declarative and interpreted by policy management systems, they can be updated at runtime to flexibly control the behaviour systems. Policies are therefore being increasingly widely used in a variety of network and system management applications to provide an element of adaptability and run-time configurability in the behaviour of networks and information systems. Policies are useful in applying a common set of operational rules to a large set of distributed managed nodes and/or to an information system with a large set of users. The ultimate aim of policy based system is to derive policies from business goals, so that the operation of an organisation's systems can respond dynamically to changes in those goals.

Policy based management rests on the assumption that sets of policies can be applied to classes of devices, users or services. This allows easier management by grouping individual units into classes with common requirements and obligations with respect to the overall system while still retaining the ability to exercise fine-grained control over the choices in the behaviour of a system. Policies have many areas of application including the following. When policies are applied to network bandwidth and routing, they provide a mechanism for specifying Quality of Service (QoS) rules for classes of service (Lymberopoulos 2002). When policies are applied to users, they provide a mechanism for specifying access control for classes of user (roles), known as role based access control (RBAC) (Sandhu et al 1996). When policies are applied to nodes they provide a means for distributed configuration management (Crane 1995). By enabling decisions to be made closer to where the event and condition are detected, policies allow a less centralized and more flexible management

architecture, which could be a particularly important feature in the complex and heterogeneous environment of Smart Spaces.

### 3. Analysis

This analysis first addresses the structure and capabilities of various policy languages and the challenges presented by developing policies in these languages, e.g. conflict between different policy rules addressing the same resource. An understanding of the expressiveness and limitation of policies is required if they are to be used in engineering Smart Space management systems. The section then goes on to analyse the application of policies in role-based access control and resource management. Smart space management requires multiple people to monitor and exert control over resources, such as bandwidth, display real-estate and application services, without the benefit of boundary-based grouping of resources or people. A finer grained and more flexible mechanism is therefore required to control access to resources, for which role-based access control using policies is a strong candidate. The process of managing and sharing resources in smart spaces must react quickly to changes in the physical and system environment and so the intelligence needs to perform such reactive adaptation must be placed as close to the resource as possible. Policy-based resource management provides a mechanism for achieving this, and we examine the state of the art with respect to network resource management and its relationship to delivering quality of service to users.

#### 3.1 Policy Languages

There are a number of approaches to the definition of policies, and accompanying policy languages, which represent a number of different levels of policy expressiveness and policy enactment semantics. There is therefore no single accepted policy languages, with many languages being proprietary in nature and tied to particular system management products. Policy languages broadly split between ones addressing access control and ones addressing resource management (Sloman and Lupu 2002). The most widely accepted standardised model for resource management is the joint policy model from the Internet Engineering Taskforce (IETF) (Moore et al 2001) and Distributed Management Task Force (DMTF) (Rafalow 2002), which has been adopted by both organisations for Internet and enterprise management applications. PONDER is one of the more expressive policy languages addressing both access control (i.e. authorisation policies) and management (i.e. obligation policies). PONDER supports the notion of domain membership for applying policies to specific subject and target objects. Authorisation policies define which subject groups are authorised to perform which actions of which target groups. Obligation policies define under which condition a subject object must perform on certain actions on a target object. Figure 1 gives some example of authorisation policies in PONDER. Work with PONDER (Lupu and Sloman 1999) has highlighted some of the broader engineering problems with policies such as the sort of conflicts that can arise through its use.

```
inst auth+ switchProfileOps {
subject /NetworkAdmin ;
target <ProfileT> /Nregion/switches ;
action load(), remove(), enable(), disable() ;}
Members of the NetworkAdmin domain are authorised to load, remove, enable or disable objects of type ProfileT in the Nregion/switches domain.
```

```
inst auth- /negativeAuth/testRouters {
subject /testEngineers/trainee ;
action performance_test() ;
target /routers ;}
Trainee test engineers are forbidden to perform performance tests on routers. The policy is stored within the /negativeAuth domain.
```

**Figure 1: Example of Policies in PONDER**

Conflict detection is a crucial area if policies are to be used on any scale. Conflicts can be modal conflicts, for instance where a positive and negative authorization apply to the same objects, or application specific conflicts related to the semantics of the resources and roles in the target and subject domains of policies. Specification-time conflict detection is important since different people in an organisation may author different policies at different times and because policies are typically interpreted at run-time, computationally expensive conflict detection at that point would be impractical. Detection of overlaps between the tuple set of subjects, actions and targets for a set of policies helps detect modal conflicts, and if one tuple set is more specific in one of those domains than another, its policy is given precedence. Meta-policies are policies about policies, and may be used to detect some semantic conflicts between policies and to guide the resolution of those conflicts. Overall, resolving policy conflicts is an unsolved problem, largely still relying on manual policy-rewriting, including the redefinition of subject and target domains.

The application of policy to role based access control and QoS management is discussed in more detail below.

### **3.2 Policy and Role Based Access Control**

Most research in the area of Smart Space Management has focused on providing a limited number of wireless services to a clearly defined user base in a specific environment. For example, the provision of location-aware multimedia content to visitors to museums (Semper and Spasojevic 2002), or providing a small number of location-aware services to students on a college campus (William et al 2002). In general these projects can make assumptions about access control and security that are not applicable to envisaged real world smart spaces since:

They assume a single role, i.e. every user has the same rights and privileges as the others with respect to the smart space services. Roles are a means of modelling classes of user in terms of what the user wants or needs to do.

The smart space services only apply to particular mobile computing devices, which are configured by the smart space managers and distributed to users. Thus there is no variability in terms of potential devices within the space.

There is a limited number of smart space enabled services which do not vary dynamically.

For these reasons, many of the current smart space research projects have been able to adopt relatively simple security and access control models, which can rely upon the fact that the mobile devices and users which utilize the smart space services are pre-defined and static. However, there are several current projects that have started to look at the type of problems associated with more universal and interoperable smart space management, such as GLOSS (Gloss 2001) and OXYGEN (Oxygen 2002).

When attempting to devise general security models for real world smart space management systems, there is a need to accommodate a much greater diversity in terms of the users, devices and services available. This can be done with traditional Access Control Lists (ACL's), whereby each resource is associated with a particular ACL and only users included in the ACL have access to that resource. Policies provide several advantages over traditional ACL's since they are much more expressive and several powerful policy description languages exist (eg PONDER), and they allow for much more

convenient distributed management. In addition there are a number of very good reasons, specific to smart spaces, for using policies and role based access control in any smart space security model.

Firstly, it is almost always the case that access control rights are not randomly variable with respect to individuals, but are clustered around certain classes of user or roles. For example ordinary users are commonly granted certain rights, to access printers, network resources, etc... Other classes of users such as administrators are granted certain extra privileges. In general rights are related to the role of a user rather than to the individual identity of the user, thus it is much more convenient to manage these rights with respect to roles rather than on an individual basis, especially when most systems will have far more users than roles defined on them. This is the traditional justification for policy and role based access control.

The majority of RBAC research has focused on defining models that allow decentralisation of role creation and administration. One of the most influential initiatives is the ARBAC 97 (Sandhu et al 1997) (Administrative Role Based Access Control) family of models which proposes 4 different types of RBAC models with special administration roles which allow creation of further roles. However this approach is problematic for dynamic collaborative environments since there is no guarantee that participants will be able to fill pre-defined administrative roles (Crook et al 2002). The type of RBAC model used also depends on the policy language. Some languages, like PONDER are very expressive, including obligation, refrain and delegation policies alongside environmental role activation constraints, while others opt for simplicity in pursuit of comprehensibility, like OASIS (Yao et al 2001).

In smart spaces, a user's access rights and obligations towards resources is not necessarily static. Access rights can vary due to context. For example a user may have different rights depending on what other users are present in the space. This dependence of rights on context presents a difficult problem for traditional ACL mechanisms. However, when using policies and RBAC, these contextual changes can be easily modeled as role changes. The roles available to a particular user can change according to the context. Therefore, as long as we can manage this type of role variability, the access control system does not need to be changed in any way to take account of context – a significant advantage – however, the question of role mapping with changes in context remains a difficult problem and little work has been done in the area.

Furthermore, smart spaces present particular problems to traditional authentication systems. Smart spaces can be composed of a variety of services, some of which will not be particularly sensitive from a security point of view, while others may require high levels of confidence in the authentication of users. Ideally we would like to allow users to authenticate themselves to the smart space only to the required level for those services which they will use during their presence in the smart space. By associating different types of authentication with different roles and allowing users to choose which roles to activate upon entering the smart space, we can provide a seamless, integrated means by which a user can authenticate herself to the smart space to the minimum required level. For example, a typical user might be able to authenticate herself to a space by means of an active badge (Want 1992), but if the user wants to activate her 'administrator' role, a password or biometric identification mechanism may be required. Thus variable levels of authentication can be associated with different roles in a seamless manner.

The Gaia project (Roman and Campbell 2000) at the University of Illinois has adopted a powerful model for security and access control using roles, policies and different authentication methods with variable confidence levels among other measures. Gaia uses credentials, similar to Kerberos tickets, as a basis for authentication and access control (Viswanathan et al 2001). In Gaia, all resources in the system are associated with policies. Users present credentials to the system which are compared with policies in order to validate particular resources. Credentials are closely linked to roles. Users can activate any subset of their valid roles and receive credentials associated with these roles, which then allow them to access resources. This work demonstrates the flexibility and power of policies and role based access control in the realm of smart spaces. However, it is a very new research area and much work remains to be done.

Projects like Gaia show how the latest techniques in access control mechanisms can be very useful in solving some of the particular problems associated with Smart Spaces. Roles, in conjunction with policies, provide a convenient mapping for access rights and authentication methods that vary with context. However, there are still several areas that have yet to be addressed in the realm of Access Control for smart spaces. Policy refinement, role delegation, collaborative management and management of shared resources are all areas that have yet to be touched on in the smart space context.

### **3.3 Policy-based Resource Management**

To manage a smart space with all its devices and their interactions, a Policy-based QoS management system is considered in order to assure best end to end QoS for the individual user as well as to the smart space group.

IETF and DMTF have jointly produced a set of standards on policy and policy-based management systems. The two main elements in their model of a Policy-based management system are Policy Decision Point (PDP), a logical entity that makes policy decisions for itself or for other network elements that request such decisions; and Policy Enforcement Point (PEP), a logical entity that enforces policy decisions (Westerinen 2001). PDP is likely to store its policies in a repository, such as a Lightweight Directory Access Protocol (LDAP) directory service.

The basic interaction between the components begins with the PEP. The PEP will receive a notification or a message that requires a policy decision. Given such an event, the PEP then formulates a request for a policy decision and sends it to the PDP. The PDP returns the policy decision and the PEP then enforces the policy decision by appropriately accepting or denying the request (Yavatkar 2000). Common Open Policy Service (COPS) (Fernandez 2001) can be used as a policy transaction protocol between the PDP and PEP for transporting the policy requests and decisions.

Optionally, there can be a local PDP (LPDP) in each network domain and a high level PDP for the overall the network. PEP will first use the LPDP to reach a local decision. This local decision and the original policy request are next sent to the PDP, which renders a final decision for the good of the overall network (Yavatkar 2000). This way gives a better scalability to the policy management system.

One of the main application areas of policy research in network management is the management of Quality of Service guarantees. According to (Ponnappan 2002), there are two models used in Policy Based Network (PBN) for policy management, namely outsourcing and provisioning. In outsourcing model, when PEP has to make a decision regarding an event, it outsources the decision-making to the

PDP. It is typically used with RSVP requests, where PDP receives a request from PEP for an admission control decision regarding an RSVP request for resource reservation. Based on high-level policies found in the repository, PDP either allows or deny a RSVP signalled packet to enter the network. If using COPS for the policy transaction, COPS-RSVP (Herzog et al 2000) is a specification of functionalities for this model.

Whereas in provisioning model, the PDP typically predicts future configuration needs, and proactively pre-provisions for them ahead of time. It is most commonly used for controlling network policy for non-signalled protocols, such as Diffserv, where PDP decides, based on various criteria, whether the newly added policy should be installed in the PEP by sending policy rules as configuration commands that the device can interpret, so that policy decisions are enforced by classifying data packets as they enter the network and processed accordingly. COPS specifies functionality COPS-PR (Chan et al 2001) suited for this model.

The policy server, in most cases, is the PDP but it might have other components to help in its decision-making, such as a bandwidth broker to manage the bandwidth usage information in the network. There would probably be a policy repository connected to the policy server to store the policy information and a policy management tool for creating/modifying/deleting the policies. Policy server might optionally contact an external/remote entity, such as Authentication, Authorization and Accounting (AAA) server to check whether a user is authorised to use the service requested in order to make its decision. E.g., the AAA server could be part of the ISP.

Policies are either created by an administrator using the policy management tool or they are translated from a Service Level Specifications (SLS) (LyMBERopoulos et al 2002). After validation and static conflict tests, the new policies are stored in the policy repository. High-level policies are mapped down to lower-level policies or device specific commands.

PDP is the heart of the system and its decision algorithm is vital for the system performance. There are several different type of algorithms developed, such as a bacterial/genetic algorithm (Marshall et al 2001) that autonomously removes policies that degrade its performance, or a fuzzy logic algorithm (Fernandez et al 2001) to offer better QoS in a Diffserv domain because of the uncertainty and inaccuracy characteristic of the data flow estimate. Also, the PDP can have an algorithm for an adaptive policy framework (LyMBERopoulos et al 2002) that dynamically changes the parameters of the QoS policies (high-level) at run-time or enable/disable policies from a pre-defined set. Still there is plenty of scope for more research on this area.

#### **4. Research Directions**

Currently Policy-based management suffers from fragmentation of approach, partly due to differences in semantics between access control policy languages and resource management policy languages. As a consequence there is no commonly accepted policy language and no common approach to the engineering of policy based systems. The validation of policies prior to their deployment is not a solved problem, with some policy conflicts still only detectable through unwanted runtime behaviour. The promise of policies being driven from business goals remains largely unfulfilled, with policy decomposition and hierarchical translation still presenting difficult challenges.

Some policy-related research directions that have been identified with relation to adaptive smart space management are:

- One of the strengths of policy based management is that it is essentially a decentralised activity, allowing different people to distribute their policies onto a distributed system. However this raises the problems of conflicts between policies, especially when generated by different people in different roles. Though the use of role and domain abstractions in languages like PONDER help constrain the policy conflict detection problem, a fuller conflict resolution approach may be, as suggested in (Lupu and Sloman 1999), may result from deriving policies from higher level user goals and use analysis of these goals to shape the definition of non-conflicting requirements. As adaptive Smart Space management required some model of user context and goals, this may integrate well with the generation of conflict free policies. Application specific conflict detection may also be aided by better semantic descriptions of subjects, targets and actions, using ontology-based mark-up languages such as DAML+OIL and DAML-S.
- Current policy-based management schemes assume that policies operate on a population of existing objects, with most applications addressing fine grained managed objects that directly represent resources to be managed. There seems however, little attention paid to the potential role of policies in providing flexibility to coarser grained object in a generic way, in particular COTS components. This could involve the vendor of a software component specifying the policy events, conditions and actions that may be combined to modify the behaviour of that component at run-time. The responsibility for ensuring the expression of policy semantics for a component and the correct behaviour of different policy combination within the components is delegated to the COTS developer, arguable the best placed person to perform this task. This approach also raises the issue of adaptive techniques that merge service composition of services with configuration of behaviour of the COTS components that offer those services.
- Projects like Gaia show how the latest techniques in access control mechanisms can be very useful in solving some of the particular problems associated with Smart Spaces. Roles, in conjunction with policies, provide a convenient mapping for access rights and authentication methods that vary with context. However, there are still several areas that have yet to be addressed in the realm of Access Control for smart spaces. Policy refinement, role delegation, collaborative management and management of shared resources are all areas that have yet to be touched on in the smart space context.
- There are also several areas that remain to be researched in the complex area of access rights and roles. There are several decidability problems relating to accumulation of roles and the attendant matrix of access rights. We may find mutually contradictory policies associated with different concurrently held roles, or we may have a combination of roles which gives unintended access rights to the holder of specific combinations. The analysis and resolution of these decidability problems is a difficult problem which has thusfar barely been touched upon by the research community.

## 5. References

Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., Smith, A.(2001), "COPS Usage for Policy Provisioning (COPS-PR)", *IETF RFC3084*, March 2001.

Crane, S., Dulay, N., Fossa, H., Magee, J., Sloman, M.(1995), "Configuration Management for Distributed Software Services", *Proc. IFIP Int. Symposium on Integrated Network Management (ISINM 95)*, Santa Barbara, Chapman Hall, May 1995, pp. 29-42.

Crook, R. Ince, D., Lin, L., Nuseibeh B. (2002), "Security Requirements Engineering: When Anti-requirements Hit the Fan" Security Requirements Group Department of Computing, The Open University Walton Hall, Milton Keynes.

Durham, D., (Ed.), J. Boyle, J., Cohen, R., Herzog, S., Rajan, R., Sastry, A., (2000), "The COPS (Common Open Policy Protocol) protocol", *IETF RFC2748*, January 2000.

Herzog, S. (Ed.), Boyle, J., Cohen, R., Durham, D., Rajan, R., Sastry, A.(2000), "COPS Usage for RSVP", *IETF RFC2749*, January 2000.

Fernandez, M.P.; Pedroza, Ade.C.R.; de Rezende, J.F., (2001) "QoS Provisioning across a Diffserv Domain using Policy-based Management", *Proceeding of Global Telecommunications Conference (GLOBECOM '01)*, IEEE , Volume: 4.

"Gloss", (2001), ([www.gloss.cs.strath.ac.uk](http://www.gloss.cs.strath.ac.uk)), Available:  
<http://www.gloss.cs.strath.ac.uk/project.html> (accessed: 2003, March 4).

Lupu, E., Sloman, M., (1999) "Conflicts in Policy Based Systems Management", *IEEE Transactions on Software Engineering*, Vol.25, No.8, November/December, pp 852-869.

Lymberopoulos, L., E. Lupu, E., Sloman, M. (2002), "An Adaptive Policy Based Management Framework for Differentiated Services Networks", *Proc. 3<sup>rd</sup> IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2002)*, Monterey, California, June 2002, pp147-158.

Marshall, I.W.; Gharib, H.; Hardwicke, J.; Roadknight, C., (2001) "A Novel Architecture for Active Service Management", *International Symposium on Integrated Network Management (IM'01)* IEEE/IFIP.

Moore, B., Rafalow, L, Ramberg, Y., Snir, Y., Strassner, J., Westerinen, A., Chadha, R., Brunner, M., Cohen, R. (2001), "Policy Core Information Model Extensions", Internet Draft <draft-ietf-policy-pcim-ext-01.txt>.

"Oxygen Project, Pervasive, Human-Centred Computing" (2002), ([oxygen.lcs.mit.edu](http://oxygen.lcs.mit.edu)), Available at: <http://oxygen.lcs.mit.edu/index.html> (accessed: 2003, March 4)

Ponnappan, A.; Lingjia Yang; Pillai, R.; Braun, P.(2002), "A Policy based QoS Management System for the InServ/Diffserv Based Internet", *Proceedings of Third International Workshop on Policies for Distributed Systems and Networks, 2002*.

Rafalow, L. (2002), "CIM Policy Model Whitepaper", DMTF document DSP0108, v2.6.0, Available at: <http://www.dmtf.org/standards/documents/CIM/DSP0108.pdf>

- Román, M., Campbell, H. (2000), "GAIA: Enabling Active Spaces", *Proceeding of 9th ACM SIGOPS European Workshop*, September 17th-20th, 2000. Kolding, Denmark, pp. 229-234,.
- Sandhu, R., Coyne, E., Feinstein, H., Charles Youman, C., (1996) "Role-Based Access Control Models" *Laboratory for Information Security Technology, George Mason University, , IEEE Computer*, Volume 29, Number 2.
- Sandhu, R., Bhamidapati, V., Munawer Q. (1999), "The ARBAC97 Model for Role- Based Administration of Roles", *ACM Transactions on Information and System Security*, Vol. 2, No. 1.
- Semper, R., Spasojevic, M., (2002) "Exploratorium - The Electronic Guidebook: Using Portable Devices and a Wireless Web-based Network to Extend the Museum Experience - Overview and Findings to Date", *HP Labs. Paper presented at Museums and the Web Conference*, April 2002
- Sloman, M., Lupu, E., (2002) "Security and Management Policy Specification", *IEEE Network*, vol.16 No. 2, March/April 2002, pp.10-19.
- Viswanathan, P., Gill, B., Campbell, R.H. (2001) "Security Architecture in Gaia", *Technical Report UIUCDCS-R-2001-2215 UILU-ENG-2001-1720*, University of Illinois at Urbana-Champaign.
- Want, R., Hopper, A., Falcao, V., Gibbons, J. (1992), "The Active Badge Location System", *ACM Transactions on Information Systems*, Vol. 10, No. 1, January 1992, pp 91-102
- Westerinen A. et al. (2001), "Terminology for Policy Based Management", IETF RFC3198.
- Griswold, W.G., Boyer, R., Brown, S.W., Truong, T.M., Bhasker, E., Jay, G.R. Shapiro, R.B.(2002), "ActiveCampus - Sustaining Educational Communities through Mobile Technology?", *UCSD CSE technical report #CS2002-0714*
- Yao, W., Moody, K., Bacon, J. (2001), "A Model of OASIS Role Based Access Control and its Support for Active Security", *Proceedings of SACMAT'01*, May 34, 2001, Chantilly, Virginia, USA.
- Yavatkar, R. Pendarakis, D. Guerin, R. (2000), "A Framework for Policy-based Admission Control", *IETF RFC2753*.

## Software Infrastructure

Though M-Zones will be conducting research into adaptive and policy-based techniques for performing adaptive smart space management, this can only achieve results if supported by a clear view of common software services and capabilities that will be available for application developers. We call this set of common software services, technologies and capabilities the software infrastructure. In any particular domain of computing, the software infrastructure is constantly developing, with new functionality being added, often at higher levels of abstraction than before. A good indication that some piece of technology is accepted widely enough to become part of the 'infrastructure' is when associated interoperability standard emerges. Sometimes the emergence of such standard is the catalyst for widespread uptake and 'infrastructuralisation' of a technology. This section reviews aspects of the state of the art in the software infrastructure that appears to be in place for ubiquitous computing. M-Zones moves forward with the assumption that such an infrastructure will form the basis of our research into adaptive smart space management, but also with the expectation that technologies that are at the research and development stage now will become part of the software infrastructure as the project progresses. Equally, improvement of the infrastructure will be an active research area in M-Zones.

This section addresses software infrastructure through three state of the art whitepapers:

- Context Management by Keara Barrett (WIT) and Ruaidhri Power (TCD)
- Service Composition by Steffen Hignels and David Lewis (TCD)
- Middleware by Ray Carroll (WIT), Sinead Cummins (CIT), Fergus O'Reilly (CIT) and Jason Finnegan (WIT)

Context information is important both to enable ubiquitous computing systems to adapt to a changing physical and computing environment and to support them in understanding changing user needs and the best corresponding course of action to take in any particular situation. Context information includes computing context, the user context and the physical context as well as historical information about these. As any form of adaptive smart space management system will need access to context information, inclusion of context management capabilities into the infrastructure is a sound architectural decision. Context management is already in place in a rudimentary sense, for example in the way desktop office application use information on past and current activities to provide context sensitive help or in the way web browsers and servers collaborate to detect if someone has visited a web page before. Much work to date in relation to ubiquitous computing has focussed on location detection as a primary source of context information, however the Context Management white paper points out that a much broader view of context information is required if context management is to be offered as an infrastructural service. This will require standardisation of context information through adoption of common models and service for manipulating context information, rather than the wide range of highly application dependent models and service currently available. This will require some commonality in how context data is collected from sensors, applications and other parts of the infrastructure, but also strong privacy control mechanisms to provide people with the confidence to allow personal context information to be shared with encountered applications and ubicomp environments.

A core assumption that is essential to developing adaptive system that can dynamically integrate components from multiple developers and operated in multiple organisational domains, is that components only interact through well defined interfaces. This is known as a service-oriented approach, where such well defined interfaces, or services, are reflective, i.e. their capabilities can be examined at run-time. The level of expressiveness and semantics used in a reflective service

description determines the level of automated decision making that can be performed by an adaptive system. The assembly of services into a more highly functional aggregate service is called service composition, and a principle aim of adaptive smart space management is to perform service composition in as dynamic and seamless a manner as possible. The Service Composition white paper discusses the various activities that are involved in service composition, including the description of services at various levels of abstraction, the advertisement of services and the use of brokers to locate services that best fit requirements and trust requirements for using foreign services. The automated composition of services is still a very open research area and centres on the combination of various reasoning techniques with common mechanisms for expressing semantics via ontologies. Standardisation of ontology language is underway in ISO (Topic Maps) and W2C (based on DAML+OIL). Building on these DAML-S utilises concepts from workflow and web service flow languages to provide a semantic service description and composition language.

Underlying application development in any distributed computing environment is the concept of middleware, which aims to provide the application programmer with a portable API and a set of common services. Middleware systems and standards are very mature, with CORBA, Sun's Java RMI and Microsoft's .NET being the most widely used. A wide range of general purpose and application specific services are available for these platforms and interoperation solutions between platforms are widely available. In addition, most of these platforms now take a component-oriented approach to application to ensure ease of portability and deployment on different platforms. Middleware systems also typically support exposing interfaces as web services so that these services can be easily accessible over the Internet (most middleware platforms cannot negotiate unmodified firewalls). Despite this maturity the middleware area is developing fast along a number of axes described in the Middleware whitepaper, which may be useful for adaptive smart space management. One specific example is Jini, which is Java based middleware aimed at ad hoc communication between a wide range of personal, office and domestic devices. Another broader movement is that of intelligent mobile agents. These primarily take advantage of the code mobility offered by Java to allow active software components, called agents, to move around a network in order to accomplish some task. As agents will have no a priori knowledge of other agents they will encounter and collaborate with in accomplishing tasks, there is quite an amount of research on agent communication mechanisms where capabilities are exchanged and mutual activities negotiated. This makes agent very applicable to the highly dynamic computing environments of smart spaces. Finally there are some standardised application specific APIs that may be relevant to the integration of smart spaces with other systems. Parlay is a good example, being an industry API intended to make management and control functionality of telephony systems available to other systems.

## State of the Art: Context Management

Keara Barrett  
Telecommunications Software Systems Group  
Department of Informatics, Mathematics and Physics  
Waterford Institute of Technology

Ruaidhri Power  
Knowledge and Data Engineering Group  
Department of Computer Science  
Trinity College Dublin

### 1. Introduction

The main objective of ubiquitous computing is to decrease the effort required to exploit computing to aid human activities.

“Ubiquitous computing has as its goal the enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user” (Weiser 1993).

Furthermore if the user wishes to extract the maximum benefit from the computing environment, the associated systems and services must cooperate and integrate their information and the general information about the situation in which the user wishes to carry out the task. This information about the user’s situation is called context. Humans have always used situational information, or context, to make inter-personal interactions richer. If computers use context while interacting with humans, they can offer more useful services and information to humans than is possible without the application of context. The main challenge with computers using context while interacting with humans, is that there is no standard, reusable model that can be used to handle context.

Computers are separated from the reality around them, limited to the explicit input that they receive from their environment. This can lead to differences between what the computer attempts to do and what the user wishes to do, because the computer may not be given enough information to recognise what the user aims to do. Giving computers more information about the world in which we live and work will enable them to assist in our daily lives. The field of context management attempts to address the difficulty of using context in computing by proposing context information that can be extracted from different computing situations. By supplying context information to applications involved, the user experience can be enhanced and new applications can be produced.

The main propose of this paper is to investigate the use and management of context in a system and to examine the components needed to create a proficient context management system. This paper introduces the notion of context, the element of a context-aware system and context management. Benefits and possible uses of context are described and an overview of some existing context-aware systems is given. The remainder of the paper is organised as follows. Section 2 presents an overview of both context and context-aware computing. In Section 3 context is classified, characteristics of context are outlined, elements and features of context-aware computing and context-aware systems are discussed and management models are outlined and compared. Section 3 also discusses some uses of

context, it examines privacy as it applies to context, and finally examples of context-aware systems are outlined. The paper concludes with Section 4, which talks about future work in the area of context management.

### **1.1 Relevance to M-Zones**

The main objective of the M-Zones project is to “undertake fundamental research into novel management infrastructures to enable collaboration and management, between and within Smart Spaces.” Smart Spaces are work environments with embedded computers, information appliances, and sensors allowing people to perform tasks efficiently by offering unprecedented levels of access to information and assistance from computers

One of the areas of research that is vital for the collaboration and management between and within Smart Spaces is the organisation of context information, which involves the gathering, interpretation, storage and dissemination of context information dynamically and in real-time. The organisation of context information, otherwise called context management, should allow for context-aware services in smart space environments. This should entail the seamless accessibility to context-aware services as the entity, be it a person or device, moves between different smart spaces.

The challenge for the M-Zone project is to examine context and the progress that has been made thus far in the field of context management to recognise the characteristics that a context management standard should include. When the characteristics have been identified they should be integrated into the overall inter-smart space management framework. At present no context management standard exists, so the M-Zones project should research the characteristics of context to recognise the essential aspects of context management.

The M-Zones project should also consider the dynamic exchange of context information between, private, semi private and public smart spaces administered by different parties, to identify the different constraints and effects that mobility and security have on the management of context information.

## **2. Overview**

Much debate has occurred and is still taking place about the meaning of both context and context-aware computing. Therefore one of the first steps in context management is to determine what information constitutes context. Schilit and Theimer, the pioneers of context-aware computing, regard context to be location, identities of nearby people and object, and changes to those objects. They consider where you are, whom you are with, and what resources are nearby to be the important aspects of context. Abowd et. al.’s more recent classification of context (Abowd, Dey, Orr, & Brotherlon 1997) expands the Schilit et al definition. They define context as:

“...any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.”

This means that any information that depicts the situation of a user can be entitled context. The temperature, the presence of another person, the nearby devices, the devices a user has at hand and the orientation of the user are examples.

By incorporating the “five W’s” into the examination of context we can clarify the scope and importance of context in pervasive computing. The questions to be asked include: What is context? Who might benefit from an awareness of their context, whose context is important to who, or what? Where can an awareness of context be exploited? When is context-awareness useful? Why are context-aware applications useful? (Morse, Armstrong & Dey 2000)

When humans talk directly to one another they are able to use implicit information about the situation, i.e. context, to enhance the conversational bandwidth. Unfortunately this implicit information does not transfer naturally to human-computer dialogue (Dey 2002). The concept behind context-aware computing is to exploit the progress in sensing and mechanisms for observing the environment to systematically collect implicit context (Abowd et al 2002).

Context information can be formed into an abstract model of all the actors in a smart space system. A system is context-aware if it uses the context information in the abstract model to provide relevant information and/or services to the user. Services consist of an application and the composition of that application for the task in hand in the system. Context-aware systems can make more informed decisions about information to be presented to users and how to react to commands received from users. According to Schilit and Theimer context-aware computing is any system that “adapts according to its location of use, the collection of nearby people and objects, as well as changes to those over time” (Schilit, Adams & Want 2002)

### 3. Analysis

#### 3.1 *Classes of Context*

Context is effective only when shared (Winograd 2001). To ensure context is shared, context must first be gathered and managed by a context-aware system. This implies that the context-aware system must understand what context is before it can go about seeking and categorising this information. Schilit et. al. propose the following classification of context information: (Schilit, Adams & Want 1994)

- *Computing Context* - network connectivity, bandwidth, and nearby resources such as printers, displays, or workstations.
- *User Context* - the user’s profile, location, nearby people, and current social situation.
- *Physical Context* - lighting, noise level, traffic conditions, temperature.

Each of these categories contains a wealth of information relevant to the context-aware system. They cannot however be used in isolation to full effect; computing context will be combined with user context, and user context will be combined with physical context to provide a full picture for the context-aware system.

##### 3.1.1 Physical and Virtual Context

Context as defined in section 2 may alternatively be subdivided into two categories: Physical Context and Virtual Context. The intention of a context-aware system is to gather both physical and virtual context and merge them together to achieve an overall picture of the situation. Virtual Context may include the version of the operating system, the interface capabilities, the wireless technology used to

accomplish communication, email messages sent and received, and documents edited. Physical context on the other hand may be the presence of another entity, be it a user or device, the proximity to a particular printer, information indicating if the user is standing, walking or sitting or the current weather conditions. When both physical and virtual context are unified a different reasoning about the user's situation may emerge. In the past the main focus was on physical context, with a particular emphasis on location, however there is now a move toward a vast range of possibilities. The Kimura system (Voids et al 2002) accumulates virtual context from the user's desktop using a monitoring system for Microsoft Windows. It manages this using the hooks feature exposed through the Win32 API (Voids 2002). Many systems exist for gathering physical context and especially location, for example the Active Badge System (Want, et al. 1992), the Cricket Location System (Priyantha, Chakraborty, & Balakrishnan 2000), the Radar System (Bahl and Padmanabhan 2000) and the EasyLiving computer vision system (Shafer, Brumitt & Meyers 2000). After the context is gathered and stored in a buffer or repository, a proficient system would merge them and decide what is relevant to the user at the present moment.

### 3.1.2 Context history

Historical context information, is where computing, user and physical contexts are stored across a time span. Potential uses for this stored information would be to establish patterns of smart space usage. Historical context information is particularly useful for mobile-aware applications. For example applications that predict resource consumption based on observed patterns of mobility and usage. Su et. al. describe a method of mobility prediction in (Su, Lee & Gerla 2002).

*Historical context is generally considered to be useful, but it is rarely used in current context-aware systems. Firstly, the context-aware system must decide what historical information is worthy of being kept, and at what level of precision. Evaluating all historical context information that is collected, to acquire the necessary information would be prohibitively costly, and efficient algorithms must be implemented to process the wealth of information available to extract meaningful data.*

## 3.2 Characteristics of context information

The characteristics of context are desirable when designing a context-aware system to ensure the context-aware system manages context effectively and efficiently. Researchers at the University of Queensland have classified four characteristics: (Candolin & Kari 2002)

### (1) Context Information Exhibits a Range of Temporal Characteristics

Context information has already been subdivided into physical and virtual information; it can be further subdivided into static and dynamic information. Static information is any information related to the user's environment that is invariant. Static information may be retrieved directly from the user. The vast majority of context information is dynamic. Dynamic information must be accumulated continuously, frequently and automatically. Additionally, past context information may be needed to understand the full state of the environment.

### (2) Context Information is Imperfect

This considers the validity of context, in particular dynamic context information. Reasons for the doubt over the soundness of the context arise due to the speed at which the context information changes, and the subsequent necessity to process this information before it is used to facilitate work carried out by the user. This "delay between the production and use of context information" (Candolin & Kari 2002) is a concern. Other sources of concern regarding the soundness of context information include the reliability of the producers of the context

information, for example the sensors and the possibility of a broken path between the producers of the context and the point where it is utilised by the user, which means the user may be using out of date information.

(3) Information has many Alternative Representations

There is a considerable difference between the raw data that is gathered from the environment, both virtual and physical, and the processed information that is used to assist the user. The raw data can take on many forms when combined with other context information and when processed by the context-aware system. The probability of the context-aware system obtaining a 100% success rate in “capturing the relationships that exist between the alternative representations” (Candolin & Kari 2002) of the context information and the one apt to the current situation is extremely low.

(4) Context Information is Highly Interrelated

Context information derived from a particular origin may have a very close link with its source, so much so that it is dependant on the origin. Context may not be reliable “where the characteristics of the derived information are intimately linked to the properties of the information it is derived from” (Candolin & Kari 2002).

### 3.3 Context-Aware System

A context-aware system must be capable of mimicking a human’s ability to recognise and exploit implicit information in the environment, in order to advance the operations of its functionalities. Although identifying and deducing a human activity is a challenge, it is critical that context-aware applications should operate by conveying the appropriate information to the right place at the right time through inferring the user’s intention. To accomplish this objective the context-aware systems must:

- (1) Gather the information from the environment or the user’s situation.
- (2) Translate this information into the appropriate format.
- (3) Combine context information to generate a higher context. A higher context is context information that is derived as a result of the merger of other context information.
- (4) Automatically take action based on the retrieved information.
- (5) Make the information accessible to the user, immediately, in the future, or when it is required, to enhance and aid in the completion of the user’s task.

The researchers of the Context Toolkit at the University of Berkeley propose that there are three categories that a context-aware application can support: (Dey 2000)

- (1) Presentation of information and services to a user
- (2) Automatic execution of a service for a user
- (3) Tagging of context to information to support later retrieval

The Kimura System integrates independent tools into a pervasive computing system. The developers of this system have designed distributed components of a context-aware system. These components fall into three classes (Voidsa et al 2002), these are:

- (1) Context Acquisition, the system gathers context information and adds it to a repository.
- (2) Context Interpretation, the system converts the gathered context information into a working context.
- (3) User Interaction, the system displays the working context to the user.

### 3.4 Elements of Context-aware Computing

The steps outlined in section 3.3 ought to be implemented by a context-aware application for an application to advance from a 'typical' application into a context-aware application. No model has been standardised for managing context, yet there is a necessity to design a model to optimise the benefits gained from employing context in the application. The management model should handle context in a reusable manner to permit context from one source to be exploited by many distinct applications that perform a variety of tasks.

Context-aware computing is a computing paradigm in which applications can discover and take advantage of contextual information such as location, time of day, people and devices, and user activity. Context-aware computing is especially suited to the areas of mobile and pervasive computing. Instead of adapting systems and applications so that mobility is hidden, context-aware computing provides support for mobile-aware applications.

*"Context-aware computing is the use of environmental characteristics such as the user's location, time, identity and activity to inform the computing device so that it may provide information to the user that is relevant to the current context."* (Burrell & Gay 2001)

Context-aware computing was pioneered by XEROX PARC and Olivetti Research Ltd. Context-aware computing involves many individual elements:

- *Sensor technology* - Numerous hardware devices must be equipped with the capability to collect information that will form part of the context of the system. These devices should be relatively inexpensive and readily available, and should hopefully require a minimum amount of configuration and management. They must also be capable of transmitting information to some central location, or else communicating with nearby devices.
- *Context model* - A model of the context information must be formulated to provide a resource for applications to avail of. Projects such as Cameleon (<http://giove.cnuce.cnr.it/cameleon.html>) are looking at constructing models of context information relevant to applications, thereby "endowing these versions with the ability to dynamically respond to changes in context such as network connectivity, user location and ambient sound and lighting conditions."
- *Decision systems* - Once the context model has been formed, elements of the system must make decisions based on the information available. These decisions can be made either at application level based on the information available to it, or can be made centrally and then disseminated to the individual agents in the system.
- *Application support* - Application programmers need to be aware of context information, as the addition of this information will fundamentally change how they work. Instead of being driven primarily by explicit user input as they have been in the past, applications will begin to 'act for themselves', but must do so in a way that attempts to adhere to the law of least surprise for the user. Ideally, user interaction with a context-aware application should be simpler and more productive than with a traditional application (Cheverst et al 2000).

### 3.5 Gathering context information

Some context information can be given to the context-aware system explicitly, such as a user's name or age; other context information can be obtained through the use of sensors. Many types of sensor are already commonly in existence and can provide primitive physical information such as light, heat and pressure readings. Other types of context such as facial recognition rely on fairly simple sensors such as cameras, but require considerable processing such as image recognition in order to make use of the information obtained.

Location and identity are the most frequently sensed pieces of context. Active Badges (Want et al 1992) produced by Olivetti and AT&T emit infrared signals which give a rough location and ID. Optical systems for context determination are also possible and research is underway in the areas of optical tracking and motion detection, stereo and 3D reconstruction and object recognition.

Location is an important element of context information. Many different approaches have been taken to determining the location of agents within a context-aware system. GPS, Infrared and radio signals have all been explored. Many context-aware systems that have been produced are *only* aware of location information. While these applications are useful, location is only one element of the wider context information. Schmidt et. al. discuss this fact in (Schmidt, Beigl & Gellersen 1999).

*Sensors are not always 100% accurate or reliable, particularly if they are disposable. The information gathering system must be tolerant of sensor failure, and any information gathered from sensors must be subjected to sanity checks to help verify its correctness. Sensor fusion is one method of avoiding this difficulty.*

Sensor fusion means aggregating the results of different sensors together to produce a reasonable approximation of the state of the system. This means that some sensors can fail or give erroneous answers, but the system will still be able to determine the real state through the use of a voting mechanism. When considering the output of temperature sensors for example, it might be prudent either to simply average the results or alternatively to discard reported values that differ too greatly from what other sensors report. This method would avoid drastic measures being taken by the context system to correct what it considers to be temperature variations but are actually simply the result of sensor failure. Gellersen, Schmidt & Beigl describe a project making use of sensor fusion in (Gellersen, Schmidt & Beigl 2002).

Sensor discovery is another issue: with the proliferation of sensors that will be required for a context system to be useful, configuring and managing a large number of sensors can be a difficult task. It is possible (perhaps even likely) that there would be more sensors than people in such a system, so being able to automate the detection and configuration of sensors would be necessary. The MUSE project (Castro & Muntz) uses Jini for automatic sensor detection.

### 3.6 Retrieving Context

Push and pull are the two options available to a context-aware system to extract the necessary context information from context sources. Context information is periodically sent to the application in the push model. This means that the context source collects the context information before it is needed, which may result in a better performance. The shortcoming of this approach is the consumption of resources for gathering and disseminating context that may never be exploited by the context service. The pull model in contrast gathers only the context information that is required by the service, however

this approach exposes the context service to network delays and unavailability. In the case of the push model, a trade off between freshness of context information with the cost of frequent updates must be measured (Ebling, Hunt & Lei 2001), while the pull approach must consider the trade off between reserving resource and the probability of faults.

### **3.7 Models for Managing Context**

*The first mandatory step in managing context is to supply the context-aware applications with a software component that offers access to context information. Three models that present a structured approach to providing context to an application are:*

- Context Widgets
- Infrastructure approach
- Blackboard

#### **3.7.1 Context Widgets**

The context widget, which is the most prevalent model, resides between the context-aware application and the environment, comparable to the way in which the GUI widget resides between the application and the user. The main objective of the widget is to separate the application from context acquisition issues. This separation hides the complexity of gathering and managing context information, thus the approach and problems related to accumulating the context has trivial or perhaps no impact on the application. In addition the widget is used as a mediator to pass only the pertinent information to the application, any other accumulated context information does not concern the application. Finally the context widget provides a reusable building block, which permits many applications to exploit context information that is detected by the sensors. A context widget functions independently of applications, which permits multiple applications to use it simultaneously. A context widget is also responsible for maintaining a complete history of the context acquired for the user's situation.

#### **3.7.2 Infrastructure Model**

This model is equivalent to the client-server model (Martinka 1996). It is a more flexible than the context widget, as it promotes the independence of the components in the context-aware system. Each component is empowered with the capability to perform all the required functionality, including establishing connections, organising input and output messages and managing faults, significantly increasing this model's complexity. This model supports a greater range of devices and applications by using standard coding and networking protocols. It also facilitates both independent and dynamic changes in the different components, be it the sensors, services or devices. Finally the third advantage of this model is the straightforward approach available for developing and deploying sensors, processing power, data and applications. (Hong & Landay 2001)

#### **3.7.3 Blackboard**

The blackboard model adopts a data-centric point of view. When applying the blackboard model the application "posts messages to a common shared message board, and can subscribe to receive messages matching a specified pattern that has been posted." (Dey & Abowd 2000) Many problems exist when all communication must traverse a central database (the common shared message board), for example the existence of a single point of failure.

### 3.7.4 Trade-off criteria

When deciding on the most apt model for a context-aware system it is vital to consider trade-off characteristics. The different dimensions that should be measured are efficiency, effort in configuring, robustness, simplicity and extensibility. (Winograd 2001)

#### **Efficiency**

The efficiency of the model depends on the bandwidth and latency. The objective of the model, in terms of efficiency, is to accelerate the throughput of information. It could be argued that efficiency should not be an issue with the present networking and processing speed, but when the explosion in the number of networked applications and devices are considered, it becomes apparent that efficiency is still a major concern.

#### **Effort in Configuring**

These context-aware systems comprise many components, so it is imperative that adding, removing or changing component is not a tedious task. Configuring individual components of the system must be achievable without the disruption or total failure of the system when possible.

#### **Robustness**

The context-aware system should be designed to continue to operate in a failure mode when components fail. When a failure mode is not possible, the system should fail in a controlled manner. The degree to which the system can cope with failure is termed its robustness.

#### **Simplicity**

It is important that the model is not excessively complex because “a system that requires complex understanding by system builders in order to make use of its facilities will be used only by those who have the dedication and motivation to master it.” (Winograd 2001)

#### **Extensibility**

The concept of context-aware computing is still looked upon as an emerging technology thus it is important that “services supporting a general notion of context must be easily extensible to accommodate new and unanticipated sources of context information.” (Ebling, Hunt & Lei 2001)

### 3.7.5 Comparisons

*The trade-offs described in section 3.7.4 can be used to compare and contrast the different models for context management. The widget model has tight coupling of system components, which may make it the most efficient model in some circumstances, however it suffers from complex configuration and issues with powerlessness in the cause of failure. The infrastructure model with its independent components may be extremely complex, which is a no factor for many developers, however its positive criteria of straightforward configuration and robustness may compensate for the negative implication of complexity. Finally the blackboard model with its very loosely coupled components may suffer from efficiency problems, but it is a simple, robust, easily configurable model. (Winograd 2001)*

Although the models assist in managing context, further abstractions are essential to handle context information effectively. These abstractions are detailed in section 3.8 to 3.10 inclusive. .

## 3.8 Interpreters

After sensing, acquiring and saving the context from various sources the next activity for the context-aware system is to produce a mechanism for achieving context interpretation, so that the gathered context information is utilised in a fitting manner. “Interpretation refers to the process of raising the level of abstraction of a piece of context.” (Dey & Abowd 2000) The interpretation may involve

integrating numerous contexts into one to provide a higher-level context, thus the interpreter alters context information by raising its level of abstraction.

One approach is to use context fusion to convert the “lower level context into higher level usable by applications”. (Mitchell 2002) Context fusion may be viewed in different ways. One could regard it to be the synthesis of context from the same type of sources in order to increase the validity of the information so that erroneous sensors or readings are detected to avoid improper decisions by the system. Context fusion is also deemed to be the aggregation of context of varying types from a different variety of sources to produce a context that is exploitable by the system.

The Technology for Enabling Awareness (TEA) project has produced a layered architecture for fusion. The layers include Sensors, Cues, Context, and Scripting. The Sensor layer has physical sensors for measuring physical factors in the environment and logical sensors for gathering host criteria. The Cues layer produces an abstraction from the physical and logical sensors. The Context layer derives a depiction of the situation from the abstractions in the Cues layer. Finally the Scripting layer “provides a mechanism to include context information in application” (Schmidt, Beigl & Gellersen 1998) The Scripting layer has three states: (1) entering a context, if a situation is specified with a probability that is higher than a threshold an action is performed after a certain time. (2) Leaving a context, if a situation is specified with a probability that is less than a threshold an action is performed after a certain time. (3) While in a context, if a situation is specified with a probability that is higher than a threshold an action is performed every specified time interval. (Schmidt, Beigl & Gellersen 1998) “Context fusion must handle seamlessly handing off sensing responsibility between boundaries of different context services. Negotiation and resolution strategies need to integrate information from competing context services when the same piece of context is concurrently provided by more than one service.” (Abowd & Mynatt 1999)

### **3.9 Aggregators**

“Aggregation refers to collecting multiple pieces of context information that are logically related into a common repository.” (Dey & Abowd 2000) This abolishes the need for the context-aware application to gather the required context from different sources that would otherwise be obligatory as a consequence of the distributed nature of the context-aware systems. Aggregators support the delivery of particular context to an application, by accumulating related context that the application seeks into one logical placement. An aggregator facilitates interpretation of context hence it will aggregate diverse context information for different requesting applications. Individual applications can be alerted to alterations in the aggregator’s context and can push or pull context from the aggregators. “Aggregators provide an additional separation of concerns between how context is acquired and how it is used.” (Dey & Abowd 2000)

### **3.10 Context Services**

The three management models outlined along with the two abstractions, interpretations and aggregators, described above deal primarily with the acquisition and manipulation of context information to provide the application with relevant information in an apt manner. If these components are implemented by a context-aware system three of the five steps, outlined in section 3.3, necessary in developing a proficient system will be realised.

The context-aware system will automatically take action based on the retrieved information. “Services are components ...that execute actions on behalf of applications,” (Dey & Abowd 2000) they use actuators to manipulate or change the state of the environment in response to a decision made by the application based on acquired context. Context service may be synchronous or asynchronous. An asynchronous context service “requires that the application waits for a response” (Ebling, Hunt & Lei 2001) but because there is no guarantee that a response will result, the application requesting the response will be notified that the service has been initiated and the service results will be delivered when they are available. (Dey & Abowd 2000) A synchronous context service requests that the application waits for a response.

### **3.11 Application support**

Chen and Kotz (Chen & Kotz 2000) divide the awareness that applications have of context information into two different types:

- *Active Context Awareness* - an application automatically adapts to discovered context, by changing its behavior.
- *Passive Context Awareness* - an application presents the new or updated context to an interested user or system. Alternatively the application makes the context persistent for the user or system to retrieve later.

Once context information has been obtained, the use of passive context information can be implemented in an application. Simply displaying the data from a sensor or the current location of a person would be use of passive context. The case of active awareness of context is the more challenging, as it requires fundamental changes in the way applications operate. While the user is somewhat ‘in control’ of the current generation of applications (requiring the user to initiate almost all action), there is the danger that users could start to feel helpless and out of control of systems and applications that begin to make decisions on their own because they have more information than the user does.

With context-aware systems, applications can exist that can adapt and need less explicit input from the user. For example, many devices could use context information to do things such as turning on and off depending on whether they are (or are likely to be) used, using a discreet mode depending on situation, sharing a controlled amount of information with other systems with the user’s authorisation, adapting information output based on terminal types, etc. Applications must provide support for these capabilities, but if these systems make decisions based on the information available to them, the user will perceive a better result and be more productive.

### **3.12 Discoverers**

Another useful component for a proficient context-aware system is a discoverer. A discoverer may be labelled the active management component, as it supervises the activities of the other components in the context-aware system. The discoverer maintains a registry of the components and capabilities of the system, thus if a component is added, removed or modified the discoverer’s registry is updated. If any of the system components fail, the discoverer updates its register indicating that the particular

component is no longer available for use. The main purpose of the discoverer is to supply the other components in the system with data relating to one another, thus it is vital that the discoverer's registry is updated promptly. (Dey & Abowd 2000)

### **3.13 Privacy**

Personal user information is a necessity to construct a system, which adapts according to a user's needs and goals, but this raises the issue of user privacy. Context-aware systems raise privacy challenges not previously considered by traditional systems. "Privacy is intrinsically bound up with control" (Weitzner, Ackerman, Darrell 2001) so control of the system must lie in the user's hands if complete user privacy is to be realised. However in most cases this is not viable, as the user will not be allowed to participate in control or because a user will not want to administer the control when acquisition and dissemination of context information is not simple. All the information that is useful to context-aware applications must be considered carefully in each situation that it is used, to avoid giving applications too much information which could be a risk to users' privacy. A method of information spaces for ensuring users' privacy and protection of data is described by Jiang and Landay in (Jiang & Landay 2002), and also by Bellotti and Sellen (Bellotti and Sellen 1993). A balance must be struck between the advantages of disclosing personal information (enabling the application to tailor services to users' needs) and the disadvantages of such disclosure in the loss of privacy.

Privacy rules may state that:

- (1) A user should be notified about the type of information collected,
- (2) A user should have the choice to halt the collection of personal information,
- (3) A user should have the ability to access the information that is gathered and delete information if desired.
- (4) Security to eliminate the possibility of unauthorised persons accessing the user information should be in place.

It is not clear how privacy rules will be governed, as different people will desire different privacy rules. In addition Human Computer Interface (HCI) problem will have to be resolved, complex information about a privacy policy will have to be displayed to the user in a manner that is easily understandable, so that the user will take the necessary steps to protect personal information without being diverted from of the main task at hand. Context-aware computing introduces extra privacy concerns but it may also be applied to "aid the negotiation over privacy agreements... by considering what privacy choice the user made the last time she entered this particular context, what privacy choices have the user's trusted colleagues made in the current context, and what privacy choices did the user make when dealing with the same data collector though in an otherwise different context." (Weitzner, Ackerman, Darrell 2001) The MIT Oxygen project is currently developing a context prototype with user privacy as the foremost driving factor.

### **3.14 What context lets us do**

There are many sample uses for context-aware systems, in fact the applications of such systems are mostly limited only by one's imagination. Applications that have been suggested so far can be roughly broken down into the following categories, based on those described in (Schilit, Adams & Want 1994):

- *Context-triggered Action* - Simple if-then rules that specify how context-aware systems should adapt. For example, if the user is in a meeting and there is a phonecall for him or her, then the call should be routed to voicemail. Triggers can also be placed on certain events, to make the context system take action such as notifying an administrator of an increase in temperature in a server room.
- *Contextual Information and Commands* - commands issued by the user can produce different results depending upon the context in which they were issued.
- *Proximate selection* - given information about the location of the user and the situation of the user, objects located nearby can be emphasized more or made easier to choose in the user interface. This prioritisation would be particularly useful on a mobile device which might have limited screen real estate. For example, a nearby printer would be proposed before one further away when the user wishes to print something. The same idea would hold true for any location-based service the user would wish to avail of.
- *Automatic Contextual Reconfiguration* - adding components, deleting components, or changing the connections between components based on current context. The issues that arise have synergies with the field of service composition (Kiciman, Melloul & Fox 2001). For example, Schilit et. al. produced a system whereby anyone carrying a PARCTAB (Schilit et al 1993) and entering a room would have it bound to an instance of a virtual whiteboard associated with that room. When the user was finished with the whiteboard and left the room, the PARCTAB's connection to the whiteboard was broken. In this situation, the context information on the user's location was used to configure the connection between the mobile device and the 'static' virtual whiteboard.
- *Metadata Tagging* - Context information can be attached to existing pieces of information, to give us more implicit information about objects in our system. This context acts as metadata about both physical and virtual objects in our system. For example, when a user records an audio clip on a handheld device, the system can attach the current context information (date and time, people present, current activity) to the clip for easy retrieval and indexing. Diaries can be created automatically, simply by 'recording' relevant context.
- *Terminal adaptivity* - the usability of mobile devices suffers from small and cluttered user interfaces. Context information can help to present the user with only relevant information, rather than overwhelming them with many different options.

The most frequently created context-aware application makes use of the user's location information by acting as a sort of virtual tour guide such as Cyberguide (Abowd et. al 1997) from Georgia Tech, or the Teleporting System (Bennett, Richardson & Harter 1994) from Olivetti Research Ltd. Such a system uses GPS or infrared tracking to determine user location with a reasonable degree of accuracy. It displays this location on the screen of a handheld device, and uses that knowledge to display information on the screen when the user is near to a point of interest. It can also be used as a log, to record places that have been visited by the user for later examination. Simple applications of such a location system would be to redirect computer output to a nearby screen, or to automatically forward a user's phone calls to the nearest phone (Want et al 1992).

PDAs which have sensors attached are a low-cost entry to a context-aware system. By fitting PDAs with sensors such as proximity sensors, touch sensors, orientation sensors, light sensors and

temperature sensors, the device itself can adapt to this context information. For instance, the user can choose whether to use the device in portrait or landscape mode by simply rotating the screen. Power management can be made more intelligent by telling the device to turn on if being held and tilted, and turn off if left idle with nobody nearby. These self-contained context-aware devices are attractive because they can make use of context without the need for the surrounding environment to be instrumented.

### **3.15 Examples of Context-Aware Systems**

Four examples of system that use context information are described in this section. These examples show different approaches to managing context and display the vast amount of areas and issues that have to be addressed when using context in a system.

#### **Context Toolkit**

The context toolkit supports context-aware applications by assisting non-context-aware applications to use context and by evolving existing context-aware applications. It shields the applications from changes in the context and the consequence of the changes. The context toolkit separates the application from context acquisition issues through the use of widgets, it also stores context gathered from the environment and constructs a history, which is made accessible to applications to make decisions on the intent of a user. The context toolkit encompasses a context interpreter to perform functionalities as outlined in the interpreter section and an aggregator to accumulate related context that the application seeks into one logical placement. (Dey 2000)

#### **Owl**

A context-aware system currently under construction by Ebling et al is the Owl context service, which “aims to gather, maintain and supply context information to clients. It tackles various advanced issues, including access rights, historical context, quality, extensibility and scalability.” (Candolin & Kari 2002) “It offers a programming model that allows for both synchronous queries and asynchronous event notifications. It protects people’s privacy through the use of a role-based access control (RBAC) mechanism.” (Ebling, Hunt & Lei 2001)

#### **Kimura**

The motivation of the Kimura System is to integrate both physical and virtual context information to enrich activities of knowledge workers. It utilises a blackboard model based on tuple spaces. The four components that operate on the tuple spaces are: (MacIntyre, Mynatt & Darrell 2001)

- (1) Desktop monitoring and handling components, which uses low-level Window hooks to observe user activities and the interpreter component
- (2) The peripheral display and interaction components that read and display the context information so that the user can observe and utilise the context in tasks.
- (3) Context monitoring component that writes low-level tuples, which are later interpreted.
- (4) Interpreter component, which translates low-level tuples into tuples that can immediately be read by the whiteboard display and interaction component.

#### **Solar**

The Solar system is a middleware system designed to support context-aware applications. The system comprises of information sources, which are sensors that gather physical or virtual context information,

filters, transformers and aggregators to modify context to offer the application practical context information. Researchers at Dartmouth College are designing and experimenting with this system, aspiring to produce a system that is flexible, scalable and reusable.

#### **4. Research Directions**

Location was unquestionably the initiator of context-aware computing, however from this paper it should be clear that although context-aware computing is still a relatively new area it has advanced far beyond the premise of location. The chief goal is to utilise as many types of context information as possible in a manner that enhances the functionality of the application and abolishes much of the effort contributed by the user. At present current mobile applications are being designed with the same location-independent mindset as traditional applications. The advantages of context information for mobile applications seems clear - applications of all types will be able to avail of a much richer set of information with which to make decisions. However, providing this information to applications in a timely and efficient manner is a challenge. High mobility applications need to be aware of changing context, even more so than their low-mobility counterparts.

Research must be carried out into the modelling of users in context systems, and into the privacy of those users. User modelling is a mature field in other areas, but there is a definite need for the expertise to be applied to context systems, and privacy is such an important issue that it must be borne in mind in all stages of the design of such systems.

The design of context-aware systems must facilitate the prevention of potentially embarrassing or dangerous situations for the end user. There are situations in which the forwarding of phone calls could be inconvenient or even dangerous, for example if a surgeon were conducting an operation. Context-aware systems need to conduct 'risk management' in order to take into account the costliness of mistakes, and factor that information into the decisions they make.

A context-aware system needs to determine how much value context will add to each individual situation, and not rely on it as a panacea. At least in the short term, context information could be unwieldy and difficult to make use of. It seems clear though that there is enough value in the potential of context-aware computing to persevere.

Strides in sensors, recognition, and wireless networking are enabling new classes of context-aware applications. It is however still necessary to lower the barriers to entry of these systems. Current methods of context provision are extremely application-specific and non-general. Context-aware systems will only become widely used once there are standard mechanisms for the sharing of context. There are many technical issues in this field, and even more people-related issues. When these issues are overcome however, there is much potential here for new kinds of interaction and applications that make use of context.

A standard is required for context-aware computing to eliminate repetition of effort experienced by developer. This standard must be put in place to allow reuse of context-aware system components so that an entirely new system need not be produced for every context-aware application. If this context standard is not fabricated the benefits of context may never be realised as the production of context-aware systems will never become prevalent. Another aspect that must be addressed in the near future is the security and privacy problems produced by context-aware computing. Mechanisms to guarantee

security of personal information must be studied to give users of context-aware applications easy of mind. Some models and abstractions that may be considered in the production of a context standard and problems that need to be addressed and resolved for privacy have been outlined in this paper.

This survey of work in the area of context-aware computing is enlightening in the different approaches to classifying types of context information. It seems clear that almost any type of information could be regarded as context, so it is important to classify this information into different types in order to be able to use it efficiently, as has been addressed by many researchers in the field. This classification will assist us in managing context information, by being aware of its individual characteristics. One can however never hope to enumerate all the information that could constitute context.

Research is also being carried out into the concept of modelling of users, systems and devices for context-aware systems. This research will hopefully lead to a reusable framework for the storage of context information, which will enable different context systems to communicate with each other and share context. Only once context information is truly ubiquitous will its full potential be realised. However it is possible to build up from simple applications of context to a more advanced level, which fully leverages the power of context-aware computing.

Candidate management systems for context are being explored and will hopefully result in a system that is flexible and intuitive, both for the user and manager of a context system. There are many issues still to be addressed, however the work so far explored leads us to believe that the future for context-aware computing is bright.

## 5. References

Abowd Gregory, Ebling Maria, Hunt Guerney, Lei Hui & Gellersen Hans-Werner, (2002), Context-Aware Computing, *IEEE Pervasive Computing*, IEEE

Abowd Greogory D. and Mynatt Elizabeth D. (1999), Ubiquitous Computing: Past, Present and Future, Available: [http://www.cc.gatech.edu/classes/AY2000/cs7470\\_spring/readings/tochi.PDF](http://www.cc.gatech.edu/classes/AY2000/cs7470_spring/readings/tochi.PDF) [2002, Nov. 27]

Albrecht Schmidt, Michael Beigl, and Hans-W. Gellersen (1999). There is more to context than location. *Computers and Graphics*, 23(6):893–901.

Bill N. Schilit, Norman Adams, Rich Gold, Michael M. Tso, and Roy Want (1993). The PARCTAB mobile computing system. In *Workshop on Workstation Operating Systems*, pages 34–39.

Bill Schilit, Norman Adams, and Roy Want, (1994). Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, US.

Cameleon project - context aware modelling for enabling and leveraging effective interaction; <http://giove.cnuce.cnr.it/cameleon.html>.

Candolin Catharina and Kari Hannu H. (2002), Modelling Context Information in Pervasive Computing Systems, Available: <http://www.ietf.org/internet-drafts/draft-candolin-cam-00.txt> [2002, Nov. 27]

Dey Anind K. and Abowd Greogory D. (2000), A Conceptual Framework and Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Available: <http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf> [2002, Nov. 27]

Dey Anind K. (2000), Understanding and Using Context, Available: <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf> [2002, Nov. 27]

Ebling Maria R., Hunt Guerney D. H. & Lei Hui (2001), Issues for Context Services in Pervasive Computing, Available: <http://www.cs.arizona.edu/mmc/13%20Ebling.pdf> [2002, Nov. 27]

Emre Kiciman, Laurence Melloul, and Armando Fox, (2001). Towards zero-code service composition. In *Eighth Workshop on Hot Topics in Operating Systems*, Elmau, Germany, May 2001.

Frazer Bennett, Tristan Richardson, and Andy Harter, (1994). Teleporting - making applications mobile. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 82–84, Santa Cruz, California, December.

Gregory D. Abowd, Anind K. Dey, Robert Orr, and Jason A. Brotherton (1997). Context-awareness in wearable and ubiquitous computing. In *ISWC*, pages 179–180.

Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton (1997). Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3:421–433.

Guanling Chen and David Kotz, (2000). A survey of context-aware mobile computing research. Technical report, Department of Computer Science, Dartmouth College, 2000.

Hans-Werner Gellersen, Albrecht Schmidt, and Michael Beigl (2002). Multi-sensor context-awareness in mobile devices and small artefacts. *Journal on Mobile Networks and Applications, Special Issue on Mobility of Systems, Users, Data and Computing in Mobile Networks and Applications*, October 2002.

Hong Jason I. and Landay James A. (2001), An Infrastructure Approach to Context-Aware Computing, Available: <http://guir.berkeley.edu/projects/cfabric/pubs/context-essay-final.pdf> [2002, Nov. 27]

Jenna Burrell and Geri K. Gay (2001). Collectively defining context in a mobile, networked computing environment. Short talk summary in CHI 2001 Extended abstracts, May 2001.

Joseph J. Martinka 1996, Requirement for Client/Server Performance Modeling Available : <http://www.hpl.hp.com/techreports/95/HPL-95-96.pdf> [2003 Jan 16]

Keith Cheverst, Nigel Davies, Keith Mitchell, and Christos Efstratiou (2000). Using context as a crystal ball: Rewards and pitfalls. In *Proceedings of Workshop on 'Situated Interaction in Ubiquitous Computing' CHI*, 2000.

MacIntyre Blair, Mynatt Elizabeth D., Tullio Joe & Volda Steve (2001), Hypermedia in Kimura System, Available: [www.cc.gatech.edu/fce/ecl/projects/kimura/pubs/kimura-hypertext2001.pdf](http://www.cc.gatech.edu/fce/ecl/projects/kimura/pubs/kimura-hypertext2001.pdf) [2002, Nov. 27]

Mitchell Keith (2002), A Survey of Context-Awareness, Available: <http://www.comp.lancs.ac.uk/~km/papers/ContextAwarenessSurvey.pdf> [2002, Nov. 27]

Morse David R., Armstrong Stephen, Dey Anind K. (2000), The What, Who, Where, When and How of Context-Awareness, Available: <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf> [2002, Nov. 27]

Nissanka B. Priyantha, [Anit Chakraborty](#), & [Hari Balakrishnan](#) (2000), The Cricket Location-Support System Available: <http://nms.lcs.mit.edu/papers/cricket.pdf> [2003 Jan 16], Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)

Paramvir Bahl and Venkata N. Padmanabhan (2000) RADAR: An In-Building RF-based User Location and Tracking System Available: <http://www-bsac.eecs.berkeley.edu/~ldoherty/radar.pdf> [2003 Jan 16] Microsoft Research  
Paul Castro and Richard Muntz (2000). Managing context for smart spaces. *IEEE Personal Communications*, October 2000.

Roy Want, Andy Hopper, Veronica Falco, Jonathan Gibbons (1992), The Active Badge Location System Available at: <http://citeseer.nj.nec.com/cache/papers/cs/13941/http://zSzzSzwww.parc.xerox.comzSzeslzSzmemberszSzwantzSzpaperszSzab-tois-jan92.pdf/want92active.pdf> [2003 Jan 16]

Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons, (1992). The active badge location system. *ACM Transactions on Information Systems*.

Schmidt Albrecht, Beigl Michael, Gellersen Hans-W. (1998), There is more to Context than Location, Available: [http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/schmidt\\_cug\\_elsevier\\_12-1999-context-is-more-than-location.pdf](http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/schmidt_cug_elsevier_12-1999-context-is-more-than-location.pdf) [2002, Nov. 27]

Schilit Bill N., Adams Norman & Want Roy (2002), Context-Aware Computing Applications, Available: <http://www.fxpal.com/people/archive/schilit/wmc-94-schilit.pdf> [2002, Nov. 27], IEEE Workshop on Mobile Computing Systems and Applications, IEEE

Shafer, S., Brumitt, B., and Meyers, B. (2000), [The EasyLiving Intelligent Environment System](#), CHI Workshop on Research Directions in Situated Computing Available:

[Http://www.research.microsoft.com/easyliving/Documents/2000%2004%20Steve%20Shafer%20CHI.doc](http://www.research.microsoft.com/easyliving/Documents/2000%2004%20Steve%20Shafer%20CHI.doc) [2003 Jan 16]

Victoria Bellotti and Abigail Sellen (1993). Design for Privacy in Ubiquitous Computing Environments. In *Proceedings of the Third European Conference on Computer Supported Cooperative Work*, pages 77–92. Kluwer.

Voida Stephen, Mynatt Elizabeth D., MacIntyre Blair & Corso Gregory M. (2002), Integrating Virtual and Physical Context to Support Knowledge Workers, *IEEE Pervasive Computing*, IEEE

Mark Weiser (1993), Ubiquitous Computing, IEEE Computer “Hot Topics”

Weitzner Daniel J., Ackerman Mark & Darrell Trevor (2001), Privacy In Context Available: <http://www1.ics.uci.edu/~jpd/NonTradUI/SpecialIssue/ackerman.pdf> [2002, Nov. 27]

W. Su, S. Lee, and M. Gerla (2000). Mobility prediction in wireless networks.

Winograd Terry (2001), Architectures for Context, Available: <http://hci.stanford.edu/~winograd/papers/context/context.pdf> [2002, Nov. 27] HCI Journal, Lawrence Erlbaum Associates, Inc, Mahwah, NJ

Xiaodong Jiang and James Landay (2002). Modeling privacy control in context-aware systems using decentralized information spaces. *IEEE Pervasive Computing*, 1(3), July-September 2002.

## State of the Art: Service Composition

Steffen Higns, Dave Lewis  
Knowledge and Data Engineering Group  
Department of Computer Science

### Abstract

This paper addresses the state of the art in service composition as driven by the current interest in web services. This paper examines this state of the art from the viewpoint of the application of state of the art to ubiquitous computing. It addresses the various issues related to service composition, including interoperability of service descriptions, service discovery, service brokering, security and trust and the modelling of service composition. Research directions in the application of service composition to ubiquitous computing is suggested.

### Keywords:

Service composition, web services, WSDL, DAML-S

## 1 Introduction

Service oriented development involves developing well defined interfaces to units of software functionality and using these to integrate software into applications. As smart spaces will consist of software functionality from a very wide range of sources, a service oriented approach to smart space system development would seem appropriate. Service composition addresses techniques for rapidly integrating existing services into richer, composite services and therefore will be important in adaptive smart space systems.

## 2 Overview

Service Composition is the orchestration of a number of existing services to provide a richer composite service assembled to meet some user requirements. The current major interest in service composition, however, stems from the emergence of web services and the possibility of composing them to provide value-added services over the WWW. Service composition techniques typically involve expressing elemental services and composite services, the latter being compositions of elemental services and other composite services. The definition of composite services requires the expression of the flow of control and information between the elemental services. Techniques for this draw heavily on business process modelling and languages for workflow. The service composition domain also overlaps with software engineering in the assembly of systems from pre-existing software components. Architectural Description Languages (ADLs) address system assembly by assuming components offer well-defined services which are composed to meet system requirements. ADLs address static aspects of such composition, including the use of connectors to express the positioning of protocol or data transformation functions between services.

In the Smart Space environment users (or more likely their agents) will be faced with a changing array of local services, plus varying access to remote web-based services, as users move about. The task of orchestrating these services to meet the needs of whatever tasks the user currently wishes to undertake therefore requires adaptive service composition, i.e. the rapid composition and re-composition of services. The problem of how to dynamically compose Smart Space services with little a priori knowledge is therefore very analogous to the problem of web service composition, solution to which could thus be exploited for adaptive management of ubiquitous computing environments

### 3 Analysis

This analysis first categorises some of the different approaches that can be taken to service composition and then examines aspects of the state of the art in service interoperability, service discover, service brokering, security and trust and the modelling of service composition. These topics are of importance in service composition since they address the fundamental problems of combining services originating from a number of sources. Services require a common interoperability mechanism so that they can be readily interconnected in a single platform. They require mechanisms for registering available services so that service composition agents can later discover their availability. Service brokers extend service discovery to support the locating of services based on partial requirements, rather than a full service specification. Security and trust are essential both in communicating information between distributed services and in trusting that information submitted to a service will not be misused and that results are a true outcome of the advertised service. The modelling of services and service composition are the key to the level of automation that can be incorporated into service composition and these related activities.

#### 3.1 *Approaches to Service Composition*

A useful taxonomy of composite service is defined in [chakraborty01b]. This differentiates between proactive composition, which is performed off-line for deployment on stable, always-up, resource rich platforms, and reactive composition where compound services are created on the fly under the auspices of some composition manager, often optimising for real-time parameters, e.g. available network bandwidth. The taxonomy also differentiates between mandatory-composite services which require the correct behaviour of all subcomponents, and optional-composite services where not all components need to be in place for service operation, e.g. data compression services optimising dataflow throughput. While the dynamic nature of smart spaces will require reactive composition, there will still be a need to integrate this with more stable model of a smart space host's business process model, which involves smart space administrators in proactive composition. This also raises the relatively unexplored issue of reconciling of personal business processes (of the Smart Space visitor) with organisation business processes (of the Smart Space host) [gary97a]. Smart Space environments, with their arbitrary mix of services and communication technologies will require reactive service composition that is optimal-composite in order to accommodate user movement between Smart Spaces, intermittent wireless connectivity and changes in accessibility to limited shared resources.

[chakraborty01b] also points out that many service composition system are centralised in how the flow of signals and information between services is managed, and this would seem to present a problem when applying them to Smart Space environments based on ad hoc networks and peer-to-peer applications, where no natural central control point may exist. However, [benatallah02a] provides a decentralised mechanism for controlling distributed control and information flow. This involves the

static analysis of state graphs that represent such flows to generate local flow rules that can be passed to lightweight schedulers operating with each service. This scheme is extended in [faudet01a] to support monitoring of the execution of composite services, another usually centralised function, in a decentralised, peer-to-peer manner.

### **3.2 Service Description and Interoperability**

It is clear that service composition requires some level of commonality in the manner in which information is passed between services. Much of the interest in service composition in the WWW has been motivated by the emergence of the Web Services Description Language (WSDL) [wsdl] and its standardised binding to SOAP [soap] and, hence to HTTP, as a common communication infrastructure. As well as defining common communication protocol mapping, the transport of data as XML allow for greater flexibility in data encoding and transformation, e.g. using XSLT [xslt]. However, though WSDL would seem a likely service description mechanism for Smart Space services, it can potentially be bound to any communication protocol. In a Smart Space, where limited device profiles and specialised wireless networks may be common, service composition may have to operate over a variety communications protocols.

### **3.3 Service Discovery**

Service discovery is the process of locating which services are available to take part in a service composition. The methods used for service discovery vary greatly depending on the environment in which the services exist. For a large scale local area network, a single database of online services might be appropriate. Service discovery is completed in this manner by the DySCo system [piccinelli01a], that uses a single service description manager, which keeps a database of all available services and the role they can play in a compound service.

As the scale of the network expands beyond that of a LAN, the methods used for service discovery must go beyond that of a simple “shout and reply” multicast. The core backbones of the internet would quickly be flooded by traffic if a sufficiently large number of clients asked every host on every network if they provide any services. For a huge array of services existing on a network on the scale of the Internet, a hierarchical lookup system like the domain name system [mockapetris87a] might be the best way to achieve scalability. The eFlow system developed by HPL addresses service discovery in this manner [casati00a]. If a service to complete the selected task cannot be found, the local service broker will ask any known external brokers if they know of the required service. For an ad-hoc wireless network, the problem becomes more complex and a further set of considerations must be made. The Anamika architecture [chakraborty01b], which focuses on service composition in an ad-hoc environment uses the service discovery features of Bluetooth to locate devices within range of the node requesting a service. In the event of a service not being located, every device that replies, is instructed to forward the request on to any devices they have within their range. This process is repeated recursively until the desired service is found.

Also worthy of note are frameworks like UPNP and JINI which represent a future direction for operating system network stacks. On top of the typical roles of such a stack (the basic delivery of information from an application on one computer to an application on another), they provide service

advertisement, service discovery and other network management protocols like NAT traversal to applications.

A UPNP enabled device uses a multicast (i.e. UDP based) version of HTTP to ask all devices on the network to respond if they are UPNP enabled as described in [Steinfeld 2001]. Recent versions of the Windows Operating System can be configured easily to become UPNP devices. Answers are transmitted by a unicast UDP variant of HTTP. Because service discovery is non-centralized, UPNP becomes impractical for any network larger than that of a large corporation. Each device can provide an XML list of all services it provides to any device.

JINI is a framework with similar goals to UPNP, though based heavily on Java. Service discovery happens via a Lookup Service, which is a centralized database of available services. If the hostname and port number of the Lookup Service are known beforehand, a simple unicast can be used to perform registration. It also supports multicast Lookup Service discovery. If a device requires a list of available services, it can locate the Lookup Service in a similar manner. If the network's router supports multicast routing, it can discover a Lookup Service on a remote network. The practicalities of this service discovery technique are explored further in [Chitrarasu et al. 1999].

### **3.4 Service Brokering**

Service Brokering is the process of selecting which services should be included in any particular service composition. To make this choice, the service broker should be supplied with some self-description by the selected services. The criteria used to make this decision could consist of the time required by the service to complete execution, the role the service can play within a composition and the price charged by the service provider. When some pre-knowledge exists of what devices will be available in an environment, the selection of a service broker is relatively simple. It can simply be designated as such by the designer of the architecture. In a number of the service composition systems investigated, the task of service brokering was given to one or more predefined devices. In an ad-hoc network, where all devices might be of similar processing power and we have no guarantee that a given device will be available at a given time, some form of nomination and acceptance should be used. The Anamika architecture uses this approach. Another approach, taken by ICARIS [tosic00a], is to have a one centralized registration manager with which devices that are willing to act as service brokers can register themselves. The selection of which device will act as service broker is made by the registration manager.

### **3.5 Security and Trust**

The secure communication between services is also a vital consideration in service composition for Smart Spaces. The ICARIS architecture approaches this problem in an interesting manner. Various cryptographic methods and protocols are encapsulated within different services, and can be chained together based on the needs of a given application. Secure, SSL style protocols (which combine asymmetric cryptography for initial key exchange and then symmetric cryptography for data exchange thereafter) can be built and then integrated with services and applications which were never designed with security in mind.

In an ad-hoc environment, it is assumed, we have many computers offering various services. For various reasons, it is important that a client (be it an end user or an adaptive service composer) can establish the identity of the machine providing the service. It is inevitable that in a truly ad-hoc environment, where we can never completely rely on any one device being present, that at some point, one device is going to have to make a "leap of faith" and agree to do business with a given service of which it has no prior knowledge. Taking this hindrance as a given, we can begin to examine an easy algorithm for establishing a degree of trust in this environment:

The client will determine that a given available service is capable of providing the functionality we require. Before trusting it with anything confidential, we establish what experiences other trusted devices in the environment have had and we ensure that the device is who it claims to be.

This is an ideal task for public/private key cryptography. Firstly, we generate a large random number. We send this to the service. The service then signs this number with its own private key and sends the signature back to us, along with its public key. We verify the signature using the public key. We broadcast to all devices on the network asking if they have also used a service running on this device. Along with the request, we send the service's public key, so each device can verify that they are referring to the same service. Each client's reply should contain two pieces of data:

1. A confirmation that the service in question, when used before, was using the supplied public key when used.
2. Some rating of the level of satisfaction with the service that was received.

How the results are handled is open to some research and debate.

There are two problems with this:

- Firstly, in the ad-hoc environment, we will probably have no pre-knowledge of the devices in the space. How do we trust them? How do we know that not only has the device providing the service hasn't been compromised, but that all the hosts telling us how great it is haven't also been compromised.
- Secondly, some thought needs to be put into what is being authenticated. Are we authenticating the device, along with all services that run on it, or do we authenticate the service only, and forget the notion of the device completely? Can we trust a service without trusting the underlying device, and how is this managed when service code is downloaded and run dynamically?

Mobile code security is also an important consideration in this context. In a dynamic computing environment, it is also important that the issues of trust are examined, to prevent a composition from being poisoned by a rogue service. Without the aid of hardware support, it is more or less impossible to trust completely any methods for verifying that a service running on a foreign system is what it claims to be[gong97a]. For this reason, we must be selective in the information that is transmitted, ensuring that sensitive data is transmitted only to a service in which we have a certain minimum level of trust.

One possible direction towards a practical solution is the wallet-of-certificates approach[lkagal01a], where trust is built up slowly through a series of cryptography based authentications. Following this thinking, a reasonable level of confidence in the identity of the service with which we are communicating can be developed.

### 3.6 *Modelling Service Composition*

Many different modelling languages have been used in expressing service composition, e.g. UML state diagrams in [benatallah02a], and often the semantics of these languages are derived from business process modelling techniques. The recent interest in web services and their expression in WSDL has resulted in XML based languages being used to define composition between WSDL service definitions, e.g. BPEL4WS, allowing service composition patterns to be readily exchanged between tools and thus reused. Such service composition specifications necessarily need to be expressed at the level of abstraction used for constituent services. Though techniques for developing business process models from business requirements are well established, they involved skilled manual activities and are not amenable to the dynamic service creation needed to meet the needs of Smart Space users lacking in business process modelling skills. A service composition transformation approach is taken to this problem in [wang00a], where existing user tasks are analysed as service compositions in one service environment and transformed to another when the user moves to another service environment. Though this effectively provides a form of seamless session mobility between smart spaces, it does not help users in composing services to achieve new tasks, perhaps taking advantage of novel services found in a recently entered Smart Space. A 'semantic gap' therefore exists between the model the user possesses of what they want to do, and the service composition models that simply express how this may be accomplished. Some work has been performed in reconciling the behaviour of a system with a high level representation of its requirements [feather98a]. However, even these high level requirements are complex to express and are typically elicited by skilled requirements engineers. Adaptive techniques are used in other contexts, such as e-learning, to dynamically map basic user profile information to adapt hypermedia documents to user's knowledge and device display capabilities [conlan02a]. A close analogy between the expression of composition in hypermedia authoring languages and ADLs is identified in [muchaluat-saade01a]. As ADLs address many of the static aspects of service composition, this may point to the utility of adaptive hypermedia techniques in adaptive service composition.

Ultimately, however, accurate mapping of user level requirements to service composition requires modelling of the real world which forms the context in which the user naturally expresses their Smart Space usage needs. Ontologies provide a way of capturing and exchanging models of the real world and making them available to automated agents. The DARPA Agent Mark-up Language initiative is defining XML-based standards for supporting the development of the Semantic Web [berners-lee01a], which aims to make the content of the web more amenable to processing by automated agents. As part of this initiative DAML-S [damls02a] uses the DAML+OIL ontology language to provide semantic mark-up of web services. This includes modelling the links between a service and the outside world, by using ontologies for expressing the inputs, outputs, preconditions and effects of a service and the resource which provides it, in a way that can be mapped to semantic descriptions of the real world. DAML-S uses WSDL to ground its service definitions and thus bind it to the actual interface technology used for a service instance. DAML-S also provides mechanisms for defining composite services. Thus the use of WSDL for defining all Smart Space services, coupled with the integrated semantic richness added by DAML-S, opens the door to using a range of rule-based, inference and other AI techniques to the problem of adapting user needs to Smart Space service composition [mcilraith01a]. Currently such application range from using of rule-engines reasoning on simple conditions on service input and output to perform limited automated service composition [ponnekanti02a] to the use of situation calculus to reason about semantic information in DAML-S

specs in tools to aid developers in the simulation, verification and automated composition of web services [narayanan02a].

It is interesting to note the web services are not the first domain to encompass machine reasoning about service composition. Machine reasoning has previously seen application to service engineering, principally in the telecommunications industry. Formal languages such as the Service Description Language (SDL) [ellsberger97a] were used to provide formal definition of services implemented in component-oriented telecommunication control systems, often referred to as intelligent networks. Reasoning with such formal service descriptions was used initially to detect unwanted feature interactions in the service creation process for intelligent networks, but research was also conducted into using them for user evaluation trials, design simulations and automation of test case generation [lodge99a]. SDL bears many similarities to DAML-S, as both express logical assertions about preconditions and post-conditions, i.e. effects, of service invocation, and can also express logical relationships and constraints between service inputs and output. However, the use of formal reasoning in intelligent network service creation was often seen as an addition to the service creation process, and therefore an overhead the benefits of which had to be directly assessed against the equivalent costs of traditional software engineering processes. Also, the logical concepts used in defining formal assertions for a particular intelligent network service was tied to that particular service and offered little benefit outside the development of that service. At best such formally defined objects were restricted to reuse in the intelligent network software market which was dominated by a few large players, and thus not open to the utility of formal definitions as a commodity or shared resource.

## 4 Research Directions

In identifying research directions related to service composition, a distinction needs to be made between research into algorithms for defining optimal service compositions, and the assembly of a suitable service platform on which different such algorithms can be evaluated. In terms of a service platform, considerations are:

- WSDL would seem to provide a good basis for the definition of service syntax and forms the basis for a lot of web service composition research. The ability to bind WSDL descriptions to various protocols, allows application specific services to be implemented or simulated on one communication platform while remaining open to use on other development platforms. A range of WSDL tools are available.
- Consistent with this choice, the use of DAML-S for defining the semantic of web services may provide a common platform for a number of service composition approaches. The movement from the use of DAML+OIL to the derivative OWL which is being standardised by the W3C points to DAML-S as a likely basis for future standardisation of semantic web definitions.

In terms of specific research areas related to service composition, the following should be considered:

- Automated Service Composition: Smart Spaces require dynamic and thus largely automated service composition. Mechanisms for such zero-code composition are required [kiciman01a]. We aim to examine the extent to which service composition may be automated and will examine schemes where pre-existing, manually developed compositions can be integrated with and used to improve the automated service composition process.
- Bridging the semantic gap: Though DAML-S may provide us with a way of openly exchanging semantic information on services, exactly how this is used to map service to what the user

wants to do is an open issue. This involves both matching semantics between service-oriented models of what user's wish to do, e.g. as tasks, and service composition models based on existing services and their semantics they possess.

- Ontologies for Smart Space Management: Can we establish some common ontologies for adaptive smart space management? This will examine the possible application of existing management models in an ontological form to the management of smart spaces.

## 5 References

[benatallah02a] 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

[berners-lee01a] Berners-Lee, T., Hendler, K., Lassila, O. (2001), 'The Semantic Web', Scientific American, pp 35-43, Issue 284 (3), 17th May 2001

[casati00a] Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M. (2000), 'Adaptive and Dynamic Service Composition in eFlow', HPL-2000-39 20000406 External, March 2000

[chakraborty01b] Chakraborty, D., Joshi, A. (2001), 'Dynamic Service Composition: State-of-the-Art and Research Directions', Technical Report TR-CS-01-19, Department of Computer Science and Electrical Engineering, University of Maryland, 19th December 2001

[chakraborty02a] Chakraborty, D., Perich, F., Joshi, A., Finin, T., Yesha, Y. (2002), 'A Reactive Service Composition Architecture for Pervasive Computing Environments', Technical Report TR-CS-02-03, Computer Science and Electrical Engineering, University of Maryland Baltimore County

[chitrarasu99] Chitrarasu, M., Joseph, K., Rao, M., (1999) "Jini by Example – Whitepaper", published online.

[conlan02a] Conlan, O., Wade, V., Bruen, C., Gargan, M. (2002), 'Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning', Proceedings of Second Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eds. De Bra P., Brusilovsky, P., Conejo, R., Springer, LNCS 2347, May 2002, pp 100-111

[damls02a] 'DAML-S: Semantic Markup for Web Services', The DAML Service Coalition, <http://www.daml.org/services/>, October 2002.

[ellsberger97a] Ellsberger, J., Hogrefe, D., Sarma, A. (1997), "SDL, Formal Object-oriented language for communicating systems", Prentice-Hall

[fauvet01a] Fauvet, M.C. Dumas, M., Benatallah, B., Paik, H.Y. (2001), 'Peer-to-Peer Traced Execution of Composite Services', Second International Workshop on Technologies for E-

Services, Eds. Fabio Casati; Dimitrios Georgakopoulos; Ming-Chien Shan, Rome Italy, 14-15 Sep. 2001, Springer, Heidelberg, Germany, pp103-117

[feather98a] Feather, M.S., Fickas, S., van Lamsweerde, A., Ponsard, C. (1998), 'Reconciling System Requirements and Runtime Behaviour', Proceedings of Ninth International Workshop on Software Specification and Design, IEEE, 16-18 April 1998, pp 50 – 59

[gary97a] Gary, K., Lindquist, T., Koehnemann, H., Sauer, L. (1997), 'Automated Process Support for Organizational and Personal Processes', Proceeding of the International ACM SIGGROUP conference on supporting group work, Phoenix, Arizona, ACM Press, pp 221-230

[gong97a] - Li Gong (1997), 'Survivable Code is Hard to Build', DARPA Workshop on Foundations for Secure Mobile Code, March 1997

[kagal01a] - Lalana Kagal, Tim Finin, Anupam Joshi (2001), 'Moving from Security to Distributed Trust in Ubiquitous Computing Environments', unpublished [kiciman01a] Kiciman, E., Melloul, L., Fox, A. (2001), 'Position Summary: Towards Zero-Code Service Composition', Eighth Workshop on Hot Topics in Operating Systems May 20 - 22, 2001 p.0172

[lodge99a] Lodge, F., Kimbler, K., Hubert, M. (1999) "Alignment of the TOSCA and SCREEN Approaches to Service Creation" in Proceedings of the 6th International Conference on Intelligence in Services and Networks, Barcelona, Spain, Springer-Verlag, pp277-290

[mcilraith01a] McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), 'Semantic Web Services', IEEE Intelligent Systems, 16(2), March/April 2001

[muchaluat-saade01a] Muchaluat-Saade, D.C., Soares, L.F.G. (2001), 'Towards the Convergence between Hypermedia Authoring Languages and Architecture Description Languages', Proceeding of the ACM Symposium on Document Engineering, Atlanta, Georgia, USA, ACM Press, pp 48-57

[narayanan02a] Narayanan, S., McIlraith, S.A., (2002) "Simulation, Verification and Automated Composition of Web Services", Proceedings of 11th World Wide Web Conference, May 7-11, 2002, Honolulu, Hawaii, pp 77-88

[piccinelli01a] Piccinelli, G., Mokrushin, L. (2001), 'Dynamic e-service composition in DySCo', 21st International Conference on Distributed Computing Systems Workshops (ICDCSW '01) April 16 - 19, Mesa, Arizona p. 0088

[ponnekanti00a] Ponnekanti, S.R., Fox, A. (2002), "SWORD: A Developer Toolkit for Web Service Composition", To appear in The Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, May 7-11, 2002 (<http://swig.stanford.edu/public/publications>)

[soap] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Frystyk Nielsen, H., Thatte, S., Winer, D. (2000), "Simple Object Access Protocol (SOAP) 1.1", W3C Note 08 May 2000, <http://www.w3.org/TR/SOAP>

[steinfeld] Steinfeld, E., (2001) “Devices that play together, work together”, EDN Magazine September 2001

[tosic00a] Totic, V., Mennie, D., Pagurek, B. (2000), ‘On Dynamic Service Composition and Its Applicability to E-Business Software Systems’, Carleton University, Ottawa, Canada Published online

[uddi00] “UDDI Technical White Paper”, 6th September 2000, [www.uddi.org](http://www.uddi.org)

[wang00a] Wang, Z., Garlan, D. (2000), ‘Task Driven Computing’, CMU document CMU-CS-00-154, May 2000.

[wsdl] Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2002), “Web Services Description Language (WSDL) 1.1”, W3C Note 15 March 2001, <http://www.w3.org/TR/wsdl>

[xslt] XSL Transformations v1.0, W3C 1999: <http://www.w3.org/TR/xslt>

## State of the Art: Middleware in Smart Space Management

Sinead Cummins: Cork Institute of Technology  
Alan Davy, Jason Finnegan, Ray Carroll: Waterford Institute of Technology

### Abstract

Ubiquitous computing, and specifically Smart Spaces, is an emerging paradigm for interactions between people and computers. Its aim is to break away from desktop computing to provide computational services to a user when and where required. Large numbers of heterogeneous computing devices provide new functionality, enhance user productivity, and ease everyday tasks. In home, office, and public spaces, ubiquitous computing will unobtrusively augment work or recreational activities with information technology that optimizes the environment for people's needs (Manuel Román et al, 2002). The aim of the M-Zones program is to do novel research into the management concerns connected to such spaces.

The rationale behind this paper is to review the existence (or lack) of a suitable middleware infrastructure for the development and management of applications and services in ubiquitous computing environments. This paper will present current practice in middleware infrastructure, analyse the requirements of a ubiquitous computing environment, and draw conclusions as to its relevance to the M-Zones project and future research required.

This paper will begin by giving an overview of current middleware practices, analysing the requirements of a ubiquitous computing environment, and drawing conclusions as to its relevance to M-Zones. It will then proceed to look at specific aspects of middleware such as service creation (OSA/Parlay), networking architectures (Jini) and management using agent-based technology.

Keywords: Smart Spaces, Middleware, Jini, Agents, Parlay, Service Creation Environments, Application Servers

## 1 Introduction

This paper provides a state of the art overview and a discussion of important approaches in the field of middleware. Special focus lies on the point of view of Smart Spaces and Managed-Zones (M-zones) with their specific requirements. A Smart Space is a physical space rich on devices and services that are enabled to act autonomously on behalf of an individual user. An M-Zone is an administrative domain that deals with all issues on managing one or more Smart Spaces in order to run services and devices in an efficient way.

A major topic for Smart Spaces is the provision of a middleware. Such a middleware should hide specific (and mostly proprietary) aspects of the infrastructure (devices) to ease the development and deployment of components for Smart Spaces. This is an important basis for intra- and inter-domain service offerings as well as their management. Furthermore, this provides users the opportunity to roam between Smart Spaces or even M-Zones without losing their well-known service environment.

This paper investigates available and mature middleware technology from three different viewpoints. Section 2 starts with a general introduction into middleware and a categorization of middleware

technology. This general overview should enable the reader to understand the different approaches of middleware technology deployed today. Furthermore, the section 2 introduces three technologies in more detail, whereas all of these three technologies have been identified as potential candidates for a Smart Space middleware by the M-Zone program. These technologies are Jini (for device access), mobile and intelligent agents (as a technique to generate a flexible service environment, including decision making components and mobility components) and finally OSA/Parlay as the momentarily only solution to access a telecommunication network as a service provider.

Section three defines the requirements of Smart Spaces with regard to middleware technology. These requirements are kept generic (e.g. scalability, re-configuration, quality of service) in order to allow the identification of criteria for the evaluation of middleware technology. The requirement part of section three is followed by detailed discussions of Jini, Agents and OSA/Parlay.

Section four describes the future directions of research work within the M-Zones program. The authors of this paper intend to give guidelines and recommendations for the application of middleware technology to Smart Spaces. Furthermore, the applicability of the three candidate technologies will be investigated. This process leads to a number of important questions like:

- Is the integration of Smart Spaces with telecommunication networks feasible?
- Is management by delegation a suitable paradigm for Smart Space management?
- What approaches can be taken to integrate middleware and management, e.g. to build a decentralized management system that incorporates SNMP and (mobile) agents?

This paper can not give answers to those questions. This paper is intended to provide the necessary background information that allow researchers of the M-Zones program to define, specify and later also to deploy a distributed management platform for Smart Spaces that is build upon state of the art middleware technology.

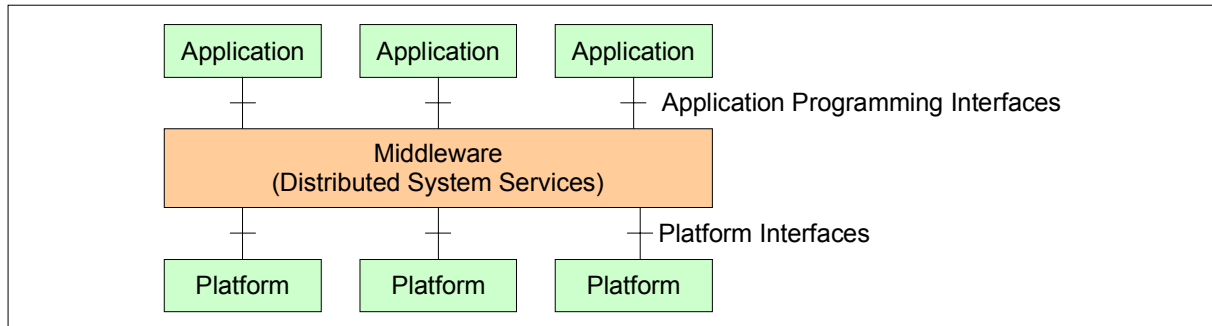
## 2 Overview

The following section starts with a general introduction into middleware and a categorization of middleware technology. This general overview should enable the reader to understand the different approaches of best practice middleware technology deployed today. Furthermore, this section introduces three technologies in more detail, whereas all of these three technologies have been identified as potential candidates for a Smart Space middleware by the M-Zone program. These technologies are Jini (for device access), mobile and intelligent agents (as a technique to generate a flexible service environment, including decision making components and mobility components) and finally OSA/Parlay as the momentarily only solution to access a telecommunication network as a service provider.

### 2.1 *Middleware*

The term middleware is used today to describe any technology that provides an abstraction from the heterogeneous components of computer environments, like hardware platforms, networks, protocols, operating systems, applications, groupware, and databases. Middleware ties components of a distributed application together and solves the problem of scalability (Bakken, 2002). It supports application deployment using well defined, probably standardized, Application Programming

Interfaces (API). Those APIs and related interfaces decouple the applications from technologies as shown on Figure 1- Middleware – Logical View of Middleware



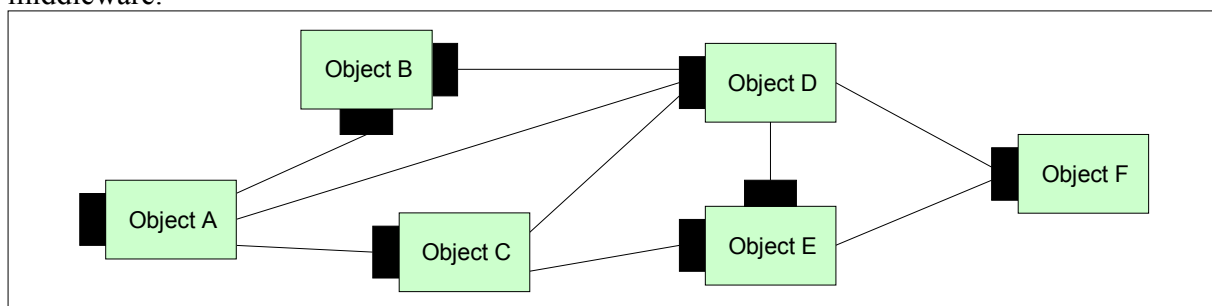
**Figure 1- Middleware – Logical View of Middleware**

Two different types of middleware are present: Inter Process Communication (IPC) and Distributed Transaction Processing (DTP). IPC based middleware systems operate with a direct communication among distributed components. IPC based middleware is made up of Remote Procedure Call (RPC) middleware, Message Oriented Middleware (MOM), and Object Request Broker (ORB).

DTP based middleware realizes the link between a client and any kind of database. Those middleware systems define abstract interfaces to databases for handling transactions and concurrency. DTP based middleware includes portable Transaction Processing (TP) Monitors and interconnected Database Management Servers (DBMS).

### 2.1.1 Paradigm of Client – Server

Distributed objects communicate in a client/server relationship. The client calls operations on a server and receives return values. The server offers those operations on its interface(s), runs the business logic once an operation is called, and generates the return value. The technical realization of interfaces on the server to be used by the client and the communication between client and server is the task of the middleware.



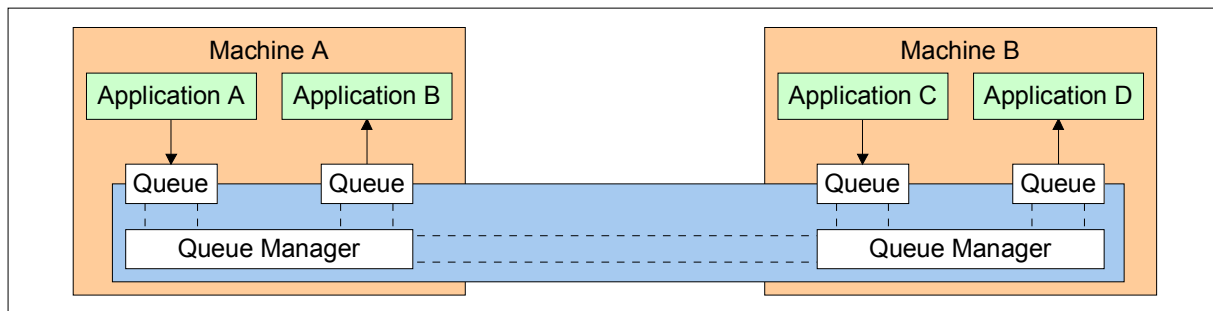
**Figure 2 - Middleware – Distributed Objects**

The actual relationships between distributed objects are not defined by the paradigm. Each object can access every other object. The definition of the relationships needs to be done in the development process of the distributed application. Hierarchical structures are not supported. They need to be realized individually.

### 2.1.2 Message Oriented Middleware

Message Oriented Middleware accomplishes the communication of application components through the exchange of records called messages (A. Campbell et al, 1999). Messages are strings of bytes that have meaning to the applications that exchange them. MOM is event-driven. It can support asynchronous as well as synchronous communication. However, there is no general standard or common set of specifications. At least three different techniques for the exchange of messages are available.

Message passing applies a direct communication model. A message is sent directly from one application to another. The model is connection-oriented. A logical connection between applications needs to be established. The exchange of messages can be either asynchronous (via a polling model or by callback routines) or synchronous (sender blocks until a message is returned).



**Figure 3 - Middleware – Message Queuing**

Message Queuing applies an indirect communication model. Applications communicate via a message queue. This model is connectionless. Messages are put in queues for immediate or subsequent delivery. A Queue Manager is responsible for message handling and delivery. Message queuing implies the support for different types of Quality of Service (QoS), such as reliable message delivery (no packet loss on the network), guaranteed message delivery (messages are delivered immediately or eventually, at least after a specified period of time), and assured non-duplicate message delivery (message are delivered only once).

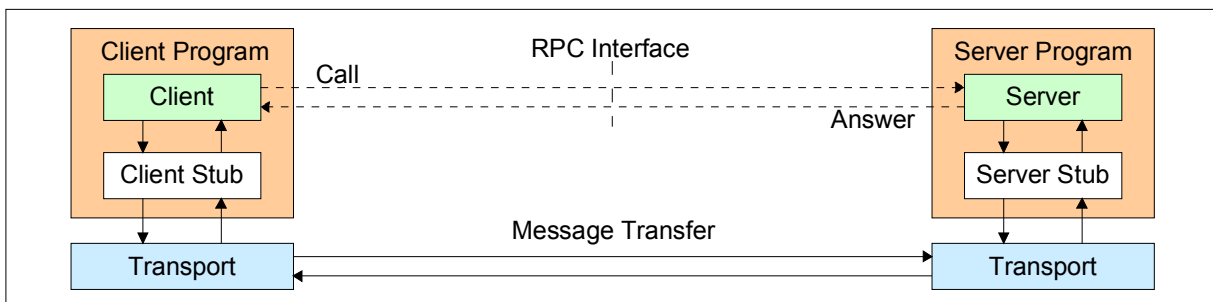
Publish and Subscribe is based on trading. Publishers produce information and offer them. Subscribers sign in to certain topics and receive messages. Publishers and Subscribers usually do not know the others existence. Applications can be loosely coupled increasing the flexibility of the design and operation of a system. Those systems can be reconfigured dynamically without interrupting operations. New applications can be added without disturbing existing applications. For more information refer to (M Erzberger, M Altherr, 1999).

### 2.1.3 Remote Procedure Call

The Remote Procedure Call is a common accepted and widely used technique for the communication between applications in distributed environments. All major operating systems support client/server communication via logical RPC interfaces. That is, clients and servers have local, logical interfaces that

are called stubs. Those stubs realize proxy objects for remote functions. To the client, the stub looks like the remote procedures thus providing location transparency (hiding the actual location of the server).

The logical function calls on the RPC interface are mapped to physical invocations on the stub. The latter one has the knowledge on how to access the network's transport layer to send the request to the server and to receive the reply for the client. The client itself usually blocks until the server replies to the remote call. On server side, the stub calls the requested function (procedure), as the client would have done if called locally. The programmer of the server has to provide some functionality to support a remote call, like directory, security, etc.



**Figure 4 - Middleware – Remote Procedure Call**

To use the network's transport layer for data exchange between client and server, the data has to be streamed through a communication channel. The serialization of the data for the transport inside of the network is called marshalling. The serialization requires that all data to be transported be pre-described, including type, format, and length. Such a description is to be provided by the programmer of the server in form of an interface definition using an Interface Definition Language (IDL). An IDL is a high level, universal notation that is capable to describe interfaces independent of the actual programming language (Bakken, 2002).

#### 2.1.4 Object Request Broker

Object Request Brokers can be classified into a group compliant to the Object Management Group's CORBA and one compliant to Microsoft DCOM. In the following paragraphs, a look at a generic ORB architecture is explained, which is the basis of both CORBA and DCOM (J. Bacon et al, 2000).

The ORB is a software component that enables objects to initiate requests and to receive responses. All inter-object communication happens via the ORB, independent of the actual location of the objects (local or remote). The ORB enables objects to communicate over networks with different communication protocols and to reside on different hardware platforms. It provides the mechanisms by which objects make requests and receive responses. Furthermore, it is responsible for managing and locating the objects, supporting both inter-object communication and communication between objects and external services.

Most ORB products provide a set of components as specified by the Object Management Group (OMG). At first, an IDL is used as notational language to describe interfaces of objects. It is

implementation-neutral and needs specifications for the mapping to implementation languages. An Interface Repository contains all IDL definitions for attributes, operations, user-defined types, and exceptions. The Basic Object Adapter (BOA) represents the interface between the ORB and server applications. It dispatches objects that the server application maintains, and exchanges messages with the server objects. The Static Invocation Interface (SII) is a stub-based interface used by client programs in order to invoke services on application objects. Finally, the Dynamic Invocation Interface (DII) provides a generic interface that does not require stubs, but supports dynamic construction of object invocations by the client program at runtime.

The ORB establishes a client/server relationship among objects. A client can transparently invoke any method on any server object. The ORB intercepts the call, looks for an object that has implemented the call, passes the parameters to that object, invokes the method, and returns the result. Clients can invoke either 'one way requests' when no result is expected or synchronous requests.

The ORB itself is responsible for locating the server object by use of the implementation repository, to exchange data between client and server (including marshalling), to invoke the server (dynamic server invocation), and to recover from failures of the server object. The ORB provides objects with services like naming, lifecycle, property, relationship, query, and licensing.

### 2.1.5 Distributed Transaction Processing

Transaction Processing (TP) has been used for mainframe-based applications where TP Monitors have managed the limited number of connections to databases. They have acted as connection concentrators, reducing overhead and increasing the performance of database access. Today, TP Monitors are used as a solution for transactional integrity in database environments (P.A. Bernstein, 1990). They support the so-called ACID properties that describe the quality of transactions:

- **Atomicity:** All operations that an application performs, which involve updates to any kind of resource, are grouped into a "unit of work." This unit is referred to as atomic, meaning it is indivisible. Any partial completion (due to system failures) will be rolled back.
- **Consistency:** At the end of a transaction, all resources that have participated will be in a consistent state.
- **Isolation:** Concurrent access to shared resources by different units of work (performed by different applications) is coordinated so that they do not affect each other. Transactions that compete for resources are isolated from each other.
- **Durability:** All updates to resources that have been performed within the scope of a transaction will be persistent or durable.

The standard for DTP represents a basis for TP Monitor products. It allows programmers to share resources keeping the overall system in a consistent state. Databases are connected to the TP Monitor via a standardized interface called XA.

The XA interface is the bi-directional interface between a transaction manager and a resource manager. The XA interface is not an ordinary Application Programming Interface (API). It is a system-level interface between DTP software components (Technical Standard a, 1992).

Applications use the TX interface to communicate with the TP Monitor. The TX (Transaction Demarcation) is the application-programming interface by which the application program calls the Transaction Monitor to demarcate global transactions and direct their completion (Technical Standard b, 1995). The Standard Transaction Definition Language (STDL) provides a TP Monitor independent API in the form of a higher layer, procedural language for the description of transactions.

## **2.2 Jini**

### **2.2.1 Middleware and Jini**

In the following review, Jini will be examined under headings, which describe the key requirements of any network Middleware technology. These broadly are designed to provide access to services on the network, in an abstract and flexible manner, in order to satisfy requesting users. Users typically can be software client processes or possibly specific applications run by human users. The middleware thus stands in the middle between the servers and the clients facilitating the communication. For networking Middleware this general aspiration can be broken into requirements in the following areas.

- Networking
- Service Description
- Service Connection
- Connection Management
- Reliability
- Scalability
- Security

Jini has specific capabilities in each of these areas, which we examine separately. In addition, we will examine the main competing technologies to Jini, before reaching a conclusion as to its applicability to the M-Zone research.

### **2.2.2 Introduction to Jini**

Jini is a network infrastructure that runs on top of Java, which allows services to dynamically join and exit without impact on the network or the network users. As a service based architecture, Jini provides a solution for the evolving ubiquitous computing requirements. Kumaran [2002,Preface] explains that “It abstracts both the devices and software under a service notion and supports dynamic community formation and dissolution.” This community, or federation is described by Mahmoud [2000,pg 244] as “a set of services that can work together in a single distributed computing space, to perform a task” Jini is a technology that will allow developers and manufactures create a range of computerised devices that can instantly connect both wired and wireless into a network to share services regardless of the underlying operating system or hardware. “Jini software gives network devices self-configuration and self-management capabilities; it lets devices communicate immediately on a network without human intervention.” Stang & Whinston [2001,pg33]. The networks are described as “self healing” Stang &

Whinston [2001,pg33], in that devices can leave a network without affecting the operation of the remaining devices, this eliminates network failure in the case of machine crashes or power surges. The means by which devices can achieve this is by leasing- a fundamental concept in Jini.

A Jini federation is made of look-up servers clients and services. A client finds a service by specifying a Java object type, and interacts with that service through a java object of that specific type. The java object can be downloaded from the network without the client having any knowledge of the service beforehand. We will now look at each of these fundamental concepts and how they are applicable to middleware software.

## **2.3 Agents**

In section 2.1 a taxonomy of today's common middleware approaches was detailed. However agents are a new and evolving form of middleware technology and so are not often considered in such classifications. Agents differ from other middleware technologies in that they cannot be addressed under just one of the listed middleware approaches. Instead agents can incorporate a number of middleware technologies and concepts such as message passing, RPC, distributed object technology, etc. The volume of research carried out in this field indicates that agents need to be considered in this taxonomy as an emerging middleware approach.

The term "agent", in the computing context, has exploded in popularity in the last number of years. A simple count of the number of references to agents in the first 3 issues of IEEE Pervasive Computing could be taken as an indication of the current research popularity of agents. However 'Agent' can be a very ambiguous word and is often used to describe software elements that would not be classified as agents in the context of this paper. The purpose of this overview is to clarify that context as well as introduce agents to the novice reader. This will be done by examining where agents have evolved from, what exactly an agent is and the types of agents that exist.

### **2.3.1 History of Software Agents**

Software agents have their roots in Artificial Intelligence (AI). Some of the earliest works on AI were the neural network (McCulloch & Pitts, 1943), Vannevar Bush's (1945) depiction of a computer assisted future in his article "As we may think" and the Turing Test (Turing, 1950). The idea of an agent originated with John McCarthy in the mid-1950's, and the term was coined by Oliver G. Selfridge a few years later. "They had in view a system that, when given a goal, could carry out the details of the appropriate computer operations and could ask for and receive advice, offered in human terms, when it was stuck" (kay, 1984).

Software agents really began to emerge from work that involved Distributed Artificial Intelligence (DAI). One of the approaches developed under this title was to use multiple, distributed intelligent agents or Multi-Agent Systems (MAS). According to Nwana (1996) this research first started in 1977 and is the first of two main strands of research into agent technologies. This strand concentrated on issues such as "...the interaction and communication between agents, the decomposition and distribution of tasks, coordination and cooperation, conflict resolution via negotiation, etc." This strand was much more focused on pure AI and many of the agent mechanisms were inherited from AI technology.

The second strand started in the early 90s, and here agents moved from their pure AI roots into a much wider variety of applications (e.g. Personal Agents, Information Agents, Negotiation Agents etc). Pattie Maes, now one of the world's leading voices on agents, lead the way with many well-received papers including "Agents that Reduce Work and Information Overload" (Maes, 1994). Agent research continued through the 90s and into the new millennium with much attention moving to the latest agent paradigm of mobile agents.

### 2.3.2 What are Agents?

An agent is a piece of software. However, the problem is that numerous definitions of how agents differ from conventional software exist (Franklin & Graesser, 1996) and reaching a consensus on a single definition is virtually impossible. The Oxford English dictionary provides the following definition of the word Agent:

"...a person or thing that exerts power..." and "...who acts or has the power to act"

When talking about software agents most definitions attempt to identify a set of properties they believe an agent should command. Some of these properties are (Nwana 1996), (Kampis 1998):

- Autonomous: act on its own without being controlled by others or by outside forces.
- Cooperative (Communicative): interact with other agents to fulfil some goal.
- Intelligent (learn, reason, adapt): acquire knowledge, process this knowledge in order to arrive at some conclusion and adjust based on this conclusion.
- Reactive: act as a response to some stimuli, event or condition.
- Proactive: act in advance to deal with an expected difficulty (anticipatory).
- Mobile: move from one location to another in order to fulfil a goal.

An agent does not necessarily possess all of these attributes and often there is a trade off between them. For instance, a key attribute in a mobile agent will obviously be mobility. This does not mean that it is not intelligent or cooperative. It still may have these properties, if required, but perhaps to a lesser extent than mobility. However, there is one property that distinguishes an agent from a common piece of software: autonomy. This power of self-government is the basic attribute required by all agents. This leads to the following definition of an agent: "A software component that can autonomously fulfil some goal or set of goals". The context of this discussion focuses on agents that are autonomous. Definitions from other areas, e.g. management agents or search agents, are not within the scope of this discussion.

One issue that has caused confusion in defining agents is the overuse (and misuse) of the term agent itself. In recent times more and more technologies appear to use the word agent to describe software components. Looking at the properties above many of these so-called agents could not justify the use of the term.

### 2.3.3 Relevance to M-Zones

M-Zones' focus is on the management of mobile and dynamic environments. Agent technology provides intelligent, flexible and mobile software that can facilitate a variety of management functions in such environments. Through the incorporation of the characteristics listed in section 2.3.2 (such as mobility, intelligence, cooperation etc) agents can be a very powerful technology, especially in

dynamic environments such as smart spaces, where intelligence and mobility can facilitate smarter and more efficient management.

### 2.4 Parlay/OSA

The purpose of the Parlay/OSA API's is to allow third party applications access to features of the telecommunication networks. Parlay provides a standard way for applications to access information such as user location or account balance from the telecoms secure servers. The Parlay APIs allow access to the telecoms network through a Parlay gateway. This gateway is inside of the telecom network and is customised for that particular network. Applications that want to make use of the parlay gateway are run on application servers. These application servers can be on the same intranet as the Gateway or can access it over the Internet.

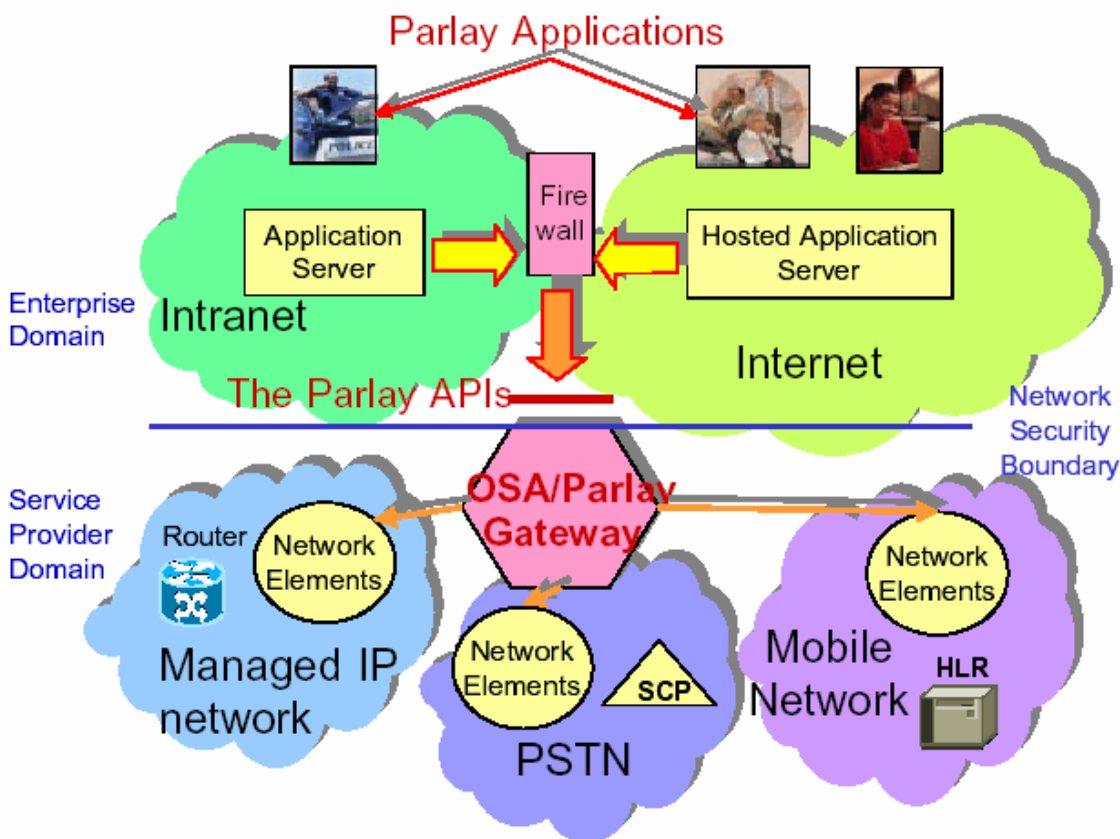


Figure 1 - Parlay gateway between networks

#### Relevance to M-Zones

The M-zones programme is aiming to “ Develop novel information and communications management technology to support dynamic, integrated management of participants, information appliances, and smart space infrastructure.”

The Parlay/OSA API's allow would allows us to integrate local smart space management with existing GSM , 3G and PSTN(Traditional Telephone) Phone Networks. This would enable M-zones management to switch sessions from Wide area Networks to Local Area Networks seamlessly and to locate Users at all times when outside of Smart Spaces.

### 3 Analysis

The following section defines the requirements of Smart Spaces with regard to middleware technology. These requirements are kept generic (e.g. scalability, re-configuration, quality of service) in order to allow the identification of criteria for the evaluation of middleware technology. With these requirements in mind this section is followed by detailed discussions of Jini, Agents and OSA/Parlay. Each of these three emerging middleware technologies deals with different aspects of the Smart Space. Jini is a middleware technology developed for smart devices access. Mobile and intelligent agents may be considered as a technique to generate a flexible service environment, including decision-making components and mobility components. OSA/Parlay momentarily is considered to be the only solution to access a telecommunication network as a service provider. The purpose of reviewing these middleware technologies is to highlight the fact that current state of the art middleware technologies will have to change the way they operate if they want to survive in the emerging smartspace environment (Kurt Geihs, 2001).

#### 3.1 Middleware

The few ubiquitous computing environments that exist today tend to be highly specialised and based on application-specific software. Applications developed for interactive environments should be able to interconnect and manage large numbers of disparate hardware and software components. They should operate in real-time; dynamically add and remove components to a running system without interrupting its operation; control allocation of resources; and provide a means to capture persistent state information. Frequently these components are not designed to cooperate, so not only do they have to be connected, but also there is a need to express the "logic" of this interconnection. In other words, inter-component connections are not merely protocols, but also contain the explicit knowledge of how to use these protocols. Thus, viewing the connections simply as an application-programming interface is not enough. Cooperation among different applications is also difficult to achieve without a common platform. In order to model applications in this domain the need to define a common design methodology based on new paradigms independent from the technology is important. A model to abstract the main components of a ubiquitous computing environment is needed in order to formalize the development of interactive environment applications. These components may be classified into three abstraction layers:

- Physical deals with technological constrains.
- Middleware defines structure and the cooperation of abstract services.
- Application concerns the user interfaces. (Maffioletti, 2001)

Thanks to these abstractions a middleware will present a uniform access abstraction for different ubiquitous devices, allowing them to interact and cooperate. This will allow the writing of applications scaling both on services offered, and on devices composing the system. The model is intended to provide a standardized view of basic interactive environment functionality.

### 3.1.1 Breakdown of requirements

The main requirements that must be provided by a middleware infrastructure for services in ubiquitous computing would fall under the following headings.

#### 3.1.2 Mobility

New wireless communication technologies provide connectivity for laptop computers and personal digital assistants, phone organizers offer the processing power, and application-level protocols such as the wireless application protocol—a tiny first step—allow convenient applications to run on these devices. Undoubtedly, these developments point toward the widely expressed goal of accessing and processing information almost “anywhere and any time.” Independent of a person’s current location, one can already use voice communication via mobile phones almost anywhere, and have worldwide access to personal e-mail, bank accounts, and home-country news over the Web.

Mobility introduces a key technical challenge because the available resources vary widely and unpredictably. Communication bandwidth and error rates change dynamically in wireless communication networks, a mobile system’s battery power decreases, portable devices can be temporarily switched off or unreachable because of network partitions, and the monetary cost of communication can vary significantly. Envisaging a middleware system that makes these dynamics transparent is difficult; so middleware must support the applications to explicitly accommodate these changes. Another mobile-computing issue, location awareness, demands that mobile-computer applications know their operating environment for context-dependent activities, such as giving directions or employing more or less stringent security mechanisms.

The characteristics of Message Oriented Middleware would be very suitable in the ubiquitous environment. MOM does not require constant connections to be held between communicating devices, as it can pass messages asynchronously, accounting for ad hoc connections in the ubiquitous environment.

#### 3.1.3 Heterogeneity

The ubiquitous computing vision assumes that future computing will comprise diverse computing devices ranging from large computers to microscopic, invisible processing units contained in objects that are used in daily life, (e.g. light switched, mugs, etc). These computing devices may all need to communicate over different network mediums and protocols, (e.g. 802.11b, Bluetooth, IrDA, etc.).

Present middleware technologies such as CORBA can be implemented in different languages, letting heterogeneous ORBs talk to each other. Other OOM technologies such as SOAP communicate in a structured format (XML) over common network protocols (HTTP). Message oriented middleware can also provide data communication over diverse networks, such is the case for Java Messaging Service where as long as the applications communicating are running in a Java Virtual Machine (JVM) corresponding to their platform, communication is not a Problem.

Middleware for a ubiquitous environment must follow the characteristics of present middleware technologies in regards to the way they can be implemented in heterogeneous environments, and communicate over heterogeneous networks with dissimilar protocols.

#### 3.1.4 Scalability

The middleware has to scale well for both a large number of cooperating services, which realize the application in each application context, and a large number of devices involved each time the application are used. Services represent the logical dimension of the application, while devices represent its physical dimension.

Different middleware technologies can handle being scaled in different ways. OOM such as CORBA would be considered a connection oriented middleware; this is to say if two or more orbs are to communicate, there must be an open connection between them. Middleware must provide applications with consistent functionality the more a system scales. In a ubiquitous environment the amount of connections between devices at any time could be immense. The management of these connections could require more resources than available, which is not desirable in such a resource-restricted environment.

Message oriented middleware mostly works on a connectionless bases, where the sender and receiver may be disconnected when communication is initiated. When a system, using MOM, is scaled to an environment with possibly hundreds of devices communicating with each other, such as a ubiquitous environment, messages being passed may build up because of devices being disconnected for unpredictable amounts of time. This may compromise the performance of critical systems running in a ubiquitous environment.

A possible solution to the scalability issue for ubiquitous environments is to group small areas together, such as a room. In this room, because the devices in the room may be made up of static and dynamically connected devices, it would be safe to assume a limited number of devices may be in operation at any one time. Using these assumptions, different middleware solutions may be used in combination with each other in order to control the scale of the environment. For example in a small-scale environment, an OOM may be used to control connections between devices, such as the .NET framework, which is based on XML Web services communicating through basic Web protocols. A collection of these environments could be considered another group communicating at a higher level, connecting these environments together. Using a method like this would break down the environment, into controllable scaling environment.

#### 3.1.5 Configurable / Re-configurable

It is becoming increasingly apparent that most existing middleware technologies cannot accommodate the great diversity of application demands in areas like mobile and ubiquitous computing. The main reason for this is the black-box philosophy adopted by existing middleware platforms. In particular, existing platforms offer a fixed service to their users and it is not possible to view or alter the implementation of this service—that is, they are closed systems. This basically means that all network and connection information is hidden from the application layer, thus not allowing applications see changes in network topologies. Applications should be able to be configured to any network topology (802.11x, GPRS, Bluetooth, etc), thus the underlying middleware should not be considered as a black

box paradigm but more of a book that can show certain required information about the underlying network topology to the application layer.

Devices in ubiquitous environments will automatically detect other devices; forming ad hoc networks spontaneously such as Bluetooth technology. A middleware infrastructure for a ubiquitous environment will have to re-configure itself to changes in the network, and also be able to reflect this change at the application layer. The Java-based Jini system appears to anticipate these requirements. Jini defines a middleware infrastructure for spontaneous networking in which Java objects can discover, join, and interact with communities of devices. Whether the Jini approach will scale to the requirements of ubiquitous computing, however, remains unclear.

### 3.1.6 Quality of Service

Increasing concerns about service quality have led to several proposals that advocate integrating QoS management into networking and distribution infrastructures. QoS management at the middleware and application levels aims to control attributes such as response time, availability, data accuracy, consistency, and security level. From the Internet perspective, QoS concerns seem to arise automatically when commercial applications meet a best-effort communication environment. When clients must pay for a service, they are certainly concerned about QoS, and they will expect to pay less for lower quality. From a ubiquitous computing perspective, the guarantee of QoS is an even greater issue. For example if a heart monitor was part of a ubiquitous environment, then the guarantee of the quality and delivery of this important data could mean life or death. Also for public acceptance of a ubiquitous environment, people must be able to trust the security of personal data that can under go potential total monitoring (Kurt Geihs, 2000).

At present current products such as IntServ and DiffServ are being tested to guarantee QoS over IP. The new Internet protocol Ipv6, may also provide in itself QoS, but these are yet to be developed to a level of open use and acceptance. For CORBA middleware, the Object Management Group recently proposed new messaging extensions that support QoS guarantees such as message delivery and error handling. Research projects have produced specific Corba-based platforms for handling individual QoS categories such as real time (D.C. Schmidt, 1998) or replication and fault tolerance (S. Maffei, 1996). Others have addressed generic frameworks for generating and operating customized systems for various QoS categories (C. Becker et al, 2000). Although interesting QoS management into middleware architectures is essential, a procedure for doing so has yet to be agreed upon.

The ubiquitous computing environment introduces extra issues regarding QoS that present QoS architecture for the Internet or wired networks do not cover. This new environment introduces problems such as wireless channel fading and mobility (M. Naghshineh 1997). These are two very important elements that have to be addressed if QoS architecture is to be successfully mapped to a ubiquitous computing environment. As these problems can affect the performance of a network and application, it is often proposed that QoS be handled at the middleware layer (K. Nahrstedt, 2001).

(Klara Nahrstedt, 2001) suggested that QoS-aware middleware architecture for ubiquitous computing environments has to address four main aspects of QoS. First, QoS specification to allow description of application behaviour and QoS parameters; Second, QoS translation and compilation to translate specified application behaviour into candidate application configurations for different resource conditions; Third, QoS setup to appropriately select and instantiate a particular configuration; Finally,

QoS adaptation to adapt to runtime resource fluctuations. Under these four aspects the problems that the ubiquitous environment introduces to QoS may be tackled.

An example of a problem introduced by ubiquitous environments would be the transmitting and receiving of multimedia data to a wireless media player application. Applications such as these can run using different configurations, i.e. streaming videos using ISO's MPEG layer 2 (ISO/IEC, 1994) encoding versus ITU's H.261 (ITU-T, 1993) encoding. The former method is for high quality video storage and transmission, which would require a large amount of network resources to be available. The latter being used for low bit-rate videoconferencing over narrowband integrated services digital network lines (ISDN) and telephone lines, respectively. In a ubiquitous computing environment due to fading signals, network bandwidth may drop considerably. In this event a QoS aware middleware infrastructure should be able to recognize that a change in network performance has occurred, notice that the application can handle different configurations, and adjust the video encoding method to suit.

In this example it is clear that the application layer needs to know some knowledge about the network layer and visa versa. QoS aware middleware should provide an abstract link between the two layers, controlling the application when network connectivity fluctuates. Another issue that should be considered is handing off between two networks. A hand off is where a mobile device moves from one network to another, vertical hand off being when a device moved from one network topology to the next (Wi-Fi to GPRS), and horizontal hand off being roaming between similar network topologies (Wi-Fi to Wi-Fi). QoS aware middleware should be able to adjust applications to react to any changes in network resources.

In ubiquitous computing environments, the question of guaranteed QoS is open. Will applications require a single Service Level Agreement (SLA) for it to function within acceptable parameters, or just based on a best effort approach? One possibility is that for each SLA, there is a subset of agreements that follow a set of further SLA's for different situations. This approach takes into account the unavoidable characteristics of the ubiquitous computing environment, such as network fading and network hand off.

Present technologies that provide some form of Quality of Service would include 2KQ (Klara Nahrstedt, 2000b), Agilos (B. Li et al, 1999), TAO (D. Schmidt et al, 1999), and QuO (J. Zinky et al, 1997). These middleware infrastructures provide a method of application configuration and adaptation to environment changes through QoS programming environments. In a ubiquitous computing environment, containing a wide variety of devices, network topologies and applications, it would be necessary to have custom designed QoS resolutions to suite different configuration changes applications may need to undergo in the event of fading or mobility (Klara Nahrstedt, 2001). This implies that the most suitable application service paradigm for ubiquitous computing is adaptive in nature; hence the provision of this adaptive quality may be provided at the middleware layer (Markus Lauff, 2000).

## **3.2 Jini**

### **3.2.1 Networking – Federations**

Jini relies on a network for its existence. A Jini network consists of a network of many services. Applications are created by combining these services in groupings called federations. Li [2000,pg12] explains that a Jini client can use the services of a federation by first joining that federation. This same client can then at a later stage, leave this federation and join another federation in order to access its services. Any service within a Jini federation may make use of other services in order to perform its own task. This means that a Jini service can also act as a client of another Jini service. The lookup service then controls actions within the federation.” The lookup service decouples the client-to-server relationship, enabling distributed dynamic reconfiguration.” [Li, 2000,pg14]

A Jini federation is made up of three vital components: servers, clients and lookup services. Every Jini network must have one or more lookup servers. The lookup server is where all the client and services information is exchanged. Here “All the services announce their availability and unavailability, so that clients can notice the existence of them” [Rong 2001, pg 148].

### **3.2.2 Service Description – Attributes**

Edwards [2001,pg 246] explains, “Attributes are Java objects that are attached to service proxies” Services attach these attributes when they publish their proxies, and clients can search for proxies by looking for certain attribute patterns. They can then download the attributes of a service to examine them more closely. Attributes are used to add extra descriptive information to a service. For example a printer may have a location attribute to specify where the printer is, or a status attribute to determine if there is paper in the tray or not. These attributes are Java objects that are attached to the service proxy.

### **3.2.3 Service Connection – Lookup**

A Jini service finds the lookup service in which it must describe itself through a multicast protocol called discovery, it then registers in lookup services with its proxy along with some attributes. The proxy is a java object, which is capable of carrying out the services duties and the attributes simply describe the information of the service. The lookup service maintains a map between each Jini service and its attributes. “When a new Jini-enabled device advertises its service, the lookup server adds that information to the map.” [Stang & Whinston 2001, pg35]. A Jini Client can then request from the lookup server a list of Jini servers that match the requested attributes. The client can then select the desired server from the returned list. The client downloads the proxy of the wanted service and uses this proxy to communicate with the service itself and the lookup service no longer participates in the rest of the communication process between the client and the service.

A unique feature of Jini technology is its lack of necessity to transfer significant executable code between Jini devices. “Jini software sends only the code for the interface that the client uses to communicate with the server; the rest of the program remains on the server and actually executes there” [Stang & Whinston 2001,pg 36] This strategy overcomes the problem of slow speeds faced by

comparable strategies such as applets. Jini systems download less “because a Jini network usually only transfers the results of whatever code the server executes” [Stang & Whinston 2001, pg36]. When a Jini client receives the lookup service reply that responds to its request for a service, it initiates a connection with the server. The Jini specification does not stipulate how an application implements a service. The only requirement is that a Jini service must implement a specific interface; and it is this interface that the server gives to the client. The client has no prior knowledge of any server and only knows how to communicate with the server through the given interface.

### 3.2.4 Connection Management – Leasing

Jini’s approach to self-healing is through a technique called leasing. Jini requires that resource holders continually renew their leases on resources. Edwards[2001,pg80] considers the scenario of a digital camera which has joined a federation. The camera announces the fact that it is available for use. However consider the scenario where the user indiscriminately dislodges the camera from its cradle without turning it off. To the other members of the federation, this may look like a partial failure situation, they may not be able to determine if the remote host to which the camera is connected has gone down, if it’s simply slow to answer, if it’s not answering network traffic because a change in its configuration, if the camera’s software has crashed or even if the camera has been smashed with a hammer. Regardless of how it was disconnected, it has not had a chance to unregister itself from the middleware system before disconnecting. Services that wish to use the camera will see it registered but will not be able to use it. From the point of view of the lookup service for middleware, if service registrations are never properly cleaned up they will accumulate and slow down operation. The system is not self-healing, partial failures aren’t recognised and cleaned up, and services that hold resources on behalf of others may grow without bound. This flawed system will also eventually require explicit human intervention to administer the system.

Jini uses leasing as a means to overcome these problems. Rather than granting access to resources for an unlimited amount of time, the resource is “leased” to some consumer for a fixed period of time. In order to maintain a resource, a resource consumer must show continued interest in the resource. A Jini lease may be denied by the grantor of the lease. They can be renewed by the holder. Leases will expire at a predetermined date unless they are renewed. This provides a lightweight and distributed management of resource allocation. In the case of the dislodged camera, since the it will not grant new lease or re-grant any existing leases, no new users will be allowed to use it and existing users will unregister themselves automatically. Leasing ensures that the management of persistent storage and services used by the members of a Jini community is distributed and virtually maintenance free.

### 3.2.5 Reliability

Stang and Whinston [2001,pg33] define reliability as a measure of “how well a device or network performs in the presence of disturbances”. However processes often crash, or someone will shut them down. Or the machine running the process will crash or stop. Jini is capable of managing these changes due to the fact that it expects devices to randomly move in and out of the network. In Jini when a server becomes unavailable, the client automatically goes looking for an alternate server. “Once it locates another server process (or the failed process comes back up), the Jini client can reconnect. If no server

is available, the client waits or informs the user” [Stang & Whinston, 2001,pg34] This functionality is transparent to the user.

Edwards[2001,pg62] explains that “Jini supports serendipitous interactions among services and users of those services”. Services can enter and exit a federation in a very “lightweight” manner. Interested parties can be automatically notified when the set of available services changes. Furthermore, every device or service that connects to a Jini community carries with it all the code necessary for it to be used by any other participant in the community.

These features combined make Jini virtually unique among commercial-grade distributed systems infrastructures. They make a Jini community virtually administration-free. Edwards [2001,pg63] explains “A cooperating group of Jini services will be resilient to changes in network topology, service loss, and network partitions in a clean way.”

### 3.2.6 Scalability

A system is defined as being scalable by Stang and Whinston [2001,pg34] if “the overhead required to add more functionality is less than the benefit that functionality provides.” Thus if adding a server to the network makes it more difficult to accomplish a task, the system is not scalable. “Adding Jini services to a system gives clients more choices in the devices they can communicate with.” [Stang & Whinston, 2001,pg 34]. Thus more devices providing the same service increase a Jini system’s reliability.

“Jini addresses scalability through federation.” [Edwards 2001,pg 64] Federation is the ability for Jini communities to be linked together or federated into larger groups. The ideal size for a single Jini community is a workgroup, consisting of a number of printers, scanners, PDA’s and network devices required by a group of 10 to 100 people. Jini groups can then be brought together in a community in order to make their resources sharable. Access to services within other workgroups is possible through federation. “Specifically, the Jini lookup service-the entity responsible for keeping track of all the services in the community- is itself a service” [Edwards 2001,pg 64]. A lookup service of one community can register itself in other communities offering itself as a resource for the community.

A Jini federation is self-managed; a device inserts itself into a network. Jini technology “dynamically discovers the services it needs for processing client requests” (Stang &Whinston 2001,pg 34). Jini federations are designed to exist in a flexible environment thus making them more scalable.

### 3.2.7 Security

Within Jini, the main security issues are to prevent unauthorised or unknown clients from accessing protected services. Jini tackles this by ensuring that remote clients can be prevented from even seeing what services are offered by the network. Only when they become part of the federation or network are they allowed see its services. This provides a first line of defence.

The architecture of Jini also fundamentally protects against Viral attack. Jini servers provide a client the interface with which it may access the services of code residing on a server. The services are accessed through the network and the code itself is typically not transferred. Viruses on the other hand for infection, typically require code mobility and use this mobility to transfer executables and infect

other computers. Stang & Whinston (2002,pg34) explain that “In a Jini environment, before code can move onto another machine, it must satisfy the client’s security policy.” The system administrator can ensure that only trusted machines are permitted to download code.

### 3.2.8 Competing Technologies

“For Jini to accomplish network nirvana, it must become widely adopted” [Clark 1999,pg 18] Standing in its way of achieving this are a few strongly backed competitors.

#### **Universal Plug and Play (UPnP)**

Microsoft’s Universal Plug and Play (UPnP) is open distributed networking architecture for pervasive, peer-to-peer connectivity of intelligent devices. UPnP supports pervasive, invisible networking, zero configuration, and dynamic formation of device communities. UPnP is built using established open Internet standards such as IP, UDP, TCP, HTTP, XML, and SOAP, for various devices to discover and join a network dynamically. No configurations, no device drivers. UPnP internetworking is based on IP standards. Thus every device joining the UPnP community acquires a network address of its own. UPnP is platform and language independent and is geared toward dynamic ad hoc networking of devices, however it does not handle services provided by software implementation or any other entity. UPnP is dependent on SOAP as a protocol for RMI. It does not discuss security aspects within the network. “Currently, it depends on the presence of other security components, such as Firewall, authentication, and authorization servers, to provide security functions.”[Kumaran 2002,pg287] UPnP registers the various devices with DNA. It does not have a separate lookup service as in Jini. Microsoft’s partners in this initiative include Intel and Hewlett-Packard. As such it is a system for ad-hoc networking but does not address the requirements of large scale service deployment. Jini works both at the ad-hoc and enterprise level.

#### **HP JetSend**

JetSend, from Hewlett-Packard, is a device-to device communication technology that allows devices to negotiate information exchange intelligently; JetSend enables devices to transfer data without needing to know anything about the other device. JetSend is both complementary and competitive to Jini. JetSend can work with both wired and wireless protocols. “In the corporate network, JetSend technology can become an alternative to fax by allowing you to send documents from one location to another by printing directly onto a high-quality printer.”(Kumaran 2002,pg288) This however assumes the existence of a JetSend-enabled device at both ends of the network. The communication model does not require driver installation or configuration, which makes it easy to plug and play with different devices, JetSend does not depend on a transport protocol; hence it can work with any bi-directional transport protocol, including TCP/IP, IR, RF, Bluetooth, IEEE1394, and others. JetSend can work with various networking technologies and can coexist with such technologies as UPnP, and Chai ApnP. Since Jini is protocol independent, it can embrace JetSend as a communication protocol for communication between devices.

JetSend, is targeting a specific requirement, that of remote printing, using JetSend as the communication technology. While this is an initial starting point, it is limiting for other service/communication requirements.

## HP Chai

Hewlett-Packard's Chai is a full-featured Java based appliance platform for developing and running applications on embedded devices. The ChaiAppliance communications is based on Web standards such as HTTP and XML, rather than on Java. However, ChaiAppliance was developed in the Chai Virtual Machine, which is HP's clean-room developed version of Java. This could possibly make ChaiAppliance Plug and Play useful as a bridge between Jini and UPP.

### 3.2.9 Why Not Jini

"Because Jini was explicitly designed to cover such a wide range of possible deployment scenarios-all the way from servers down to light switches- there are a few hardware situations where Jini is not appropriate"(Edwards 2001,pg 103) If a device is truly isolated or is so fundamentally uninteresting that it would not be used over a network, then Jini is probably not appropriate. However, fewer and fewer devices are now truly isolated. If a device is unlikely to be in the vicinity of a device with a JVM, and needs to be so cheap that it cannot have a JVM embedded on it, then Jini is not a viable solution." Java has to exist somewhere in the loop for devices to participate in Jini"(Edwards 2001,pg 103). This is the primary limitation that of minimum hardware/performance capabilities.

### 3.2.10 Conclusion

Jini is a possible network middleware for the new age of ubiquitous computing. It enables communication between applications and devices in a heterogeneous network environment. "Jini portends the movement of computing from general-purpose machines to specialised processors." (Stang & Whinston 2001,pg38) However Jini technology is still in its infancy. "Because Jini technology represents a fundamental shift in application development, adoption has been slow, but enthusiastic"(Stang & Whinston 2001,pg38) The problem of noise filtering in portable devices has not yet been solved. Also Jini is still viewed by many as primarily for embedded systems. Although the responsibility of controlling an enterprise network is less than an attractive prospect to many human network administrators, it is unnerving for many to let a mere "piece of software" do the work, while is a possibility of Jini.

## 3.3 Agents

Currently middleware and management systems are separate entities, however, the integration of these two has become the focus of much research. This section will address agents for management while keeping the general middleware requirements identified (section 3.1.1) in mind. Agent technology implicitly addresses many of these middleware characteristics (e.g. mobility, scalability, heterogeneity etc), hence the following analysis of agents will inherently address many of these.

### 3.3.1 Types of Agents

The definition of an agent is more ambiguous than would be preferred. The lack of a concise agent definition in turn causes a lack of clarity when discussing the different categories of agents. A simple web search using the term agent will return a large amount of references, many using different adjectives to describe the agent type. For example words such as Intelligent, Information, Mobile,

Trading, Personal, Search, Negotiation, knowbot, softbot, etc may all be used to describe the agent type. This is, too say the least, a problem when trying to classify agents into types. Outlining an agent typology can be as contentious an issue as identifying an agent definition and varying opinions are common. Nwana (1996) lists seven types of agent:

- Collaborative
- Mobile
- Information / Internet
- Interface
- Smart
- Reactive
- Hybrid

The paper goes on to provide a critique of its classification detailing the opinions of two independent reviewers. Both suggested that agents defined by what they do (e.g. interface, collaborative), had been confused with agents defined by the technology that enables them (e.g. mobile, reactive).

Alternatively, Magedanz, Rothermel and Krause (1996) classify agents as either single-agent or multi-agent systems. Single-agents are further sub-divided into either local or networked agents while multi-agents are sub-categorised into DAI-based agents and mobile agents. Local agents can access only local resources. Network agents are able to access both local and remote resources. DAI-based agents are focused on intelligent behaviour and the coordination of multiple agents. Mobile agents are obviously focused on the ability of the agent to move around a network.

These two classification models approach agents from a different perspective. The latter model takes an overall system classification approach, i.e. single or multi-agent, and further categorises based on the type of agent approach used to realise that system. The former of the two models describes agents at a more individual level without really taking into account the overall context. Here an agent is described more on what they are or what they do and many of these resemble the properties of agents discussed before.

In reality, due to the lack of a formal definition, most agents still tend to be classified in a haphazard manner. Often this is based on either the agent's purpose or some of its properties. This is similar to Nwana's classification but not restricted to the seven types listed. However, two of the most frequently discussed agent types are classified based on their properties. These are Intelligent and Mobile agents. In the context of this paper 'Intelligent' and 'Mobile' only suggests that these are the properties most focused on and not exclusive of other agent properties. This distinction is made as mobile agents, in some writings, fall under the banner of intelligent agent.

### 3.3.2 The Management Problem

The management scope of the system has been previously mentioned but an attempt has not yet been made to identify specific management tasks that need to be considered. Two of the most important areas of management that must be looked at are Services and Networks. Smart spaces will offer unlimited numbers of services to their users and as such these services, along with the networked

resources they use, will have to be managed. These two ideas are key to any smart space architecture, and therefore the management of these areas are also central to the M-Zones research.

The distinction has already been made between inter-zone management and intra-zone management. An important issue in this regard will be the understanding that both services and networks may be in separate domains that are under the control of separate operators, and possibly using different smart space architectures. Any smart space management architecture will have to take this into account by allowing management systems in different domains (and possibly of different types) to interoperate and share management information.

Service management must recognise the fact that services in emerging networks will be abundant, accessible from many locations by many concurrent users, created using heterogeneous technology and provided by operators in separate domains using differing architectures. Services may be required to use network resources or other services to fulfil their task. The services themselves will be much more complex incorporating various types of media and incorporating a variety of newly available functionality/technology (e.g. location information). Another consideration is that services are not necessarily static and may be able to migrate across domains on demand. Due to the high volume and mobility of end-users, service security will become a more complex area. These are just some of the service issues that will need to be addressed by a smart-space management architecture

The emergence of smart spaces and new service architectures in turn creates a need for new intelligent and adaptable networks and network management systems. One of the major issues that new management systems will have to face is a huge increase in the number of networked resources. Also due to the highly mobile characteristics of smart spaces, devices (users) will be able to attach and detach from the network in a highly dynamic way. It must also be possible to introduce and remove devices from the network with minimal effort and highly automated configuration. Furthermore these networks will have to be capable of multiple forms of communication (i.e. voice, data and video) which adds a further dimension to the management problem.

In effect, smart space environments call for new network and service management solutions, as current management approaches alone will not suffice. Today the most common type of management is done using Network Management Systems (NMS). These focus primarily on managing the network and do not facilitate the management of services (perhaps what is needed is a 'NSMS – Network and Service Management System'). Even in today's environment, these approaches are considered by many to be inadequate and so will clearly be unable to meet future network and service management demands.

Cabri, Leonardi and Zambonelli (1999) identify a number of limitations of the traditional NMS approaches. These approaches are based on a centralised client-server model in which a central administrator client collects management information from other network components. A management protocol such as SNMP or CMIP is typically used to transport the management information. This approach is limited in terms of scalability and reliability and is best suited to small networks. Also these approaches are more suited to networks that deal with devices rather than with services and data provided by network nodes.

### 3.3.3 Agent Management vs. Agentless Management

Before progressing further it may be of value to look at some simple pros and cons of using agents in management by comparing agent management to agent-less management. Obviously the point of

incorporating agents into our management framework is to provide a solution that will improve the management capabilities of the system. This should provide more fault tolerant systems i.e. more automated, more intelligent fault correction;

### **Advantages of Agent-based Management**

- **Reduced Network Traffic:** With agents, the management logic is delegated to remain locally on individual nodes, either temporarily (mobile agents) or permanently (intelligent agents). In this way the node is managed locally and therefore network traffic can be greatly reduced by potentially removing multiple messages (polling) across the network. Also, depending on the level of intelligence the agent has, it may be possible to filter unwanted information at a node before transmission, reducing network load when the agent returns its acquired data.
- **Robust:** Agent management solutions have the potential to be more robust as management is not solely dependent on a central point. So even if the network is down the agent may still be able to complete its task at its node.
- **Heterogeneous Execution:** Agents are typically written in a scripting language and therefore can run on a variety of heterogeneous systems.
- **Scalable Management:** The traditional centralised client/server approach suffers from scalability problems due to centralised nature of management. Agents lead to de-centralised management and hence more scalable management.

### **Advantages of Agentless Management:**

- **Simpler deployment:** no need for agent or agent execution environment on target machine.
- **Widespread acceptance:** Existing management technology such as SNMP is widely accepted and this should not be underestimated.
- **Security:** With the proliferation of agents around a network there are some serious security concerns that need to be addressed in agents.

### **3.3.4 Mobile & Intelligent Agents**

With regard to management the two most relevant agent types are Mobile and Intelligent agents. Many have been looking at the potential of these approaches in management for some time, with a large number concentrating on network management aspects (Gurer, Lakshminarayan & Sastry 1998) (Zapf, Herrmann & Geihs 1999). Aside from their central properties of mobility and intelligence, both these agent types often require cooperation with other agents. Consequently a method for allowing agents to communicate is essential to both. Many systems in the past have looked to KQML for agent communication. More recently another agent communication language has begun to emerge in the form of FIPA ACL (Agent Communication Language). The two are similar in many ways but KQML is more widely known, as FIPA ACL is only a new standard. Intelligent agents are the more established methodology but the mobile agent approach has built up a great deal of momentum in the last number

of years. In fact in recent years much of the research attention has switched from intelligent agents to mobile agents

### **Intelligent Agents**

As stated before intelligence is based around an agent's ability to reason, learn and adapt to within its environment. Intelligence in these types of agents is usually derived from AI approaches such as Case Based Reasoning, Expert Systems, Neural Networks, etc (Gurer, Khan & Ogier).

When creating a knowledge-based, AI system (such as an expert system) a method of knowledge representation and reasoning has to be chosen. Knowledge representation is how the data or knowledge that is known to the system (and specific to the problem domain) is represented. This can be based on rules, frames, cases etc and is kept in a knowledge or case base. Knowledge reasoning is where the known information is used as a model for solving a new problem. Rule based systems consist of if-then (production) rules and facts. If some fact is matched successfully to the 'if' part of the rule, the 'then' part can subsequently be concluded. Starting with some initial fact, rules can be chained together to form more complex conclusions. With case-based previous experiences are stored in cases. The system then attempts to match the current problem to the similar experiences in these cases. If a similar experience is found the solution or diagnosis used in that case can be applied to the new problem and then this will be stored as a new case.

An expert system is a computer system that simulates the judgement and behaviour of a human (or organization) that has expert knowledge and experience in a particular field. Expert systems are knowledge based which means they already have specific knowledge about the application area and use this knowledge to solve problems. Expert system shells are expert system applications without the domain specific knowledge (rules) and can be purchased 'of the shelf' for quicker development.

Neural networks attempt to model the human brain and as such process information in a similar way. They consist of a large number of highly interconnected processing elements (neurons) working in parallel to solve a problem. Conventional computing uses known algorithms to solve problems (i.e. set of pre-programmed instructions) whereas neural networks learn by example. Neural networks do not know the solution, instead they learn by example and can adapt and modify their behaviour based on their environment (or more specifically their inputs).

Intelligent agents are often discussed as a potential technology for use in Network Management. This approach is based around the principle of deploying agents at nodes around the network to perform management tasks locally. Since they are intelligent as opposed to mobile (static agent) they remain permanently at their node. The focus here is on inserting intelligence into the node to allow it do a certain level of self-management. This delegation of management responsibility leads to a less centralised management system and along with reducing network traffic can increase automation and deliver a more robust and scalable system.

Intelligence, however, has many applications as it brings decision-making and adaptive capabilities to a system. This allows systems to be aware of changes in its environment and potentially re-configure/adapt themselves accordingly.

## Mobile Agents

The concept behind mobile agents is that, as the name suggests, the agent is not bound to one location and can move to other network nodes. This can come under a number of banners including Remote Programming or Mobile Code and despite the fact that this has only received serious attention in recent times the concept behind it is rather old. As early as the 70s “remote (batch) job processing” was being done and in the 80s “function shipping” or “remote evaluation”.

When talking about agent mobility there are two main forms to discuss. These are Remote Execution and Migration. Remote execution is where an agent, including its code and data, is transmitted to a node/server where it is executed and its task ends here. With migration on the other hand, the agent’s execution state is also transmitted with the agent allowing it to suspend and resume its execution. Once suspended an agent can then be transmitted to another node and resume its execution there.

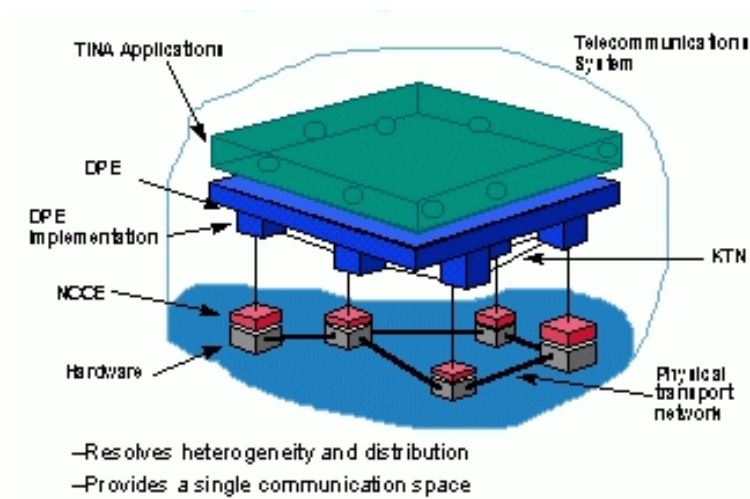
It is a generally accepted principle that mobile agents are developed in a machine independent language or interpreted language. The reason for this is that the execution environment an agent may be currently running on can vary greatly and developing the same agent for a number of different platforms is a huge overhead. Most of the mobile agent platforms available today, such as Grasshopper ([www.grasshopper.de](http://www.grasshopper.de)), are fully implemented in Java.

Mobile Agent Technology (MAT) was, in its early days, considered its own separate “religion” for distributed processing. “Remote programming is considered as an alternative to the traditional “Client/Server programming” based on the Remote Procedure Call (RPC) paradigm.” (Magdanz, Rothmel & Krause 1996). Even though both essentially had the same aim, MAT was believed by the majority to be mostly incompatible with Distributed Object Technology (DOT) such as the Object Management Groups (OMG) Common Object Request Broker Architecture (CORBA). CORBA, and DOT in general, have gained widespread acceptance within the telecommunications and networking environment due to its ability to support interoperability between objects in distributed heterogeneous domains. As CORBA allows location transparent access to objects, CORBA objects can be written in any programming language and running on any platform in a separate domain and can still interoperate. To facilitate this CORBA uses well defined interfaces.

However, the opinion expressed in the above quote has undergone a major shift in recent times. (Breugst, Hagen & Magendanz 1998) A common understanding has now been formed that MAT should enhance DOT rather than rival it. The reason for this is that both technologies offer different advantages in distributing software. CORBA agents are static and cannot move from the domain in which they were started. MAT however is designed to move so by combining both MAT and DOT the advantages of both can be extracted to realise a highly flexible distributed object environment. One major advantages of this is that (as was mentioned as a problem of new service management) service objects could now migrate across domains on demand or as required by the user. Another advantage is that non-agent (legacy) systems can be integrated into this environment. If the agent system uses CORBA then any legacy CORBA system or application can be incorporated into the new system.

### 3.3.5 Existing Management Solutions

It is a worthwhile exercise to look at some of the existing management solutions and architectures. Much work has been done in the field of telecommunications management, mostly through standards bodies such as OSI, ITU and TINA-C. One particular standard that is of interest is the Telecommunications Information Networking Architecture developed by the TINA-Consortium.



**Figure 6 – TINA layered architecture**

This architecture will not be discussed in much detail at this time but further investigation in this area is suggested. TINA contains four major architectures that define the concepts and principles for creating a TINA compliant system. These are the Computing, Service, Network and Management architectures and contain many concepts that may be of relevance to both agent-based management and to the M-Zones management objectives in general. The architecture can be split into different layers as can be seen in the diagram above. At the bottom layer is the physical hardware of the network. Upon this is the Native Computing and Communications Environment (NCCE) which is made up of heterogeneous Operating Systems, communications and support software. Above this is the Distributed Processing Environment (DPE) which is a distributed object environment very similar to systems such as CORBA. The top layer is the application layer and this is made up of application objects.

As can be seen from the discussions before regarding MAT and CORBA, any agent environment that combines the two has the potential to deliver great rewards. If the application /service objects in TINA's application layer are looked upon as Agents then many of TINA's recognised concepts can be applied (in particular from the Management architecture) to a Mobile Agent system.

Another management concept worth considering is the ever-present FCAPS (Fault, Configuration, Accounting, Performance, Security) acronym. These five management functions will be important when considering the management tasks involved in smart spaces.

### 3.3.6 Agent Standards

## **FIPA – Foundation for Intelligent Physical Agents**

The FIPA organisation was started in 1996 in Switzerland and currently has almost 60 member organisations. The group aims to develop standards that promote the interoperation of heterogeneous interacting agents and agent-based systems. Its first standard, FIPA 97 was published in October 1997 and contains specifications for agent management, agent communication language and agent software integration. FIPA has so far primarily worried about high-level agent-to-agent communication and has not said much about mobility.

Since then, FIPA97 has been further extended with a FIPA 98 and FIPA 2000 release.

## **MASIF – Mobile Agent System Interoperability Facility**

MASIF (<http://www.fokus.gmd.de/research/cc/ecco/masif>) is a joint submission of GMD FOKUS, International Business Machines Corporation, Crystaliz, General Magic, and the Open Group which was adopted by the OMG in 1998. MASIF, like FIPA, has its long-term focus on the interoperability of heterogeneous agent systems. The specification is a collection of definitions and interfaces that provides an interoperable interface for mobile agent systems. It contains two main interfaces: the MAFAgentSystem, for agent transfer and management, and the MAFFinder, for agent naming and locating (Milojicic et al 1998). MASIF has primarily considered mobility and, to a lesser extent, interoperability among heterogeneous agent systems. It does not say much about agent communication as mobile agents main consideration is mobility.

### **3.4 Parlay/OSA**

In order to create, manage and deploy services for OSA/Parlay Networks, there are several issues which need to be addressed. The first issue I will explore is Application servers, as most services need a server to run upon (except for peer-to-peer services).

Service Platforms, Reusable components and Service Creation Environments are all methods being developed to simplify the creation and management of services. Service platforms extend upon reusable components and Service creation Environments extends upon Service platforms.

The OSA/Parlay interfaces are Middleware in the form described in paragraph 3.1.3, Heterogeneity, allowing various networks running assorted operating systems and communications software to communicate with one another.

Both the OSA and Parlay initiatives to define a common interface have both been industry led, with little academic involvement. There is currently no reference implementation or ideal model of an OS/Parlay gateway. The state of the Art is being led by companies releasing their own commercial implementations.

#### **3.4.1 Application Servers**

All Parlay applications have to be deployed on an application server of some sort to be able to access the Parlay gateway. This application server may also be inside the telecoms network, which would mean that applications could not be updated, maintained or monitored easily by the third party who created them.

Another option is to have trusted application servers under the control of the third party Service developer/provider, which would have access to the Parlay gateway. This would allow for ease of updating and maintenance of the applications but would require a lot of resources and is not feasible for smaller companies/individuals.

In order to keep maximum amount of control over applications that use its network some telecoms will be using Application Development Environments and Service Creation Environments (SCE) to create another layer of abstraction between the developer and the Parlay Gateway. By using these developer environments, network operators can limit the access of third party applications to network features as much as they want while also speeding up application development time.

The question of whether the telecom network operator or the developer should host the application server is mostly down to security as highlighted by Schneier (2001). How open does the telecom want their network to be? From past and current examples (i.e. Broadband) most telecoms keep a tight grip on their infrastructure and opening up their network to third parties is not something they like to do. From a commercial side the telecoms are relying on third party applications to increase their ARPU (Average Revenue Per User), so it is not clear which route the telecoms will take.

### 3.4.2 Reusable Components

The concept of building services for Parlay/OSA networks using reusable building blocks has been investigated by Prof Hanrahan and his Team at CETAS. In their paper, Nana(2002) they divide the code that makes up a service into service dependent and service independent. Their goal is to define generic reusable components that enable the easy creation of services for TINA, Parlay and other Next generation networks.

These reusable Components would include controlling sessions and streams. Session control is generic as most services require connecting to something and this connection has to be set-up, monitored and then destroyed. Streams are required for transferring data of any type over these sessions. The data being streamed could be anything, text emails, news stories, or video clips. Service management, such as billing mechanisms, fault management and load management are handled by Service Independent Components, but these features are dealt with more comprehensively in Service platforms

### 3.4.3 Service Platforms

An Alternative to using a number of generic components when creating a new service is to build the service on top of an existing platform that provides all commonly required features. This approach is explored in "A Service Platform for Internet-Telecom Services using SIP" - Bessler (200). They propose using thin clients in the form of HTTP and WAP at the user end, while the service provider would control the service platform that the clients connect to. The services and content for the platform would be created by third party content providers and by the service provider themselves.

The service platform itself handles subscriber management and is the sole point of authentication, Bessler (2000). This allows for a User to have only one contract and facilitates easier payment for

services. One problem with this approach is the requirement for a secure certification system or other authentication system.

### **The Lucent MiLife™ Solution**

The MiLife™ Intelligent Services Gateway (ISG) Lucent (2002) is a Parlay Gateway, which also uses optional convenience classes unique to MiLife. To help developers create applications to use the MiLife ISG, Lucent have created an ISG Simulator as part of a Software Development Kit (SDK). The SDK allows the developer to create their application and test it on the simulator with script files to simulate users on the network. Sample programs and script files are included in the SDK and these show some of the features of the ISG in action. Applications can be written in any language that supports CORBA, but Java is recommended and used in all samples.

The ISG SDK is designed to help developers developing for the MiLife Parlay gateway, but due to the open nature of the Parlay/OSA any program developed for one gateway should work on them all. This is why it is important to avoid shortcuts such as the convenience classes in the ISG SDK that could tie a developer down to a particular implementation of a Parlay/OSA gateway. The MiLife SDK is available from Lucent now to registered developers, along with a list of compatible networks.

### **The NOMAD Project**

The NOMAD project is a new project undertaken by WIT, Dundalk IT and Dun Laoghaire Institute of Art and Design. The aim of the project is to develop new framework models and systems to aid in the creation and deployment of services for 3G and wireless networks. The team working on the project are focusing on location-based services and are also investigating the Parlay/OSA interfaces. This project is in its early stages but will have a large overlap with the M-zones project in the area of Parlay/OSA or other 3G type services.

Nomad's objectives are to examine the state of the art in 3G, WiFi, SIP, and Parlay. They will then identify some service(s) that will utilise the technology across the various bearer networks, i.e. 3G, IP etc. They then propose that they will develop a service creation framework, identifying the building blocks required to deploy services, and specifying methodologies to do so.

Nomad will combine these technologies and resources to develop a set of services utilising 3G and WiFi technologies. It will probably utilise Parlay and SIP. It will endeavour to develop a framework for the creation of additional services, and also hopes to deploy a toolkit which will facilitate this.

#### **3.4.4 Service Creation Environments**

The concept of using a Service Creation Environment (SCE) to create services encompasses many of the advantages of a service platform. Using an SCE means building your service within rather than on top of an existing system. The SCE helps the service builder to take a specification of a service, to model it and then to implement it. In this paper Jagot (2002), the CETAS team proposes a model for a service creation environment based on a framework they have named SEMTA (Software Engineering based on Modelling for Telecommunications Architectures).

### **Hughes Software Systems SCE (Software Creation Environment)**

Hughes Software Systems (HSS) is a leading member of the JAIN initiative that is trying to create standard interfaces for service creation and management. HSS is creating its SCE to comply with the Parlay/OSA standards so that applications created using it will be compatible with any telecommunication network that has a Parlay gateway. The SCE is designed to be useful for creating applications for different types of network including IP, PSTN and wireless networks.

The SCE uses a GUI to speed up the development of applications and services and is designed to allow reuse of service logic through the use of Service Independent Building-Blocks (SIB). A SIB is similar to a JavaBean in that it is a self-contained re-useable chunk of code. Examples of SIB's are 'Billing', 'Call Forwarding' etc

The advantages of using their service creation environment are given by HSS as "Rapid development and maintenance of revenue generating applications and services

Reduction in initial and operating costs for service creation and maintenance  
Faster time to market for delivery of new applications and services  
Quicker turnaround for customer specific application customization needs"  
HSS (2002)

The SCE Graphic Environment is a Java Applet running inside of a browser window which would allow for easy deployment of applications depending on the application server to be used.

The HSS SCE has an "Integrated testing environment " which would probably include a network simulator of some kind, allowing a developer to simulate customers using the application over the 3G/wireless/IP network.

### **TrueConverge Service Creation Environment**

The TrueConverge Service Creation Environment (SCE) as described in TrueConverge (2002), is another example of a complete graphical user environment for designing creating, testing, packaging and deploying services. There is also a TrueConverge application server to run the completed applications on.

There is little to choose between the TrueConverge and the HSS SCE as neither company has yet released their products. A positive note for the future is that both companies are building the product around the JAIN specification, which should allow applications to be run on any JAIN Compatible network. "Write once, run anywhere." Sun Microsystems (2002)

### **Implementation Independent specification of Services**

The purpose of the Parlay API's is to allow services access to network core features. However because of the different ways the networks are implemented it is necessary to model services in an implementation independent way, to allow their deployment across many different networks Sties (2002).

In an ideal world all services would be compatible with all Application servers and all Parlay/OSA Gateways, but in reality, services will have to run on various platforms. In order to minimize the work involved in creating the services for any network “A Generic and Implementation Independent Service Description Model” Sties(2001) has been proposed.

### 3.4.6 Service Deployment

Service deployment is, at its most basic, getting a working application from the developer’s computer to the application server where it will be run. This application server could be part of the Telecoms private network, or owned by the developer. Separate “hosting” companies may even own the application servers.

In the HSS Service Creation Environment Model, the SCE is responsible for passing on the finished application to the application server. This limits third party access to the servers and may be necessary if application server is inside the Telecoms internal network. This system would need lots of features to allow developers update applications already deployed, without having to start from the beginning. This system looks to be the most possible option for the security conscious network operators.

Using the MiLife SDK or simply writing an application from scratch means that a developer must then copy the application to the application server. Updating the application would mean removing the application and replacing it with an updated version.

### **The STARLITE Project**

The STARLITE project is part of the European IST program and has been working for several years in the service creation and management area. At a recent conference Triglia (2002) summarized what they had achieved.

“Fast provisioning of new services to end-users has rapidly become the key issue in the modern telecommunications environment, often being the most important element determining the success of operators and manufacturers. The convergence of IT (Information Technology) and TLC (Telecommunications) determines a framework for new, potentially appealing services, like, for instance, Intelligent Network (IN) capabilities offered to Internet (IP) users.

The Project has implemented an integrated IT/TLC environment based on a distributed IN, exploiting Parlay API specifications, as well as using Distributed Object and Mobile Agent Technologies (DOT/MAT). Above that integrated

Environment, cross-network services have been designed, created and deployed. These services are derived from the integration between distinct network capabilities (PSTN and Internet) and are uniformly controlled from creation to provisioning. DOT and MAT are used as a unified framework addressing the service lifecycle (creation, deployment, provision and operation), allowing the design and implementation of compact and extendable service objects, used in order to program active network nodes according to time-varying needs. Parlay APIs are used in the Project, as intended by the Parlay Group, i.e. to allow service developers to create services in an open network-programming framework.”

## 4 Future Directions

### 4.1 *Middleware*

The aim of the M-Zones project is to develop a management system for multiple ubiquitous environments referred to as smart spaces. Present ubiquitous computing environments such as Active Campus (2002), Oxygen (MIT, 2003) and EasyLiving (Microsoft, 2003) have each been independently developed with no consideration of interoperability. These ubiquitous environments were not designed to be compatible with each other so a management system developed these would be specialized and not generic. It would be of great benefit if these ubiquitous environments were developed from a common middleware infrastructure leaving the heterogeneous nature of these ubiquitous environments transparent to the management system.

At present, research in the area of middleware infrastructure for services in ubiquitous computing environments is in its infancy. There are, however, some encouraging developments in emerging technologies such as Sun Microsystems Jini (Sun, 2003) platform which is a distributed computing framework that allows users to access any resources available on the network without the need of any configuration. Other following technologies have also been proposed for developing application to support context aware middleware: XML , SIP and SOAP . These technologies are being used in the development of Web Services, which is a way of describing and using services from anywhere in the internet, in a common way. Middleware providers such as CORBA and .NET are adding support for these technologies, which is pushing present day middleware closer and closer to complete interoperability and ubiquity. As part of the M-Zones project it would be of great benefit if the three smart spaces that are being developed at Trinity College Dublin, Cork Institute of Technology, and at Waterford Institute of Technology, were based on a common middleware infrastructure. This would make developing a management system for managing multiple smart spaces (i.e. M-Zones) a lot easier. Possible technologies that would be of benefit to M-Zones would include the use of Sun's Jini platform as a middleware for device communication within a smart space, and Web Services as a method of smart space to smart space communication.

### 4.2 *Parlay/OSA*

The issue of how to create, manage and deploy services for 3G or other wireless networks could become a problem once the network operators start to open their networks to third party services and applications. The central issue is security, and how much control the network operator will maintain over these services. It is still too early to tell, but it is likely that the network operators (O2, Vodafone etc) will incorporate complete SCE (Service Creation Environments) into their network. Developers will have to use this environment to develop applications for their network and will not have direct access to the Parlay interface.

The implication for the M-Zones project of exclusive integrated Service Creation Environments is that it is more difficult to exchange data with other telecom networks than it should be simply using the Parlay/OSA API's. The M-zones project should allow for a lot of movement in the Parlay/OSA API's if they are to be used, as the future of Parlay is by no means certain.

The Nomad Project, which has links to the M-zones programme, is developing a Service platform which plans to link to a Parlay/OSA Gateway. Any further research into developing services that will integrate with Telecommunications Networks, through a Parlay gateway or other means, will need to be done in partnership with the Network Operators.

### 4.3 Agents

The domain of applicability for agents is extensive. Much of the analysis of agents in this paper has leaned towards network management, largely due to the fact that NM is a long-established and widely addressed area. Within this area a number of relevant future directions exist. Using agents as an enabling technology for active networks is one such area. Active networks not only allow the network nodes to perform computations on the data but also allow their users to inject customised programs into the nodes of the network, that may modify, store or redirect the user data flowing through the network. Mobile and intelligent agents could easily assume the role of these active network programs (Karnouskos 2002), traversing the network and performing customised processing of data received at network nodes. Another, more definite, direction is incorporating agent properties such as mobility, intelligence, etc into existing management technologies like SNMP. This would provide decentralized SNMP management while maintaining the existing and widely used management protocol.

Moving away from Network Management, Service management presents, possibly, the most interesting area of application for agents in regard to M-Zones. Again the mobility and intelligent facets of agents can provide many benefits to smart spaces such as more adaptive and nomadic services. Services that can customize themselves based on a variety of variables, that can be intelligently composed from a number of other services and that might move/migrate on request (possibly with a user).

As has been stated the range of applications for agents in smart spaces is extensive. How agents will be used, if even at all, is very much an open question, but the fact that agents hold potential for smart spaces is undeniable.

## 5 Conclusion

This paper began by introducing the main concepts in best practice middleware technologies. Following this, a definition of requirements for Smart Spaces with regard to middleware technology were highlighted. These requirements were kept generic (e.g. scalability, re-configuration, quality of service) in order to allow the identification of criteria for the evaluation of middleware technology. A detailed discussion of three technologies namely OSA/Parley, Jini and Agents addressed how these could be used within smart spaces or managed zones of smart spaces (M-Zones). These three technologies address different aspects of smart spaces, satisfying different middleware requirements. What is evident is that no single technology reviewed completely satisfies the requirements outlined in the analysis section. This paper concludes that present best practice middleware will have to expand its horizons to cater for the new requirements outlined in the analysis section if they want to survive in the emerging smart space environments (Kurt Geihs, 2001).

In summary it is not clear at this point exactly what Smart Space middleware will consist of. This remains very much an open question. What is likely, however, is that various smart space architectures will emerge independently of each other which greatly increases the need for middleware to provide interoperability between heterogeneous systems.

## 6 References

- Aberdeen Group 2002, Mobile Middleware Market to Triple by 2006, Retrieved November 27, 2002 from  
<http://www.pdastreet.com/articles/2002/7/2002-7-30-Mobile-Middleware-Market.html>
- Active Campus, Visited November 27, 2002 at  
[http://www.jacobsschool.ucsd.edu/newsletter/winter2002/active\\_campus.html](http://www.jacobsschool.ucsd.edu/newsletter/winter2002/active_campus.html)
- Bakken, D.E., "Middleware", Encyclopedia of Distributed Computing, Kluwer Academic Press, 2002.
- Bacon, J. et al., "Generic Support for Distributed Applications", Computer, March 2000, pp. 68-76
- Becker, C. & Geihs, K. "Generic QoS Support for Corba," Proc. Int'l Symp. Computers and Communication (ISCC 2000), IEEE CS Press, Los Alamitos, Calif., 2000, pp. 60-65.
- Bernstein, P.A. "Transaction Processing Monitors", Communications of ACM, November 1990, 33(11): 75 - 86.
- Bessler, A.V., 2000, A Service Platform for Internet-Telecom Services using SIPS.
- Breugst, M., Hagen, L. & Magedanz, T., 1998, 'Impacts of Mobile Agent Technology on Mobile Communications System Evolution', IEEE Personal Communications, vol. 5, no. 4, pp. 56-69.
- Bush, V. 1945, 'As We May Think', The Atlantic Monthly, July.
- Cabri, G., Leonardi, L., Zambonelli, F. 1999, 'Network Management based on Mobile Agents using Programmable Tuple Spaces', Proceedings of the 4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, April.
- Campbell, A., G. Coulson, and M. Kounavis. "Managing Complexity: Middleware Explained." IT Professional, IEEE Computer Society, 1:5, September/October 1999, 22-28.
- Capra, L., W. Emmerich, and C. Mascolo: 2001, 'Middleware for Mobile Computing: Awareness vs. Transparency (position paper)'. In: Int. 8th Workshop on Hot Topics in Operating Systems.
- Emmerich, W.: 2000, Engineering Distributed Objects. John Wiley & Sons.
- Erzberger, M., Altherr, M., "Every DAD needs a MOM" Message-Oriented Middleware. September 1999,  
<http://www.softwired-inc.com/pdf/technology/momdad-final.pdf>

Foundation for Intelligent Physical Agents, Available: [www.fipa.org](http://www.fipa.org)

Turing, A. M. 1950, 'Computing machines and intelligence', *Mind*.

Franklin, S. & Graesser, A. 1996, 'Is it an Agent, or just a Program: A Taxonomy for Autonomous Agents', *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag, 1996

Gelernter D., *Generative Communication in Linda*. *ACM Computing Surveys*, 7(1):80 – 112, Jan 1985

Grasshopper, Available: [www.grasshopper.com](http://www.grasshopper.com)

Gurer, D., Lakshminarayan, V., Sastry, A. 1998, 'An Intelligent Agent Based Architecture for the Management of Heterogeneous Networks', *Proc. of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*.

Gurer, DW, Khan I, et al., 'An Artificial Intelligence Approach to Network Fault Management.', [http://www.sce.carleton.ca/netmanage/docs/An\\_AI\\_Approach.pdf](http://www.sce.carleton.ca/netmanage/docs/An_AI_Approach.pdf)

HSS, 2002, Hughes Software Systems, *Convergent network Solutions-Service Creation Environment* Available at [http://www.hssworld.com/hss\\_mindsystem/archives/sce/docs/sce.pdf](http://www.hssworld.com/hss_mindsystem/archives/sce/docs/sce.pdf)

ISO/IEC 13818-2 (MPEG-2 video), November 1994

ITU-T Rec. H.263, "Video Codec for Low Bit rate Communication". 1996.

Jagot, A.R., 2002, *A Meta- Service Creation Environment for the Next-Generation Network (NGN) Centre for Telecommunications Access and Services*, University of the Witwatersrand, Johannesburg

Kampis, G., 'The Natural History of Agents', <http://hps.elte.hu/~gk/Mirror/Agents.pdf>

Karnouskos, S. 2002, 'Realization of a secure active and programmable network infrastructure via mobile agent technology', *Computer Communications Journal*, Volume 25, Issue 16, pp. 1465-1476, October 2002

Kenneth Black et al, "Wireless Access and Terminal Mobility in CORBA", *OMG Meeting in Dublin, Ireland*, November 13-14, 2001

Kurt Geihs, "Middleware Challenges Ahead", *IEEE computer*, 2001

Lauff, M., Gellersen, H.W., "Adaptation in a Ubiquitous Computing Management Architecture". *Proceedings of the ACM Symposium on Applied Computing*. pp. 566--567, 2000

Li, B. et al, "A Control-based Middleware Framework for Quality of Service Adaptations", IEEE Journal of Selected Areas in Communications, Special Issue on Service Enabling Platforms, vol. 17 no. 9, pp. 16632-1650, Sept. 1999

Lucent, 2002, MiLife™ Intelligent Services Gateway - the operator need for service Mediation, Available at [http://www.lucent.com/livelinek/0900940380020c61\\_Brochure\\_datasheet.pdf](http://www.lucent.com/livelinek/0900940380020c61_Brochure_datasheet.pdf)

Maes, P. 1994 'Agents that Reduce Work and Information Overload', Communications of the ACM, vol. 37, no. 7 (July), pp. 31-40.

Maffeis S., "The Object Group Design Pattern," Proc. Conf. Object-Oriented Technologies and Systems (COOTS 96), Usenix, Berkeley, Calif., 1996, pp. 294-303.

Maffioletti S., "Requirements for an Ubiquitous Computing Infrastructure" <http://diuf.unifr.ch/~maffiole/Documents/Papers/ParadigmForUbiComp.pdf>, 2001

Magedanz, T., Rothmel, K. & Krause, S. 1996, 'Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?', Proceedings of INFOCOM'96, San Francisco, USA, March 24-28.

Mahmoud Naghshineh, Marc Willebeek-LeMair, "End-to-End QoS Provisioning Multimedia Wireless/Mobile Networks Using an Adaptive Framework", IEEE Communications Magazine, no. 11, November 1997 pp. 72-81

Manuel Román et al, "Gaia: An OO Middleware Infrastructure for ubiquitous Computing Environments", 5th ECOOP Workshop on Object-Orientation and Operating Systems (ECOOP-OOSWS'2002), Malaga, Spain, June 11th, 2002

Mascolo, C., Capra, L., Zachariadis, S. and Emmerich, W., "XMIDDLE: A Data-Sharing Middleware for Mobile Computing". In Personal and Wireless Communications Journal, Kluwer. April 2002.

McCulloch, W. & Pitts, W. 1943, 'A Logical Calculus of the Ideas Immanent in Nervous Activity', Bulletin of Mathematical Biophysics, vol. 5, pp.115-133.

Microsoft's Easy Living project, Visited November 27, 2002 at <http://research.microsoft.com/easyliving/>

Milojicic, D. et al. 1998, 'MASIF---The OMG Mobile Agent System Interoperability Facility', Personal Technologies, vol. 2, pp117—129, September.

MIT Oxygen project, Visited November 27, 2002 at <http://oxygen.lcs.mit.edu/>

Mobile Agent System Interoperability Facility,  
Available: <http://www.fokus.gmd.de/research/cc/ecco/masif/>

Murphy, A. et al, "LIME: A Middleware for Physical and Logical Mobility", The 21st International Conference on Distributed Computing Systems, April 16 - 19, 2001 Mesa, AZ

Nahrstedt, K., et al. "QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments". IEEE Communications Magazine. 2001.

Nahrstedt, K., et al. "Distributed QoS Compilation and Runtime Instantiation" In Proceedings of the Eight IEEE/IFIP International Workshop on Quality of Service, pp. 198-207, June 2000.

Nana, P., 2002, Re-usable TINA service components incorporating the Parlay API for the Next Generation Network (NGN) Centre for Telecommunications Access and Services (CeTAS) University of the Witwatersrand, Johannesburg

Norman, D., The invisible Computer. The MIT Press, Cambridge, Massachusetts 0214, 1999

Nwana H. 1996, 'Software agents: An Overview', Knowledge and Engineering Review, vol.11, no. 3, pp. 205—244.

Petersen, K., M. J. Spreitzer, D. B. Terry, M. M. Theimer, and A. J. Demers: 1997, 'Flexible Update Propagation for Weakly Consistent Replication'. In: Proceedings of the 16th ACM Symposium on Operating Systems Principles (SOSP-16). pp. 288-301, ACM Press

Picco, G., Murphy, A. and Roman, G.-C.. LIME: Linda Meets Mobility. In D. Garlan, editor, Proc of the 21st Int. Conf. On Software Engineering, pages 368-377, May 1999

Raatikainen K. 2001. Functionality Needed in Middleware for Future Mobile Computing Platforms. Nov. Retrieved November 18, 2001, from <http://www.cs.arizona.edu/mmc/02Raatikainen.pdf>.

Roman,G-C. Christine Julien, "Using EgoSpaces for Scalable, Proactive Coordination in Ad Hoc Networks", Dept of Computer Science, Washington University in St. Louis, 2002

Roth, J., "A Communication Middleware for Mobile and Ad-hoc Scenarios", International Conference on Internet Computing (IC'02), 24.-27. June 2002, Las Vegas (USA), Vol. I, 77-84

Schmidt, D.C., Levine, D.L. and Mungee, S. "The Design of the TAO Real-Time Object Request Broker," Computer Comm. J., vol. 21, no. 4, 1998, pp. 294 324.

Schmidt, D., Levine D. and Cleeland, C. "Architectures and Patterns for High performance, Real-time CORBA Object Request Brokers," In Advances in Computers, Marvin Zelkowitz, Ed., Academic Press, 1999.

Schneier, B., 2001, Bruce Schneier Phone Hacking: The Next Generation, July 15, 2001 Available at <http://www.counterpane.com/crypto-gram-0107.html>

Sun Microsystems Jini project,

<http://www.sun.com/software/jini/overview/index.html> .  
Visited March 5, 2003

Sties P., Kellerer, W., 2001, A Generic and Implementation Independent Service Description Model, Institute of Communication Networks Munich University of Technology (TUM)

Sties P., 2002, A service creation model for network spanning services  
Peter Sties Institute of Communication Networks  
Munich University of Technology (TUM)

Sun Microsystems, 2002, Sun Microsystems :The JAIN API's  
Available at <http://java.sun.com/products/jain/overview.html>

Technical Standard a, "Distributed Transaction Processing: The XA Specification" C193 UK  
ISBN 1-872630-24-3, February 1992

Technical Standard b, "Distributed Transaction Processing: The TX (Transaction  
Demarcation) Specification", C504 UK ISBN 1-85912-094-6 April 1995

Triglia S., 2002, "STARLITE, a success story for Parlay", Parlay Meeting 9-11 July 2002,  
Montreal Canada.

TrueConverge, 2002. TrueConverge Service Creation Environment  
Available at  
[http://www.truetel.com/download/TrueConverge\\_Service\\_Creation\\_Environment.pdf](http://www.truetel.com/download/TrueConverge_Service_Creation_Environment.pdf)

Vanegas R., et al., "QuO's Runtime Support for Quality of Service in Distributed  
Objects," Proc. IFIP Int'l Conf. Distributed System Platforms and Open  
Distributed Processing, Springer-Verlag, New York, 1998, pp. 207-223. IEEE  
Personal Comm., vol. 4, no. 5, 1997, pp. 58-64.

Wolfgang Emmerich, "Software Engineering and Middleware: A Roadmap" In: A.  
Finkelstein (ed): The Future of Software Engineering. pp. 117-129. ACM  
Press. 2000.

Yau, S.S. and Karim F., "Reconfigurable Context-Sensitive Middleware for Pervasive  
Computing", IEEE Pervasive computing, 2002

Zapf, M., Herrmann, K. & Geihs, K. 1999, 'Decentralised SNMP management with mobile  
agents', sixth IFIP/IEEE International Symposium on Integrated Network Management.

Zinky, J. , Bakken, D. and Schantz, R., "Architecture Support for Quality of Service for  
CORBA Objects," Theory and Practice of Object Systems, vol. 3, no. 1, Jan. 1997.