Resources Discovery and Management Using Policies in Smart Spaces

Samir Ghamri-Doudane*, Sven van der Meer**, Robert O'Connor**,
Yacine Ghamri-Doudane*, and Nazim Agoulmine***

*LIP6 Laboratory, University of P & M Curie, 8 rue du Capitaine Scott, 75015 Paris, France.
{sghamri, ghamri}@rp.lip6.fr

**Telecommunication Software and Systems Group (TSSG), Waterford Institute of Technology (WIT), Cork
Road, Waterford, Ireland.
{vdmeer, roconnor}@tssg.org

*** Complex Systems Laboratory, LSC, University of Evry, 40 rue du Pelvoux, 91020 Evry CEDEX, France.
nazim.agoulmine@iup.univ-evry.fr

Abstract

This paper presents a new hierarchical Policy Based Management (PBM) architecture that facilitates the description and automatic discovery of services and devices in Smart Space environments and the specification of high-level policies that control access to these resources in coexisting Managed Zones (M-Zones).

The approach is performed in two phases. The first one consists of defining a generic information model of all resources that can be managed in such a space and a specification of a set of high-level policies to express the access control policy of each M-Zone. Both models are specified using the DMTF CIM. (For the next iteration, it is planned to use the DEN-ng model, extended with new specific objects). The second phase is more dynamic and uses a novel discovery protocol based on COPS to identify the various components that are present in a particular environment. This protocol is named COPS-SD (COPS extensions for Service Discovery usage).

1 Introduction

A Smart Space is a physical space rich in devices and services that is capable of interacting with people (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.

A Managed Zone (M-Zone) is representation that allows Management to be performed on one or more Smart Spaces. An M-Zone represents management domains of Smart Spaces, between which participants/devices may roam and dynamic service provisioning and adaptable services are realised. However, regarding the dynamic and heterogeneous aspects of these environments, a policy-based management (PBM) approach should be adopted to support usage, operation, control, administration and maintenance of multiple M-Zones.

One of the fundamental problems that arise in such environments is the description and discovery of services and devices coupled with securing their access. Indeed, as the communications medium used is generally radio-based, anyone with an appropriate device can capture the signal and possibly interact with the Smart Space, using internal unauthorised resources. The introduction of policies in the discovery architecture provides secure and limited access to available resources, as it must comply with a preset strategy. Another characteristic of this solution is the hierarchical abstraction, which is motivated by the fact

that the M-zone is decomposed in several Smart Spaces. Note that this decomposition allows large scale handling of dynamic spaces.

2 M-Zones

The term M-Zones is used to denote a Managed Zone and describes the core of a four-year research programme funded by the Irish Higher Education Authority (HEA). This programme is lead by the Telecommunication Software and System Group (TSSG) based in Waterford Institute of Technology (WIT), with Trinity College Dublin (TCD) and Cork Institute of Technology (CIT) as partners. The main objective of the M-Zones research programme which began in summer 2002, is to "undertake fundamental research into novel management infrastructures to enable collaboration and management between and within Smart Spaces". Smart Spaces are 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. For more information about the M-Zones research programme please refer to www.m-zones.org.

2.1 What is a Managed Zone?

A Smart Space is a physical space rich in devices and services that is capable of interacting with people (also known as users or actors), the physical environment and services originating outside the Smart Space. The aim of the Smart Space is to orchestrate the use of the integrated physical and computing environment to bring tangible benefits to people in support of their tasks. A task is the description of a user's intention that has a defined outcome. A task can be supported by a single service or a set of Smart Space services. It may represent a short or long lived set of activities. A Smart Space Service is a service offered to a person or computing entity in a Smart Space. The general principle, when interacting with people, is to use interaction paradigms that are natural and familiar to the people engaging in their tasks. The general principle, when interacting with other services, is to establish knowledge to fulfil such tasks. [1]

Based on this model of a Smart Space and its services, the management of Smart Spaces is represented in form of Managed Zones (M-Zones). An M-Zone denotes a domain for the management of one or more Smart Spaces, unrelated (at least apriori) to a physical space or organisational boundary. Smart Space Management is the collection of all activities that ensure secure and effective usage, operation control, administration and maintenance of a Smart Space, including all its devices and smart services as well as relationships to external devices and services. An M-Zone represents domains of management in which Smart Spaces can be managed and between which participants and devices roam, and dynamic service provisioning and adaptable services are realised. [1] [2]

The primary goal of Smart Space management, and therefore of an M-Zone, is to prepare a Smart Space and its services to be used and operated in a stable, secure and efficient manner. To guarantee this goal for long time operations, they need to be controlled, administered and maintained in their entirety and support the general aim of the Smart Space and its components. The terms use and operation describe the part of the management that is seen by users and customers. They are concerned with the system's ability to serve them. A company running a system relies on its efficient operation in order to generate revenue. This operation is supported by the effective control of the system. The term control describes the brief but reoccurring task of keeping the system stable to serve its customers and to generate revenue. E.g. the configuration of system components to record data for accounting purposes. Administration and maintenance reflect the long-term operation and control of a system. [2]

2.2 Why is it complex to manage?

The complexity in managing Smart Spaces has many dimensions. "the source of the most serious challenge to deploying ubiquitous computing environments … not technological but structural" [3] [4]. The management of these environments (Smart Spaces) presents a structural problem.

Current research has focused on supporting single applications in intensively instrumented spaces, reducing many of the management problems, which arise only from more general scenarios. The MIT Media Lab, for example, has conducted extensive work on highly enabled spaces for children (KidsRoom); INRIA have focussed on smart offices (INRIASmartOffice); the University of Reading is concerned with invisible building control (WarwickImplants); and a group at the University of Lancaster have developed an electronic context-aware tourist guide (TouristGuide). Some major corporations are also active in the applications area, with major initiatives being announced by (amongst others) Sun, Microsoft, Ericsson, Nokia, Orange and Intel.

Such single-application spaces have significantly simpler management requirements than the general case of a managed zone, since it is possible to constrain the nature and behaviour of the devices in the space very closely. The general case - where the population of devices in a space or multiple spaces is highly dynamic and largely unconstrained - opens up major research challenges. Without a structured approach to management these challenges must be handled ad-hoc by individual architectures or applications - a phenomenon we may observe in a number of current projects building smart-space infrastructures (ContextToolkit and MetaGlue).

More interestingly, the behaviour of devices in a Smart Space is affected by a number of factors not directly related to the devices themselves, but rather determined by the use of the space in terms of task allocations and architectural morphology. Knowledge of the tasks undertaken within a space can be used to influence management decisions about when anticipating what devices are likely to appear and their requirements (CoenStopWorrying). Equally, the way in which an architect designs a space (or set of spaces) radically affects the movement potential, communication patterns etc. which are possible within it (SpaceIsTheMachine). This makes managing the Smart Space a major challenge in holistic systems engineering as well as in telecommunications.

The real challenge is to enable the dynamic integration of Smart Spaces, which enable mobile people and devices to roam across such spaces whilst maintaining communication and information services. Mobility is a defining factor of the way people are working and living today. The ability to dynamically integrate private (e.g. within a home or office), semi private (e.g. within public transport vehicle, library) and public (e.g. park, square) Smart Spaces is a key element in realising effective mobile working and living.

The major problem for the realisation of Integrated Smart Spaces is the lack of an open management infrastructure within and between such Smart Spaces. DARPA (Defence Advanced Research Projects Agency, USA) has singled out the (network) management problem inherent in managing Smart Spaces/embedded systems as the most important challenge facing telecommunications service managers for the next decade.

2.3 What do we want to manage?

As managed Smart Spaces are interconnected, the task of management is to ensure that the seamless flow of actors (people and devices) across environments is provided. It is necessary to manage seamless interoperability and cooperation between Smart Spaces and between managed zones, as described by the terms of intra and inter Smart Space management.

The management requirements of Smart Space environments are:

- *Semantic driven service discovery and composition*: The ability to compose complex services from the basic atomic services offered by available resources is of great

importance to the usability of the Smart Space by the end user and other resources within it. Such a procedure would be specialised based on the composite service requirements and the services available at present within the Smart Space.

- *Adaptive Information Delivery*: using knowledge of people, place and devices is an important issue for the development of seamless M-Zone interoperability. The issue of what information is accessed from an external M-Zone and how this information is presented to the end user is an important management consideration. When roaming, information such as user profile information may be made available to the visiting M-Zone to support user service requirements. The transformation of information from one zone to another and the management of the persistence and integrity of information is a major management challenge.
- *Performance*: M-Zones must co-operate to support user performance requirements as they migrate from one M-Zone to another. A common example is the mobile phone scenario where users who roam expect that there is no loss of service quality as they move from one operator domain to another. Unlike the single service scenario of today's mobile phone network, the user will require access to a set of services.
- *Configuration*: As in the mobile phone environment M-Zones need to be configured to support user (people and devices) into and out of M-Zones. Issues such as IP address allocation and bandwidth allocation need to be addressed against the user service requirements.
- *Business Process*: Just as is the case in the development of solutions for the development of interoperable telecommunications networks a business process model can be developed to support the interoperability of M-Zones [5].

Several infrastructure concepts that underpin inter-domain management in telecommunications are also very valuable in realising co-operating M-Zones. These concepts include Service Level Agreement negotiation, policing and enforcement, as well as real time selection, deployment and enforcement of operational policies within a managed area.

2.4 Technical Environment

2.4.1 Network Mobility

The integration of Mobile and Cellular IP is used as a mobility solution with similar characteristics to the roaming requirements identified for smart environments. A basic hierarchy can be identified in Smart Spaces in which an administrative domain contains one or more Smart Spaces (e.g. a university may contain canteen, library and lecture hall Smart Spaces). Also a Smart Space will typically contain one or more cells. When considering Smart Spaces there are three types of roaming that may be identified. Firstly, there is the situation where mobile nodes move between cells within a space. This can be described as intra-space roaming. Secondly, roaming may occur between separate and distinct Smart Spaces within the one administrative domain. This may be termed intra-domain roaming. Finally, roaming between administrative domains is also considered. This is termed inter-domain roaming.

A Cellular IP network contains paging areas, which are a grouping of one or more network cells. Smart Spaces, which are delimited by physical space, also contain cells. E.g. a lecture hall may be an individual Smart Space offering lecture hall services that are specific to that space. In practical terms a large lecture hall may require more than one network cell to cover the entire room. Hence, Cellular IP could facilitate roaming within individual spaces. Similarly the Cellular IP network as the overall administrative domain could provide roaming between Smart Spaces.

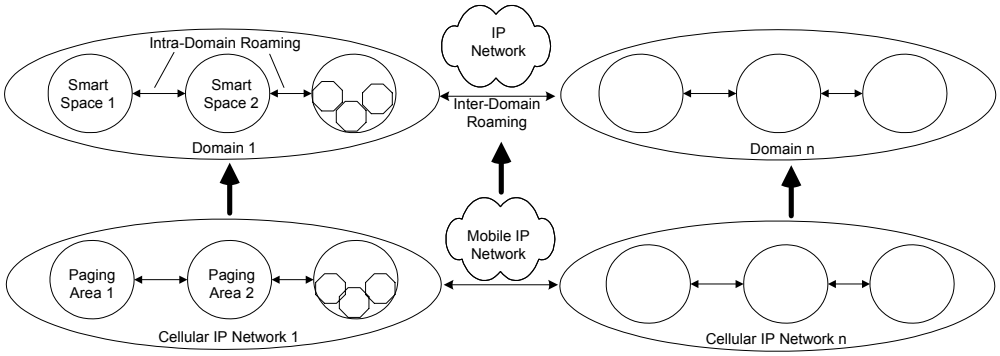

Figure 2-1 - Cellular & Mobile IP / Smart Space Mapping

In these respects, Cellular IP fulfils the intra-space and intra-domain roaming requirements of Smart Spaces. Inter-domain roaming is facilitated through the use of Mobile IP. Hence, through the co-operative use of both Mobile and Cellular IP, network-level mobility of users/devices throughout pervasive environments is achieved. [6]

2.4.2 Smart Space Components and Resource Discovery

The service component of a smart resource is made up of various internal components. These components allow a resource to publish its functionality to a Smart Space on entering the space and allow other smart resources to understand and invoke these services in an abstract way. A reflective state component of a smart resource contains representation of the device's own internal states. The reflective state component contains various internal components that allow the smart resources to publish a Reflective Meta Representation (RMR) of itself so other smart devices and services may be able to query the present state of the smart resource, invoke internal actions and react to changes.

If a smart device wishes to participate in invoking a smart device service, or subscribe to state change notifications, it must contain a *Control Component*. The control component effectively acts as a client of the services smart devices offer to the Smart Space. The control component contains various internal components, which allow it to interact with smart devices:

- Service Publication Listener - listens out for any new services being introduced by a smart device entering a Smart Space. It also would handle such things as leasing and deregistering of smart device services.
- Query Services - can query its registry of available services for a service of a particular type. It can also query the registry of other control points or broadcast a search query to available smart devices within the Smart Space.
- Invoke Services - can invoke services belonging to other smart devices within the Smart Space by binding with them through this component. An example of how this component may work can be seen in WSDL web service binding through the use of SOAP and URLs.
- Subscribe RMR (reflective meta representation) - When a device enters the Smart Space, it can publish its reflective meta representation to any interested smart devices and services containing a Control Component. The Control Component uses this internal component to listen and subscribe to the smart devices RMR publications. This component can also request an RMR of a smart device. The Control Component will then store a list of the RMRs it's subscribed to in the Available RMR Subscriptions component.
- Manipulate RMR - The Control Component can manipulate the RMRs it is subscribed to. Through this process the change in an RMR value will be sent to the

corresponding smart device, which will then reflect this change through the use of internal operations. This operation also requires a binding between the smart device and the Control Component.

- Available Services List - The Control Component must hold a list of all smart device services available to it within the Smart Space. This list can be queried for services of a certain category, returning information on how these services are to be invoked, i.e. parameters, location, device, etc.
- Available RMR Subscriptions - The Control Component must hold a list of smart device RMRs it is currently subscribed to. This component holds information on how to update any given RMR of a smart device.

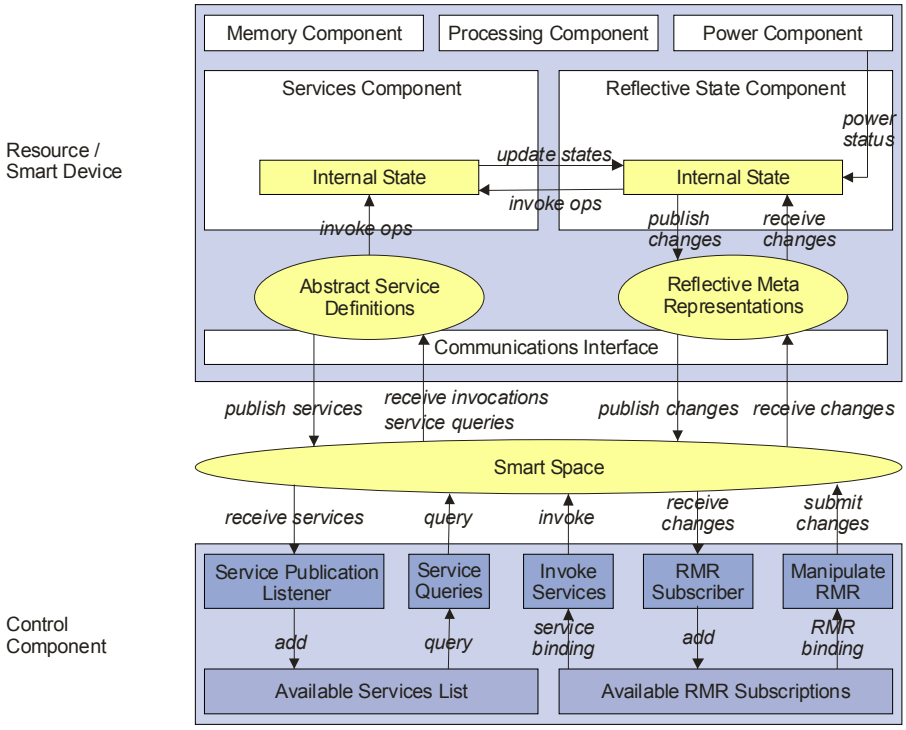

Figure 2-2 - Service and Resource Discovery

A *reflective state component* of a smart device can be taught of a Meta representation of the devices own internal states. The reflective state component contains various internal components that allow the smart device to publish a RMR of itself so other smart devices and services may be able to query the present state of the smart device and react to changes. The main internal component of the reflective state component is the internal states component. This component will hold values on various states the smart device can take on, e.g. POWER ON/OFF/LOW, ALARM RINGING, PRESENT_TIME, TEMPERATURE, etc. The internal states may be manipulates by the invocation of an internal operation of the smart device, or may be reflecting the status of a physical component such as power supply or bandwidth. Smart devices and services can subscribe the RMR of a smart device and manipulate it, through the Control Component. Manipulation of the RMR will change the internal operation of a smart device. For example an alarm clock may have a RMR showing that the alarm functionality is turned off. If a Control Component of a smart device or service were to set this attribute to on, the necessary internal operations of the smart device would be invoked to reflect this change. [7]

2.4.3 Service Discovery Protocols

Service discovery focuses on locating available services within a Smart Space or a managed zone, respectively. Currently available mechanisms and protocols for service discovery vary greatly depending on the environment they have been specified for. Service discovery can be completed with a single database by the DySCo system [8], which uses a single component for keeping a database of all available services and the role they can play in a compound service. A comprehensive discussion of service discovery for ubiquitous computing environments can be found in [13]. Besides the introduced protocols, some device and vendor specific protocols are available, such as HP JetSend and HP Chai. A comprehensive discussion of service discovery within managed zones can be found in [14]. For detailed introduction of UPnP and HPnP from an M-Zones perspective, please see [7].

For larger scale service discovery, e.g. in a Smart Space rich on devices and services or in a managed zone, multicast following the "shout and reply" paradigm might not be suitable anymore. Here, hierarchical lookup systems (e.g. DNS) are the state of the art for discovering services and resources in a scalable way (e.g. HLP's eFlow system [9]). In mobile and ad hoc networks, the problem becomes even more complex. The Anamika architecture [10] uses Bluetooth based discovery mechanisms to locate services within the range of a requesting node with recursively forwarded requests until the desired service is found. UPnP and JINI represent a possible future direction for operating system network stacks. They provide service advertisement, service discovery and other network management protocols (such as NAT traversal) to applications.

UPnP-enabled devices use a UDP based multicast version of HTTP to retrieve information about other UPnP-enabled devices connected to the network [11]. Answers are transmitted by a unicast UDP variant of HTTP. Many operating systems support UPnP (e.g. Microsoft Windows XP) as well as many smart devices. UPnP represents a non-centralised mechanism for service discovery. The multicast facets may prove impractical for large networks but it is still suitable for small Smart Space installations.

JINI is a framework based on Java, with similar objectives as UPnP. Service discovery happens via a lookup service, which represents the central database of available services. Hostname and port number of this service can be retrieved initially by a simple unicast request. Multicast routing can be used to identify remote lookup services [12].

3 Management by Policies

Policy based networking [15] is a well emerging and alternative technology in network management and control. The principle is to conceive this management as objectives integrated in a decision system. These objectives are defined in the form of clear and human-understandable rules, called management policies [16]. Then, these rules are distributed by the administrator over all the realization points in order to be applied locally. These points are network or terminal equipments. Unlike the traditional approach, this method allows concentrating the management on global operational aspects rather than equipment's specific aspects. The flexibility of this approach is vital for the reason that it allows fast and easy changes in the business strategy without the necessity of interacting with each subordinate in the network.

In order to reach a fast deployment of a new business level strategy, the definition of the policy alone is not sufficient. The network administrator needs a set of tools which allow him to set up these high level decisions. This installation must operate in a flexible way and without precise knowledge on the specificities of the equipments deployed in the network. Thus, a policy based management system [15] includes four main elements allowing the definition, the representation and finally the realization of policy rules as illustrated in Figure 3.1.

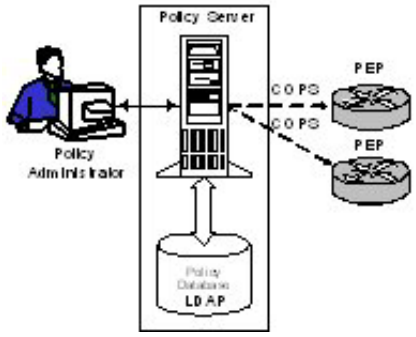

Figure 3-1 - Policy Based Management System

The **policy management tool** is an interface allowing the network administrator to interact with the management system. Indeed, through this tool, he can define and introduce the adequate strategy rules. This tool should facilitate a certain set of tasks such as:

- The definition and introduction of high level policies;
- The verification of syntactic, semantic integrity and consistency (to avoid conflicts) of these rules;
- The translation of each high-level policy into several low-level rules addressing the set of target network elements and systems;
- The storage of these policy rules in a policy repository and the detection of any change relating to these in order to notify the network elements concerned by this change.

Then, the **policy repository** is the place where the rules are stored by the administration tool. The repository may be an LDAP directory [17], a relational database or simply a web server depending on implementation choices. The policy rules as well as all information that allow the modelling of the control network element are represented into an information models, CIM [18]. CIM (Common Information Model) is an abstract model which aims to capture all information necessary for the control of an IT environment.

The **policy decision point** (PDP) is the entity responsible in taking decisions based on the policy rules stored in the repository. These decisions are the actions to be applied over the concerned network elements. Indeed, the PDP includes the following functions:

- The identification of all the rules which are applicable to the elements under its responsibility;
- The translation of these rules into PEP-understandable actions;
- The regular checking of the network state;
- The monitoring of any change in the policy rules in order to update its decisions.

The **policy enforcement point** (PEP) is the entity targeted by the defined policy rules. A PEP may be: a DiffServ-enabled router, an RSVP router, a security gateway, a proxy, or an user terminal. Its role is to execute the actions enforced by the PDP and, occasionally, to collect statistical information with the aim of sending them to the PDP.

This information exchange between the PEP and the PDP requires a management protocol such as: COPS (Common Open Policy Service), SNMP (Simple network Management Protocol) or any other similar protocol.

COPS [19] is a management protocol defined specifically to support signalling between policy elements. The PEP is the policy client while the PDP acts as a server in a request / decision paradigm. Due to reliability, this protocol uses TCP to send requests and bring decisions. Hence, it offers secure mechanisms such as: authentication, integrity control and data protection.

COPS provides a generic management model and is known as a flexible protocol. Indeed, it can be applied to several policy domains such as: QoS provisioning policies, access control policies (to the network and its resources), etc. Therefore, to reach each of these fields, one should define an extension of COPS by adding appropriate objects and adapting its operation to the target management area. These extensions are called: COPS clients. Till now, there are two standardized models (clients) within the COPS protocol: COPS-RSVP for the outsourcing model and COPS-PR for the provisioning model. Other models for other purposes are proposed as IETF drafts.

4 Policy-based Approach for Smart Spaces

4.1 Objectives and Requirements

The main objective of this work is the integration of Smart Space management and hierarchical Policy Based Management techniques. The integration serves as a basis to develop a flexible management platform for intra and inter Smart Space management based on managed zones, which enables services and resources to be used, controlled, operated, administered and maintained in a unified way.

The second objective is to provide a unified management conception that is independent of the actual Smart Space technology it manages, thus developed for multiple purposes such as application area of the Smart Space (e.g. home, office, learning), the employed resource access technologies (e.g. UPnP, Jini, HAVi) and the underlying business goal of the Smart Space.

The third objective is to support the information mapping from an abstract business goal (expressed by policies) towards concrete commands (to control and access resources) enabling business-to-network translation (top-down) and network-to-business-translation (bottom-up). The bi-directional translation of information allows the business goal to be enforced within the Smart Space and, on the other hand, to adjust policies to actual conditions in the technical environment of a Smart Space.

The fourth objective refers to the acceptance of the chosen approach itself. The history of the Internet has taught that adoption is a better predictor than perfection. The acceptance of the presented approach is not given for the sake that it integrates Smart Space management and Policy Based Management. Instead, the approach will be used only if it integrates these two worlds in a reasonable manor, employs widely known and accepted methods and technologies and creates novel (management) functions or improves existing ones.

The general requirements for the integration are to enable efficient interworking, portability and scalability of Smart Space management. Interworking reflects the need to support mobile devices and services across Smart Spaces and even across M-Zones. Portability is introduced as a requirement because to provide Smart Space management independent of employed technologies. Scalability should prepare for being deployable in all sorts of Smart Spaces, ranging from simple (e.g. home) up huge installations (e.g. an airport). The Policy Based Management must provide mechanisms that support and enable the interworking between M-Zones, it must be portable towards several technical environments and it must provide mechanisms to be scaled for all purposes.

The nature of managed zones makes it necessary for the chosen approach to offer functionality for naming and addressing of resources and users as well as for their registration. Discovery and lookup services, within a managed zone, are key to provide users access to available services and resources. This access has to be controlled by means of authenticate and authorise users and grant access only if the business goal and related policies allow to do so. The communication between the Policy Based Management components and M-Zones components needs to be modelled in a simple way with regard to (probably) limited

processing power and connectivity. The support of automated control, administration, and maintenance must be based on formal descriptions of resources and policies for their access. Policy and profile services (or data and type repositories) can be mechanisms to realise this. The visualisation of components, interactions and data types is a requirement for the manual control of a distributed system.

The technical challenges can be seen in the interoperability of components, the control of resources and the definition of formats for data exchange and evaluation. Connectivity and reliability of services might be based on middleware technology. However, the approach has to ensure them explicitly.

4.2 Integrated Architecture

The proposed architecture identifies four planes. Each plane is dedicated to a specific problem context. Each problem context describes a dedicated viewpoint to the management of Smart Spaces. The planes are used to specify the different types of information that need to be defined and processed and the different levels of management functions that are needed. The approach of this layered architecture is chosen to ease the design of the architecture and to (clearly) distinguish the different functional aspects of the introduced approach. In other words, each of the four planes can be used to describe particular aspects of the architecture. The four planes are:

- Policy Based Management, covering the business model expressed by policies;
- Managed Zones, modelling Smart Space management functions;
- Services and Devices, modelling access to core Smart Space technologies;
- Network, identifying underlying network technologies for mobility support.

On the first view, the architecture separates policy processing from Smart Space management and the latter one from Smart Space and network technologies. Policies and management functions can be designed independent from Smart Spaces and actual Smart Spaces become transparent for the processing and enforcement of policies.

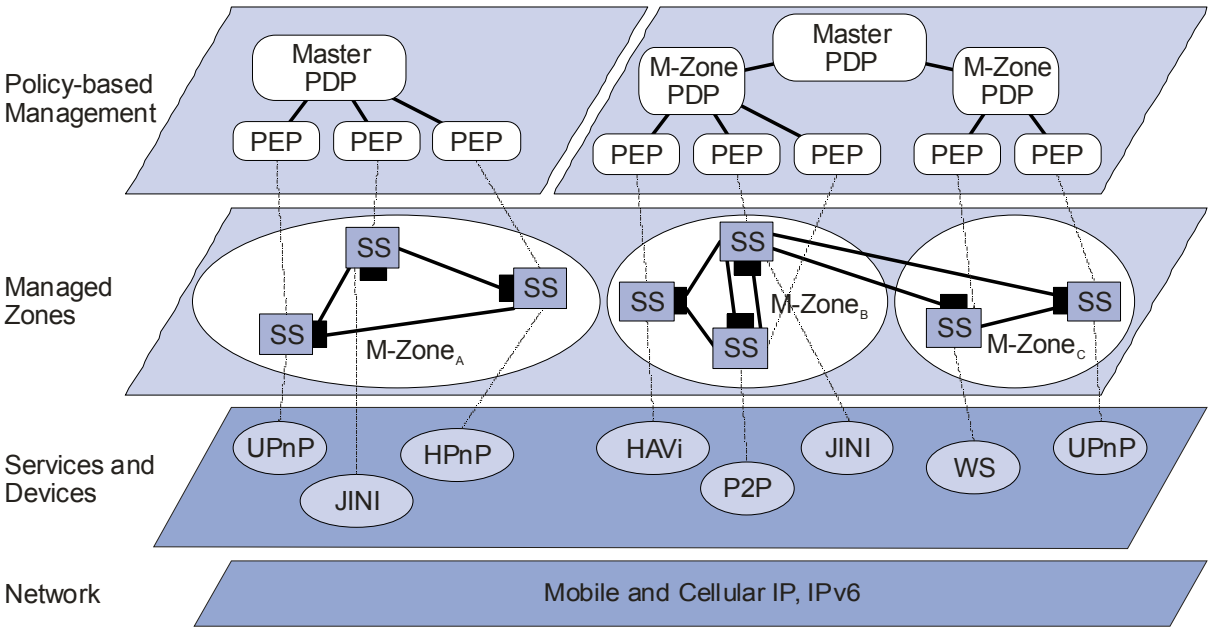

Figure 4-1 Hierarchical Architecture

However, the architecture must not been seen as a dogma. Components concerned with policy processing and Smart Space management might require direct access to underlying technologies. This is especially important for the management of resources. This direct access depends on the business model for the management system and thus of its actual design, which should not be denied to access the two lower planes directly.

In the vertical access, the figure shows the mapping between components of each plane. Here, a Master PDP is in control of one or more M-Zones, whereas an M-Zone PDP is responsible for a single managed zone. PEPs are employed to enforce policies within a Smart Space and (not shown in this figure) for resource access. On the Managed Zones plane, the policy information and processing is mapped to the actual Smart Space management system. A PDP is part of the core management component of an M-Zone. A PEP needs to be coupled with components that grant access to resources or specific management functionality. Resources and management functions are realised using a certain technology for devices and services, both are supported by networks providing quality of service and connectivity.

An M-Zone consist of logical integrated components. The Service Broker is the logical representation of the Control Point functionality outlined for resource discovery. It maintains a list of all services/devices currently deployed in the Smart Space. The User Agent is the logical representation of the User. The User Agent provides a link to a repository of personal information about the User. The Communications Manager is the central point for all communications in the Smart Space. It is deployed in a location that is accessible to all services and resources within a Smart Space. Whenever an entity wishes to perform an operation on the Smart Space, its services or resources, the Communications Manager is the initial point of contact. We apply two paradigms to an M-Zone:

PUSH over PULL Model - Often when referring to Smart Spaces, it is often said the Space moulds itself so that it best suits the user. E.g. The Lecture Hall detects that a Lecturer User is in the room and accordingly offers them different services than the Students. From a management perspective, this level of autonomy is quite complex to model. This involves information PULL by the Smart Space from the User. However, if this information PULL is changed to an information PUSH by the User to the Smart Space, the intelligence level required on the part of the Smart Space is reduced. E.g. The User informs the Lecture Hall that he/she is a Lecturer and wishes to avail of Lecturer services. The User determines the participation level with the Smart Space.

Management Functionality as Services - Different Smart Spaces require different levels of management complexity. E.g. A bank will require much heavier security functionally than a hotel lobby. The Integrated Management solution must reflect this. By treating management functions as Services, individual spaces can be configured with as much or as little management functionality as required. Management Services can be described in a similar fashion to other resources. This is facilitated through the Management Service Broker.

4.2.1 Integrated Components

The integration of the Policy-based Management with Smart Space management is shown in figure 4.2. The global access strategy is defined as a set of policies at the top level PDP (Master PDP) and then pushed to one or more M-zones. The method used for service discovery goes beyond that of a simple “shout and reply” multicast. The M-Zone PDP is directly coupled with the Communications Manager supporting the enforcement of the global strategy. The PEPs are directly coupled with the Service Broker and/or devices (resources) to allow the enforcement of the policies. Each device is registered with the Service Broker, which registers the devices with PEPs.

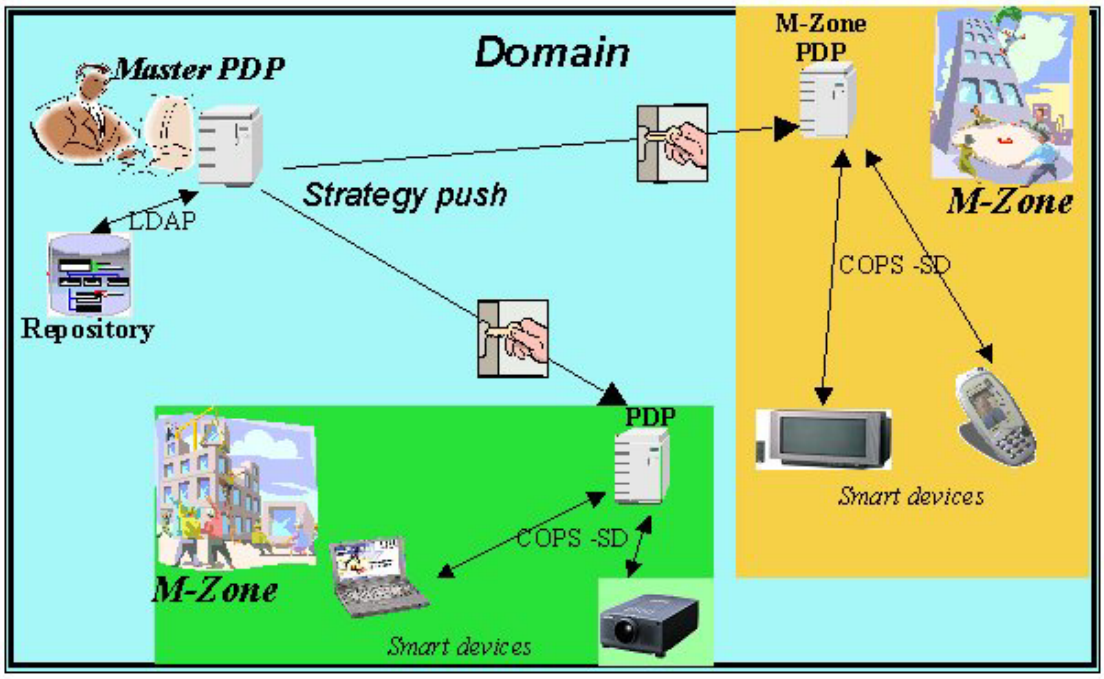

Figure 4-2- Integration of the Policy-based & Smart Space Management

Every resource and service is known to the Communications Manager, which forwards this information to the PDP allowing the latter one to have appropriate policies defined for the managed zone. The communication between the PDP and the PEPs is realized with COPS-SD, whereas M-Zones components might employ other messaging protocols to exchange information. Therefore, the components are only loosely coupled increasing the flexibility of a derived management system.

4.2.2 Policy Based Management

After having examined the major techniques in the area of service discovery, and having summarized their principal assets and weak points, we have noticed a lack common to all these methods: None of them has enough granularities to take into account the user level. Thus, any recognized user connected to the network can discover and use any available services. However, in a corporate network with various levels of users' authorization levels or in an operator network specifying several classes of service level agreements this concept becomes significant, and even necessary. Hence, each user should discover and use only the available services which he authorized to use. Therefore, these environments need to define a global strategy for controlling the service access in addition to initial discovery protocols. Thus, the service discovery becomes strongly related to the management of the services and resources access.

For these reasons, we decided to combine the mechanisms of service discovery with the policy-based management concept. In this case, the network administrator should introduce its strategy about the services usage (which users are authorized to use which services) and the discovery protocol will deal with the rest. Then, the user will only discover the authorized available services according to his profile. The network manager can define several user groups and profiles according to his strategy. This can be very useful in the case of existing "Hot Spot" networks. A customer with "Gold" profile will be able to discover and use any existing service, while a low profile customer will only discover a limited set of services. Traditional protocols were not designed to achieve this objective and will require major changes to achieve to support it.

In order to reach this target, we decided to extend the policy-based management architecture. Therefore, in addition to the introduction policies to automate the control of the environment, we had to define a new protocol to support service registry, advertising and discovery as well as the corresponding information model.

The proposed architecture follows the Policy Based Management architecture as defined by the IETF [15]. In accordance to figure 4.3, this architecture defines a set of components to enable policy rules definition, storing and enforcing. Policies are a set of pre-defined rules that govern network resources, including conditions and actions that are established by the network administrator with parameters that determine when the policies will be enforced in the network. In the case of ISP, policies are defined based on one hand the high-level business objectives of the ISP and on the other hand on the SLA (Service Level Agreement) agreed with its customers. The architecture is composed of a set of Policy Enforcement Point (PEP) components located generally in the network nodes, Policy Decision Point (PDP) components and a Policy Repository Component. In our case we have changed this architecture in order to take into account the nature of the wireless access network. In fact it is not sufficient to install PEP in the access routers and we propose to extend the PEP to the customer terminal. This means that policies can be installed also in the customer terminal.

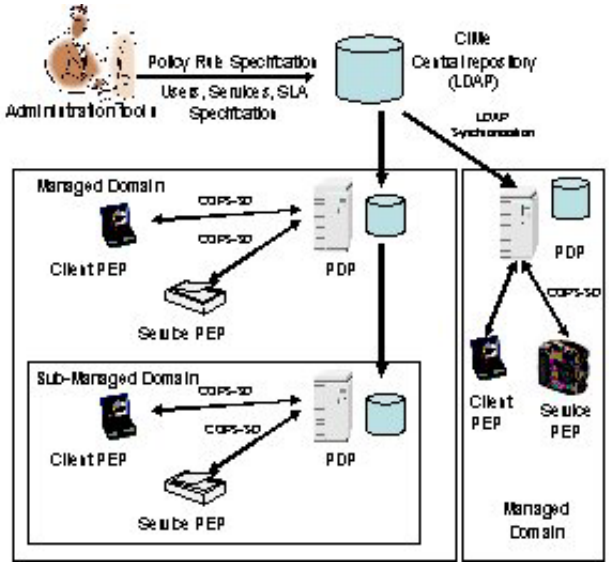

Figure 4-3 - Policy Based Management architecture

The architecture is composed of four main components, the administration station that allows the administrator to introduce the policy rules using administration tools, the Policy Repository that allow the persistency of the policy rules as well as the instantiation of the information model, the PDP that is the decision making component as described by the IETF PBM architecture and the various PEP that are installed directly in the mobile terminals (customer, printer, fax, etc.).

4.2.2.1 Administration tool

The administration station provides a set of tools that allows the administrator to define or to modify user and group profiles. Each profile defines the set of authorisations and refrains concerning the use of resources in a particular network domain (airport, railway station, etc). Each user profile will contain also a set of information that allows the authentication of the customer. The definition of profiles will facilitate the administrator to deploy a new strategy in the network and to satisfy a new customer. The concept of group permits to combine different profiles, thus the administrator can use the administration tools to add or to drop a customer from a particular group depending on his SLA (Service Level Agreement). To each profile, the administrator can add or remove a service usage, add or remove a geographical

location, etc in order to specify in details which services a group can use, from where the group can use them, at what period of time it can use them.

4.2.2.2 Policy repository

Once these profiles, groups, services and customer identified, the administrator has to save these information into the policy repository. This repository will be accessed by the PDP in order to make a decision regarding the service access rights. The technology that is used to implement this repository is LDAP [17]. It provides a shared space where different entities can store, search and update data in a fully distributed manner. This repository will not only contain the information concerning the profiles, groups and policies but also a number of information that permit to have an abstract view of the network and service environment. To define our own information model, we have used the CIM[1] model as described by the DMTF (Distributed Management Task Force). This information model is the reference model for representing network and service information in the IT world. It allows in one part to describe the static view of the environment (users, network and service description) and in another part to describe the dynamic view of the environment that governs the behaviour of the system through the definition of the policy rules.

4.2.2.3 The service negotiator: PDP

This particular PDP aims to manage all the elements that are under its responsibility. Mainly two type of PEP will be deployed a provider PEP that offer a set of services (i.e. service equipments) and a consumer PEP that consume these service (i.e. customers). Thus the role of this PDP is to take into account the registration request from newly introduced equipments (e.g. fax, printer, etc.) in the domain of its responsibility. Thus when a new client enters in the PDP domain, it has to identify itself. The PDP take this information and verify with the information that are contained in the repository. One the customer is identified, the PDP retrieves its profile that defines his rights and refrains concerning the existing services. The PDP sends to the PEP (i.e. the mobile terminal) the set of attributes that will allow him to interact with the available authorized services. If any new service is made available in the domain, the PDP is responsible for notifying all the concerned authorized PEP about the availability of this service. In the same time, if any changes occur in the business strategy of the operator or in the customer's contract, the PDP has to detect them (changes in the repository) and enforce them.

4.2.2.4 The service provider or consumer: PEP

The PEP is a process that is installed in each customer terminal and any equipment that is providing a service. It is also possible, in the case of equipments that do not support this client, to have an external adaptor but this aspect is not treated in this paper. Two type of PEP are defined: the client PEP and the service PEP. The client PEP is responsible for the discovery of service in a particular domain on the behalf of a customer while the service PEP role is to notify the available services in the case of a provider' equipment (printer, fax, copier, etc.)

Thus the role of the PEP in general is to: find the address of the local PDP; establish a connection with this PDP; authenticate itself to the PDP, perform notification or discovery of services in the particular domain using the defined COPS-SD protocol (introduced below).

Once the PEP is authenticated, the PDP identifies whether it is a consumer (client PEP) or a provider (service PEP). If the PEP is a consumer then the PDP will send him all the information about the available services for his particular profile otherwise it is the provider PEP that sends a registration request to the PDP informing him about the provided services as well as their attributes.

[1] CIM: Common Information Model

4.2.2.5 Communication protocols

The overall architecture is based on a number of protocols that support communications between its components. Between the administration station and the LDAP repository the protocol that is used is LDAP. The main services of this protocol used in this context are: search to browse the content of the repository, add, delete, modify and modifyDN to update the objects in the repository and finally bind, unbind and abandon to establish and release connexions with the repository. The other communication is between the PDP and the LDAP repository, similarly the PDP needs to browse the repository to identify the policy rules to trigger and update the repository with the information concerning the ubiquitous services that are available in its domain.

The third level of communication is between the client, or service, (PEP) and the PDP, this is the most interesting protocol. As stated before, this protocol has two main objectives: registering available services with the PDP, discovering and requesting service to the PDP. In order to achieve these goals, we have chosen to extend the COPS protocol [5]. The main reason to use COPS and not any other existing service discovery protocol is because COPS is well suited for signalling QoS requirements and we think that at some point the service discovery in ubiquitous environment will necessitate also QoS assurance. Thus it is very interesting to use the same protocol to discover services and request QoS. Another advantage of this protocol is that it was designed to transport policies decision thus it is well suitable if one uses PBM approach and it is the case. The fact that PEP is deployed directly in the customer terminal is an innovative proposal that can be seen at the first time as a constraint but if we look at the advances made in USB smart keys, we can think about the possibility to have it directly in the smart key and not in the customer terminal [6]. Finally, the COPS protocol can be associated with an information model such as CIM while it is not easily possible with other discovery protocols. This permits to have a strong information model to facilitate the representation and manipulation of the services information. A detailed description of this protocol will be presented in the next section.

4.3 COPS-SD: Operation

COPS-SD was defined in order to perform the communications between the PDP and the service PEP on one hand (for registration), and the communication between this same PDP and the Client PEP on the other hand (for authentication and service discovery). Therefore, it must allow fulfilling the functions relative to these objectives.

As this protocol is based on a COPS client, it must keep the basic pattern. Thus, a COPS-SD segment (message) has the same format as its ascendant [19], and is based on the TCP transport protocol to ensure reliability. This section will be devoted to the description of COPS-SD protocol and the details of each communication phase during the service advertisement, registration and discovery.

4.3.1 Connection establishment

At the beginning, a TCP connection is established between the PEP, and the service negotiator, PDP. Obviously, the PDP listens on a standardized and known TCP port: 3288. Once the connection is established, the PEP (Service or Client), sends a session-opening message, OPN, by specifying its identity (PEPID: ‘PEP Identification Object’) and the appropriate COPS client-type (Client-Type = COPS-SD). If no error occurs, and if this PDP supports the specified COPS client-type, it accepts the connection and it replies with a CAT (Client-Accept) message. Otherwise, it replies using a CC (Client-Close) message including the appropriate error code, ‘Error Object’.

4.3.2 Service registration

The Service PEP, after opening a session, sends a request, REQ, to the PDP. This registration message contains: a Context object specifying that the request is for registration, also, a description including all the attributes of the service. This description is conveyed in a 'Client Specific Information' object.

The PDP replies to this request by a decision message, DEC. Three situations are possible:

- No decision is applicable for this service, according to the defined access rules. So, the decision message will enclose 'decision flags' indicating a null command;
- There are decisions applicable for this service, according to the defined access rules. So, the decision message will enclose 'decision flags' indicating an install command, followed by one or several 'Named Decision Data' objects. These objects are used to describe the list of users which are authorized to access the service.
- The PDP remarks an error in the received request. The reply message will contain, in this case, an 'Error' object indicating the adequate code. So, The PEP must reformulate its request or finish its connection if the error persists.

4.3.3 User authentication and service discovery

The Client PEP, after opening a session, sends a request, REQ, to the PDP. This registration message contains: a Context object specifying that the request is for authentication, also, a list of attributes necessary for its identification / authentication (such as: login and password). Also, this description is conveyed by a 'Client Specific Information' object.

The PDP replies this request by a decision message, DEC. Three situations are possible:

- No service is available for this user, according to the defined access rules. So, the decision message will enclose 'decision flags' indicating a null command;
- There are authorized services for this user, according to the defined access rules. So, the decision message will enclose 'decision flags' indicating an install command, followed by one or several 'Named Decision Data' objects. These objects are used to describe the list of available and authorized services.

The PDP remarks an error in the received request, or the user is unknown (erroneous credential). The reply message will contain, in this case, an 'Error' object indicating the adequate code. So, The PEP must reformulate its request or finish its connection if the error persists.

4.3.4 Unsolicited decision

The PDP is able to send a decision message without receiving a request before. In this case, it should keep the previous 'handle' value. That occurs in one of the following situations:

A service leaves the managed domain, the PDP must send to each Client PEP using this service a new decision message. So, this message will enclose 'decision flags' indicating a remove command followed by a 'Named Decision Data' object. This object is used to identify the leaving service;

A service changes its configuration, the PDP must send to each Client PEP using this service a new decision message. So, this message will enclose 'decision flags' indicating an update command followed by a 'Named Decision Data' object. This will express the new service description;

A user leaves the managed domain, the PDP must send to each Service PEP authorizing this user a new decision message. So, this message will enclose 'decision flags' indicating a remove command followed by a 'Named Decision Data' object. This last targets the leaving user.

4.3.5 Synchronization

When a policy rule is added, updated or removed; the PDP should contacts the concerned PEPs (Service / Client) by sending a 'Synchronize State Request' message, SSQ. Behind, each reached PEP, should emit, again, an appropriate request (Registration / Authentication). This synchronization period ends by sending a 'Synchronise State Complete' message, SSC, coming from the PEP.

Note: The Report messages, RPT, are used by the PEP to indicate the well reception and understanding of the decision messages received from the PDP. However, if this report indicates a failure, the PDP must send, again, its decisions. Also, if this situation persists, then the connection should be closed.

4.3.6 Connection Closing

Before leaving the domain or disconnecting itself from the network, the PEP (Service / Client) should notify its associated PDP by sending a 'Client-Close' message, CC. The session close can, also, occur when a persistent communication error appears.

An example of these communication steps between the various components of the system is provided by the sequence diagram in figure 4.4. This example follows a scenario of exchanges between the PDP and the different PEPs. Concerning user's rights: 'C1' has a high profile while 'C2' has restricted rights. At the end of the sequence, we can remark a change in strategy policies: C1's rights become limited.

4.3.7 Protocol Information Format

The information exchanged between the PDP and the different PEPs is represented by a named data structure, also known as a policy information base (PIB) [21]. This PIB can be described as a set of classes. In our system, these classes are directly mapped to the CIM-based information model.

Concerning the exchange of service descriptions, the relative transmitted objects are obtained by direct usage of LDAP entries describing these services. This process ensures the independency between the communication protocol and the service information model. Thus, this model can be changed, updated or extended without any necessary modification on the COPS-SD protocol. The interpretation of these exchanged data is left to the applications over the discovery protocol.

4.3.8 Reliability and Security

The break-down of a PDP during negotiation is possible and may cause problems and data incoherence. The base COPS protocol provides various mechanisms for reliability and fault tolerance [19] such as: 'Keep-Alive' messages, secondary PDP, synchronisation and report messages. These can be used and combined in COPS-SD to make it more reliable.

Another important issue is the trust relationship between the components involved in the discovery process. Thus, the base COPS protocol provides authentication, message integrity and replay prevention by means of 'Integrity' object. Also, IPsec (IP Security Protocol) or TLS (Transport Layer Security) can be used to ensure connection security.

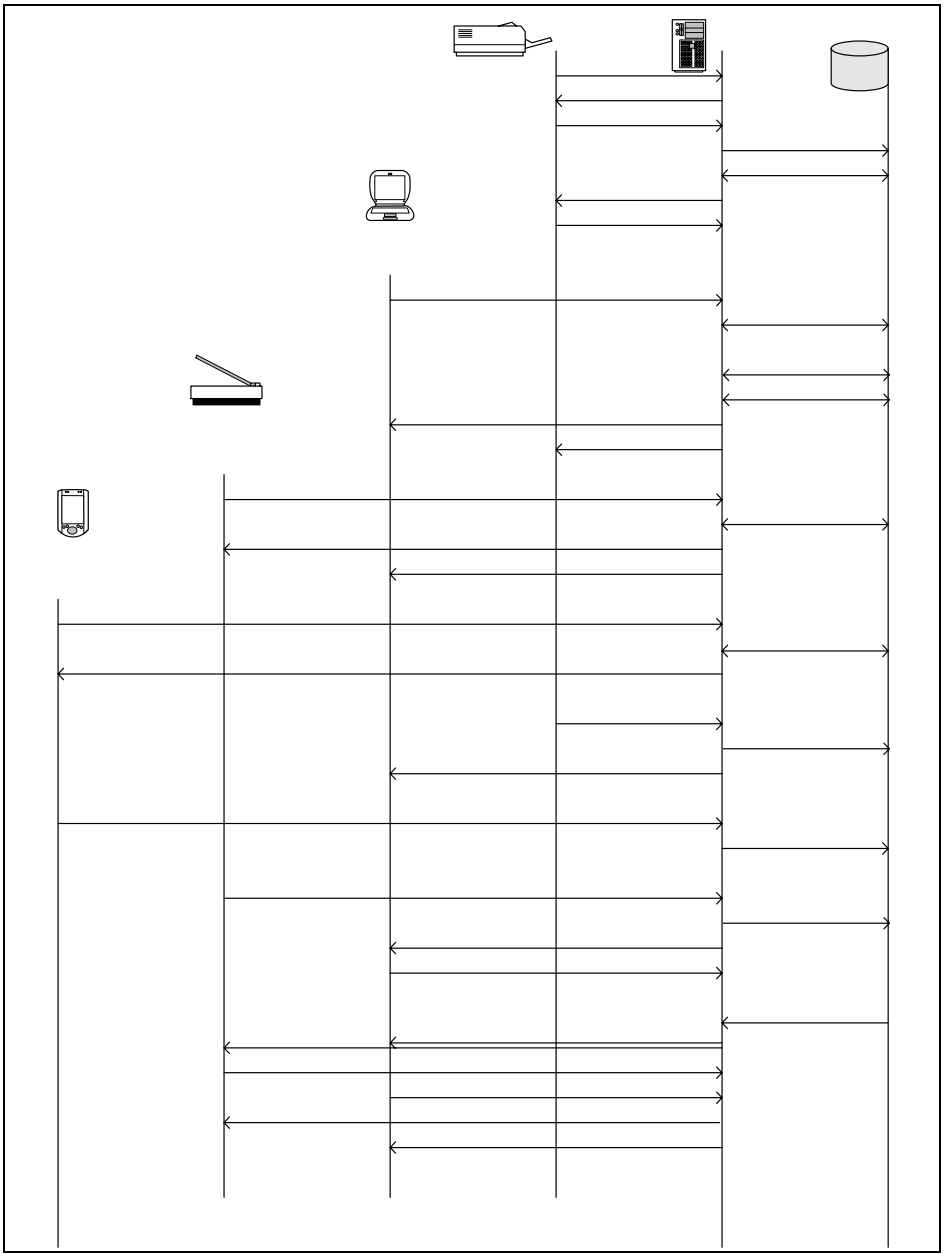

Figure 4-4 - Service publishing and discovery sequence diagram

5 Implementation

In order to implement this system, we chose the JAVA programming platform. This allows portable applications: independent from the operating system. In this section, we will just approach some comments concerning the implementation of our system's components.

The administration tool (Figure 5.1) was implemented as a java application providing a convivial interface to the administrator. This tool facilitates the introduction of policy rules and the supervision of the global access strategy. Also, the operator can introduce manually the services which are not PEP-enabled. In addition to user accounts, groups, etc.

The PEP was implemented as a communicating process. It is based on COPS-SD for service negotiation and registration. In the case of a client PEP, this offers an interface to the user. So, he can connect to the system using his ` login' and `password'. Hence, this interface will be used to display the description of all the available and authorized services. Thus, he will be able to choose the desired service.

Client PEP

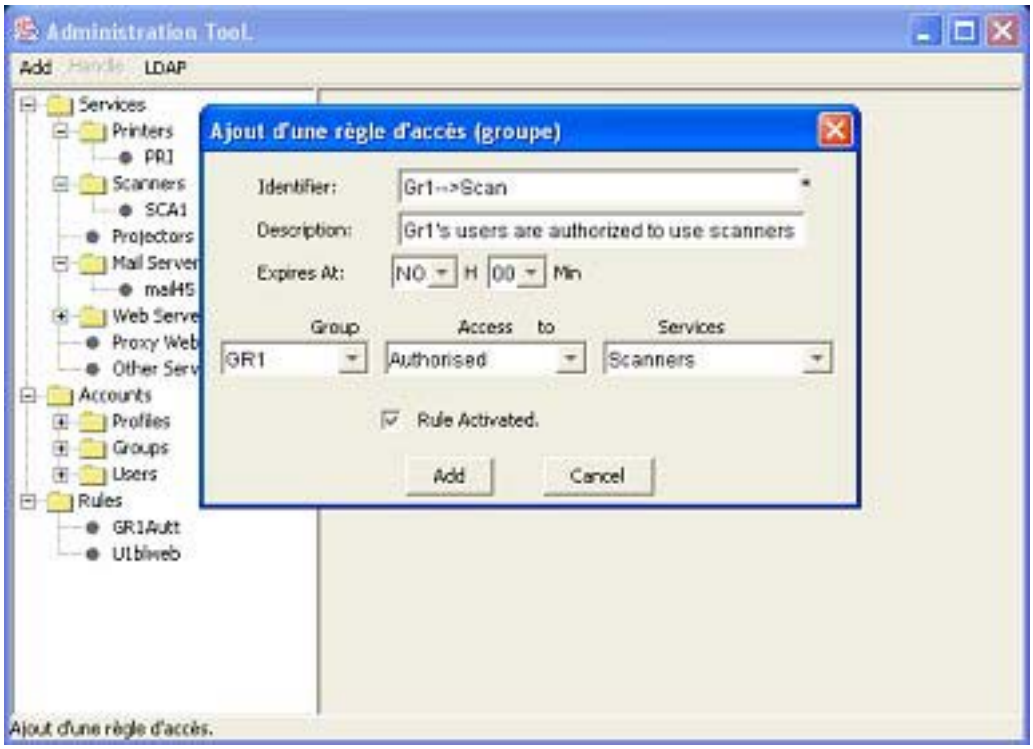

Figure 5-1 Administration Tool

Concerning the service PEP, and for testing reasons, we implemented an application emulating the behaviour of a PEP-enabled printer. So, this entity can send service registration requests by using the COPS-SD protocol. Furthermore, the PDP was realized as a multi thread application handling the communication with all existing PEPs.

The communication protocol, COPS-SD, is based on the java sockets middleware, while, the policy repository was implemented using an *OpenLDAP* [22] directory.

All these software components have been integrated in a unique system and deployed in a wireless LAN environment where different tests have been performed to validate the various concepts as introduced in this work.

6 Conclusion and Future Work

Embedded into the concepts of M-Zones and smart space management, this paper has and related developments have presented a way for the integration of management and smart space concepts with a hierarchical Policy-based Management approach. Starting with the introduction of the concept of managed zones, M-Zones, followed by introduction of Policy Based Management, the paper has defined an integrated architecture and described the components of this very architecture. Furthermore, the paper has shown how to apply the Policy-based Management approach for realising service discovery and access, including the necessary extension of the COPS protocol, which resulted in COPS-SD. The challenging aspect of this work has been on the one hand the diversity of Smart Spaces and on the other hand the fact that the complexity of their management requires a simple but flexible solution.

The implemented system is directly derived from the integrated architecture to show that the architecture is a solution that is light-weight, open, smart and service generic:

- Lightweight as the solution can be widely adopted by vendors and providers of different size and market penetration. This means the solution take into account

commonly accepted principles already adopted by service providers and network operators.

- Open as the solution includes well-defined interfaces. Interoperability with legacy systems is a key issue for a smooth integration. This thesis reflects the term open also to indicate that the market demands for an easy adaptation of new technology and interworking with other systems.
- Smart as the solutions must reflect the intrinsically dynamic aspects (particularly for the discovery of services and resources), enabling a flexible adaptation to customer requirements and operator/provider needs. This will is supported through the use of meta-data repositories throughout the whole system life-time to provide a comprehensive knowledge base improving multi-domain service provisioning and also in the operations/maintenance phase.
- Service generic as the solution is independent of the actual services that are offered. Nobody can predict if there is a killer-application for future ser-vices and which one this might be. The preferred applications, services and resources will surely differ significantly within different countries and different cultural groups.

This paper describes a new approach for the support of smart space management. However, some issues have been considered to be out of the scope for this approach. These issues are security, testing, context-awareness and adaptability of policies. All of them are important for the specification and operation of Smart Spaces. These aspects are currently under investigation in terms of the following future work:

- Distributed M-Zones testbed – Installations of three different testbeds at WIT/TSSG in Ireland and at LIP6 and University of Evry in France will provide the basis for systematic tests of the implemented system in a large scale, inter Smart Space management scenario. The deployment of these testbeds will be finished this summer to start the tests and simulations early autumn this year.
- The dynamic adaptation of security policies in pervasive environments, with contextual information as the catalyst - This dynamic approach will assure users that mechanisms to deal with the new exposures and vulnerabilities of pervasive computing are in place. However, there are numerous challenges that most be handled, including scalability issues and difficulties with managing conflicting policies. One of the most notable questions is - is it feasible to use an object you want to protect as a means of protecting? This unprecedented issue of using implicit information, which may be of a personal nature or contain sensitive intellectual property (i.e. context information), as the catalyst for developing accurate and timely security policies must be examined.
- Semantic Service Discovery and Task Driven Service Composition for Smart Space Environments - Research in the area of service composition is based on semantic knowledge being provided by the service itself. This research has been recently focused on Web Service technology and the discovery and composition of web services based on semantic knowledge the service provides. The main concept is to use machine and human understandable semantics to describe the functionality of a service. This is known as the Semantic Web paradigm. This approach looks like an appropriate solution for smart space, on demand, service discovery and composition.
- An extensible standards-based Information/Data model for a Smart Space Personal INformation System - This work package proposes to exploit standard information models from the ITC domain in defining an Information Model (IM) for a system that manages personal information in a smart environment. Once a common information model has been defined, data models containing more specific implementation level details can then be derived. This will be shown through the implementation of a personal information system based on this model. This system will then be evaluated

for conformance to the relevant standards through the use of conformance specifications.

The basic result of this work is the conclusion that an approach that integrates basic concepts of M-zones and Policy-based Management provides significant benefits for the management of Smart Spaces. The unification of use, operation and control with the tasks of maintenance and administration minimises the effort that has to be spent for the mid- and long term operation of a distributed application. The independence of concrete middleware and management technologies improves the portability of applications. An application designer and programmer can concentrate on its actual task - the realisation of profitable applications instead of dealing with constantly changing technological issues. The simplicity of the presented approach allowed for a simple and lightweight implementation that can be employed in many different environments, starting from small devices up to complex and huge service platforms.

6.1.1 Acknowledgements

The authors of this paper like to thank the M-Zones team at TSSG for their valuable input, namely Keara Barrett and Ray Carroll (Mobile and Cellular IP, Personal Information System, Security Policies) and Alan Davy (UPnP testbed and Semantic Driven Service Discovery). Furthermore we like to express gratitude to the M-Zones teams in CIT and TCD and to the various researchers at LIP6 and University Evry.

7 Bibliography

[1] Declan O'Sullivan, Sven van der Meer, David Lewis: *Straw-man Glossary*. M-Zones WP1 Working Document, The M-Zones programme, March 1, 2003.

[2] Sven van der Meer, Robert O'Connor, Alan Davy: *Ubiquitous Smart Space Management*. 1st M-Zones Workshop, Waterford, Ireland, December 11, 2003 available at www.m-zones.org

[3] Declan O'Sullivan, Ruaidhrí Power: *Bridging heterogeneous, autonomous, dynamic knowledge at runtime*. 1st M-Zones Workshop, Waterford, Ireland, December 11, 2003 available at www.m-zones.org

[4] Declan O'Sullivan, Dave Lewis: *Semantically Driven Service Interoperability for Pervasive Computing*. In proc. of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, USA, 19 September 2003, ACM Press.

[5] Robert O'Connor, Sven van der Meer: *Present and Future Organisational Models For Wireless Networks*. Workshop on Adaptive Systems for Ubiquitous Computing at the International Symposium on Information and Communication Technologies, ISICT'2003 Dublin, Ireland, September 24-26, 2003.

[6] Keara Barrett, Ray Carroll, Sven van der Meer: *Investigating the Applicability of Mobile IP and Cellular IP for Roaming in Smart Environments*. Proc. of 2nd IEE/IEEE Telecommunications Systems Research Symposium, ITSRS 2003 Dublin, Ireland, May 6, 2003.

[7] Alan Davy: *Components of a smart device and smart device interactions*. M-Zones White Paper, M-Zones, September 2003.

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

[9] Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M: *Adaptive and Dynamic Service Composition in eFlow*. HPL-2000-39 20000406 External, March 2000.
[10] Chakraborty, D., Joshi, A.: 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, December 19, 2001.
[11] Steinfeld, E.: *Devices that play together, work together*. EDN Magazine, September, 2001
[12] Chitrarasu, M., Joseph, K., Rao, M.: *Jini by Example – Whitepaper*. published online.
[13] A. Friday, N. Davies, E. Catterall: *Supporting service discovery, querying and interaction in ubiquitous computing environments*. 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access, Santa Barbara, CA USA, May 20, 2001
[14] M-Zones Deliverable No. 1: *State of Art Surveys*. The M-Zones programme, Release, May 2003 available at www.m-zones.org
[15] D.C. Verma: *Policy-Based Networking – Architecture and Algorithms*. New Riders, Nov 2000.
[16] A. Westerinen et. Al.: *Terminology for Policy-Based Management*. RFC 3198, Nov 2001.
[17] J. Hodges, R. Morgan: *Lightweight Directory Access Protocol (v3): Technical Specification*. RFC 3377, Sep 2002.
[18] DMTF Policy Work Group: *Common Information Model*
[19] *(CIM) Specification version 2.8*. Aug 2003.
[20] J. Boyle, R. Cohen, D. Durham: *The COPS (Common Open Policy Service) Protocol*. RFC 2748, Jan 2000.
[21] UCOPIA company: www.ucopia.com.
[22] K. Chan & al: *COPS usage for Policy Provisioning (COPS-PR)*. RFC 3084, Mar 2001.
[23] OpenLDAP [Online], available: www.openldap.org.