A Smart Space Management Framework

Declan O'Sullivan,
declan.osullivan@cs.tcd.ie
Vincent Wade,
vincent.wade@cs.tcd.ie

Knowledge and Data Engineering Group,
Computer Science Department,
Trinity College Dublin,
Ireland.
http://www.cs.tcd.ie/research_groups/kdeg/

Abstract

Very little attention has been paid in smart space or ubiquitous computing research to the analysis of what is necessary for the management of applications in a diverse and multiple operator environment. By analyzing the ongoing smart space research and emergent (telecom, internet, enterprise) management system approaches, it has been possible to identify management requirements that need to be addressed.

One of the critical challenges that must be faced is how to make the dynamic composition and adaptation of management services possible for a non-technical user (e.g. a home owner). This technical report proposes a Smart Space Management Framework encompassing aspects of logical architecture, development methodology, technology architecture and reusable elements, to enable this dynamic management service composition. Use of the framework is illustrated with an education smart space scenario.

Table of Contents

1.0 Introduction

It has been nearly 10 years since researchers at Xerox PARC introduced the vision of ubiquitous computing in order "to reposition computing into the environmental background, to concentrate on *human-to-human* interfaces and less on *human-to-computer* ones" [Weiser99]. This vision has proven a catalyst that has initiated a large amount of research activity across the world under many different guises, with "invisible computing" [WashU00], "pervasive computing" [Ark99], "ambient intelligence" [Ducatel01], and "smart spaces" being the most popular. "Smart Spaces" [Rosenthal00] are environments with traditional computing hardware as well as embedded computers, information appliances, and multi-modal sensors allowing people to perform tasks efficiently by offering unprecedented levels of access to information and assistance from computers[1].

Research to date has primarily focused on the issues surrounding the development of smart space applications (e.g. Georgia Tech's SmartHome, Accentures's Smart Gym, University of Tampere's Smart Living Room). Such spaces have significantly simpler management requirements than the general case of interest, since it is possible to constrain the nature and behavior of the devices in the space very closely. The general case – where the population of devices in a space is highly dynamic and largely unconstrained – opens up major research challenges. In addition, smart spaces are likely to be owned or operated by different types of entities (e.g. an individual for the home, a business for an office building or local authority for a public area). Management systems will be needed to support users roaming from space to space, and as yet this issue has received very little research attention.

Smart space applications will have significant difficulties moving beyond the laboratory into the mainstream unless the management requirements are adequately addressed. This mirrors the experience of the telecommunications community where the speed of introduction of new networking technologies and services is tied to the availability of operational, administration and maintenance management systems.

Section 2 identifies management requirements that need to be addressed by analysing current smart space research and analysing emergent (telecom, enterprise and internet) management system approaches. One of the key requirements identified is the ability by a non-technical person to operate a smart space, which in turn requires that the smart space management system can be rapidly and dynamically composed and altered. To meet this requirement, Section 3 proposes a model-driven Smart Space Management Framework encompassing aspects of logical architecture, development methodology, technology architecture and reusable elements. Section 4 presents initial conclusions and future work.

1 Take for example, a cardiac surgeon who is woken by his alarm clock. Upon sensing that he has got up and is having a shower, the home smart space sends messages to the coffee machine, toaster and fruit press appliances to have breakfast ready. His PDA is activated and downloads any emergency case patient files that may have arrived overnight, and alters the day's appointment and work schedule if necessary. Upon leaving the house, the home management system configures the intruder alarm, the call forwarding/answering service and utility management services accordingly.

2.0 Smart Space Requirements

2.1 Analysis of Smart Space research

Operational and management requirements that need to be met in order to make Smart Spaces a reality can be identified from the research. These requirements can be considered under a number of headings: how to empower participants, appliances, spaces, programmers and operators. The challenges surrounding empowering participants, programmers and operators are probably the most difficult. Ten years ago the use of the web by non-technical users was rare. Now however, ordinary people are able to use the web as a consumer, web page author or web site administrator. In a similar way, for the smart space vision to become a reality, the ordinary user needs to be empowered to play their choice of roles of participant, operator or programmer of a smart space.

2.1.1 Empower Participants

Smart Spaces need to provide natural user interfaces and interaction with an assisted and invisible approach [Abowd98]. The goal is to transform the interaction from *human-computer* towards computer facilitated *human-information* interaction and facilitated *human-human* cooperation. This involves providing more natural interfaces for people to interact with the smart space. Smart space research in the area encompasses a wide range of topics: wearable computers [Minar99], speech recognition, visual recognition, audio/visual recognition combined [Neti01], and motion and pressure sensing devices [Fishkin98]. It requires that the appliances (which vary in size and appearance, e.g. PDA to wall size display) can be used in parallel by users as individuals as well as in groups [Streitz01].

From the perspective of the management system, this drive towards natural interaction leads to many different kinds of devices with different characteristics (e.g. jacket, wall display), a diversity of connectivity and network traffic patterns, and a dynamism on a scale not seen to date. For this reason it is envisaged that devices will be designed to undertake a high degree of self management, but will need to cater for external management stimuli as well. There will also be a large degree of management through Peer to Peer interactions. Overall management within the space will involve management of devices and smart space infrastructure taking the overall context into account. Thus in contrast to emergent[2] management systems, the management intelligence will be split over a number of levels.

Today computers interact with users in terms of low level abstractions – applications and individual appliances. A smart space needs to take over most of low level management of application and appliances so that user interacts in terms of tasks (that represent user intent) [Wang00]. It also requires the smart space to fuse data available from many sources (e.g. location sensors, online databases, user/task models etc.) so that it can infer

[2] Emergent meaning current telecom, enterprise and internet management

some of the user intent rather than relying on user commands [Portolano]. Furthermore a smart space should offer proactive guidance, so that when a task cannot execute, the system offers suggestions to the user to help complete the task in some other way [Abowd98]. Users need to be able to transition between spaces seamlessly [Dobson99]. Finally, methods need to be devised so that the user can cope with the flood of information and computing power that now becomes available in undertaking a task [Abowd98] (analogous to the early days of the WWW before search engines and portals).

Smart Space management systems will be required to support participants' interactions with the smart space in terms of tasks. In particular the modeling required in mapping between tasks and service components could be complex given that several alternative service compositions may be possible to execute a specific task.

The service components that will be required for a task execution may be provided by software incorporated into the management systems themselves. For example, when a participant wants to engage in a task to alter their contact details they will not concern themselves that the task is actually achieved via profile management service components on the management system. From the participant's perspective, it is a task like any other.

In addition, the management system needs to support the management of tasks by the participants themselves at runtime. These task management service components need to appear to the participant as tasks in their own right. For example, faults at the system level need to be transformed into "task interruption issues" that are raised with the participant using the terminology of the task involved. Subsequently the coping strategy suggested by the participant needs to be transformed into control actions undertaken by management service components.

2.1.2 Empower Appliances

Typically "appliances" are considered things that are empowered for the benefit of people. Think of any appliance within a kitchen, they are designed to bring benefits to people for cooking, cleaning etc. In the same way, appliances are required for smart spaces to bring benefits to people in the information gathering, sharing, and cooperation areas. This requires some new thinking as to what kind of design approaches need to be applied to appliance design. The challenge is to make the user interface into a smart space an inherent part of the tools that the user would naturally use within a task environment. Hence the early work on smart white boards. Cheap accelerometers, sensors and advances in material sciences (e.g. changing colour of an appliance depending on status) hold out the promise for exciting new ways of a user being engaged with a smart space through the normal appliance/tools that they already use for a task [Portolano]. For example to restart a smart post-it pad, you might just give it a shake! As well as the requirement for new design approaches for designing appliances for a smart space, there is also the requirement to improve the energy management and memory management of computing devices that will be embodied in appliances [Wang00].

2.1.3 Empower Spaces

Very little consensus as yet has emerged in the state of the art regarding the architectural approach most suited for the smart space environment's computing infrastructure. This reflects the relative recent focus on the area by researchers and also the rush to apply existing approaches to this new area. For example, several research projects are looking at the applicability of decoupled integration approaches (Stanford's iRoom using IBM Tspace [Fox00], NIST's Airjava project using Sun Jini [Mills00]), several are looking at the applicability of Agent approaches (MIT's Metaglue project [Cohen99] and MIT's Hive project [Bradley99]) and several at the applicability of mediation approaches (Berkeley's Ninja project [Hong01], Stanford's mediation project [Kiciman00], Inria's Smart Office project [LeGal99]).

There is consensus however on the need for the environment to gather information about a user's activity within the smart space. Traditionally the presence of a person has been detected through the use of sensor technology and active badges [Gibb92]. More recently advances in computer vision [Moeslund01] are paving the way for this location information to be gathered through video cameras. In addition this kind of vision technology opens the possibility of not only knowing where the user is but also interpreting what actions they are performing (through gesture recognition etc.). Combining the information about where the user is, what action the user is performing, with a model of tasks that the user can engage in within that smart space, leads to the generation of "context" information about the user. In his paper, Wang provides an excellent overview of how the smart space can use this context information to trigger behaviour to support the tasks of the user [Wang00].

Making this context information available for use by various smart space applications is also a subject of intense study. No consensus on the appropriate style of application programmer interface has emerged. For example, a user interface widget style [Anind99], an architecture with "artifacts", "monitors" and "relationships"[Pascoe98], and a declarative querying style [Schmidt99]. A significant difference (from a management requirements perspective) between context information management and that of location/profile management within emergent management approaches, is the extent to which the user must have control over the context information and who is allowed access to it. This becomes even more difficult when this context management must be presented to the user in a non-technical, natural way. Another key difference relates to the catering for the dynamism of the context information and ubiquity of availability that will be expected. The frequent and rapid changes to the context information coupled with the heterogeneity and timeliness of access required, pose greater challenges than have been faced by emergent systems to date.

Due to the heterogeneous nature of the user population, the smart space will also be required to support not only ad hoc networking (to cope with the arrival of mobile appliances into the space), but also support "ad hoc servicing" to allow for the dynamic discovery and installation of application services or dynamic adaptation and composition of application services from components [Kiciman00, Hong01]. Smart space management systems will be required to support this dynamic service adaptation and composition, both at design time and runtime. Emergent management systems have not had to support such dynamic service adaptation and composition in the past, as typically they have been focused on single service infrastructure architectures designed with limited reuse capabilities (e.g. Intelligent Networks).

The above research work into environment's computing infrastructure issues raise challenges for a management of a smart space system at a number of levels: the infrastructure itself, the smart space applications which run on the infrastructure and the information about users and tasks. All of these need to be configured, managed securely and usage monitored. The question is what is delegated to infrastructure components in terms of self management and what still needs to be managed by a management system?

2.1.4 Empower Programmers

Users will "programme" the smart space by expressing requirements and expectations of a smart space in terms of *tasks* that the user wants supported and desired *strategies* (as there may be alternatives outlined) for sequencing and executing these tasks. The main requirement here is to allow the task modeling that is necessary, to be possible in user terms and be environment independent. This is analogous to the current attempt in the Software Engineering world to empower business users to model application systems [McGinnes01] and their process flow (e.g. the TeleManagement Forum NGOSS initiative [Ngoss01]) in their own terms. The User Task Language to execution components mapping needs to be available and undertaken at runtime. Furthermore this leaves open the possibility for the execution thread to consist of a dynamic composition of service components and management service components, depending on what is required to fulfil the task.

Thus with the users seen as the main "programmers" of the smart space, the role of smart space "developers" is to design infrastructure hosted components that can: be used within many different threads of user task execution; can support a range of input/output devices; can discover/use other components and resources within the smart space; and can discover/use user context information. In designing their "one.world" architecture, [Grimm01] suggest three principles to guide designers of smart space systems that would empower developers to "programme for change". First, systems should "expose change", including failures, rather than hide distribution so that developers can implement their own strategies for handling change. The leasing mechanism seen in Jini is an example of such a facility. Secondly, systems should make it easy to extend and compose applications and services at runtime. Thirdly, systems need to provide to provide a clean separation between data and functionality so that they can managed separately and can evolve independently.

Management support can clearly be provided to the user in the expression of smart space programmes at design time. One such tool for example, would support the generation of "Task Level Agreements" to express the quality contract expected between the smart space and the user. These agreements would use task related terms and vocabulary, and thus a mapping and negotiation tool is needed to connect "Task Level Agreements" with task level specifications, service composition specifications and quality of service parameters of the smart space.

Empowering users to be the programmers of a smart space, places significant new requirements upon the management systems, than those that have been faced by emergent management systems to date. First, management functionality in support of task programming (e.g. task level agreement negotiation tools) needs to be defined. Second, functionality of the management systems need to be represented as management tasks themselves (in user terms) so that they can be included in programmes.

2.1.5 Empower Operators

An operator requires facilities to be able to actively manage the smart space environment. Substantial resources have been allocated to the construction of prototype applications and laboratories in order to demonstrate the smart space vision, test the emerging solutions and study the human and social issues. Research to date has primarily focused on the issues surrounding the development of smart space applications. Thus we find research into applications for an "Intelligent Room" in the MIT Oxygen project, Stanford's iRoom project and GMD-IPSI iLand project (to name but a few). Typical of such applications are interactive conference tables, interactive wall displays and rememberance/recording agents. Focusing on the smart home environment for example includes projects such as: GeorgiaTech AwareHome, KTH comHome, Microsoft Easyliving and Frauhofer's inHaus. Here we find applications such as mood music and light controls, email browsing on flat surface etc. There has even been study by Accenture into what applications would be needed in a smart public gym.

Such spaces have significantly simpler management requirements than the general case in which we are interested, since it is possible to constrain the nature and behavior of the devices and users in the space very closely. Recently some researchers have started to address the requirements of more complex smart spaces. For example, Dobson et al highlights that little research undertaken into management of multiple users and applications within a smart space, and that traditional programming models are ill equipped to handle due to the multiple interface modalities and unknown interaction patterns [Dobson99]. They then go onto outline a possible configuration management architecture that would allow applications migrate between spaces using event, relationship and policy services.

A more complete set of requirements can be identified by undertaking a traditional FCAPS analysis of a smart space. For example, it is clear that efficient problem recognition, resolution and appropriate representation to the user is needed. Models are needed of smart spaces to enable setup and dynamic reconfiguration of resources. Resource usage needs to be accounted for, both for operational and perhaps charging purposes. Smart space operators will need to enable dynamic negotiation of task level agreements with the user (or their agent) and need facilities to monitor and control the use of resources. Key to the success of smart space operation will be the management of security and privacy of smart space occupants. Given that there needs to be a move away from single user/single device/single physical space situation, a whole new way of looking at security and privacy management is needed. Also operators will need to manage the lifecycle of the resources (components and appliances) within the smart space, how they are introduced, suspended and withdrawn. Given the massively dynamic nature of a smart space, operators will require flexibility in applying management policies such that they are driven by what users are in the space at the time, what resources are in use and operational goals and objectives. A logical consequence of empowering participants to be the "programmers" of the smart space, will mean that operators will need to allow participants to undertake "my space" management (e.g. privacy management, task models configuration etc.).

What is clear from the range of smart spaces that are being explored, is that there will be more types and instances of operators of smart spaces than are involved today in service delivery. The operator types are: Private (e.g. home user); Group (e.g. a company) and Public (e.g. a local authority). Most challenging of these from a management requirements perspective is that of the Private operator. Every home occupier will want control over their own smart space. These private operators are likely to be non-technical and will require that the management systems can easily be composed and evolved

As smart spaces will be interconnected, a major challenge is to ensure freedom of movement of participants. This requires enabling the seamless transfer of their services and their devices between smart spaces. As this kind of issue has been addressed in considerable detail by the telecoms management community, management requirements in support of this interoperability can easily be found. The key difference with interoperability in the smart spaces area, is that there is a critical need to facilitate the dynamic negotiation of service (usage and quality terms) and participant migration at runtime. This is because separate smart space domains are expected to be of varying granularity (e.g. room, corridor, floor, building, campus) and participants are assumed to roam quite frequently but will not have roaming agreements set up with all possible operators.

2.2 Analysis of Emergent Management System Approaches

The study of techniques and paradigms for management systems has increased over the last two decades in line with the increasing complexity and deployment of distributed services and systems. The question that naturally arises is whether these techniques and paradigms are sufficient to provide the basis of management solutions for smart spaces? At first glance it could be said that this is obviously the case, given that it can be argued that a smart space is just a massively distributed hardware and software system, albeit with implications upon the technologies (in terms of scalability, reliability, availability). However as illustrated in Section 2.1, a deeper analysis of smart spaces points towards the need for new management paradigms and techniques. For example to support the "programming" of a smart space using user defined tasks which results in the runtime adaptation of service components and dynamic service composition.

The following tables were produced to further the requirements analysis. Table 1 decomposes a smart space from a management perspective into main concerns:

- **Devices/transmission**: a smart space is composed (statically or dynamically) of an interconnected network of devices (sensors, servers etc.). Interconnection is achieved through wireless and fixed transmission media.
- **Computing**: a smart space will have a software infrastructure that provides the execution environment for smart space services.

Table 2 decomposes a smart space from a management perspective into main concerns:

- **Information**: central to a smart space will be its ability to gather, structure and provide information to smart space services and smart space management services.
- **Service components**: These are the software entities which are deployed and managed by a smart space operator and which provides the operational behavior of a smart space.
- **Tasks**: Users will programme a smart space by thinking in terms of tasks and sequencing of tasks in terms of strategies. Creation and management of these tasks and strategies by the users must be supported by the management system.

For each area of concern, each table identifies the main similarity it has with emergent management approaches and illustrates with some examples. A starting point for discussion (in the form of questions for smart space management researchers) is then proposed under each of the FCAPS[3] management and interoperator management headings. Interoperator management is given equal status with FCAPS management, because it will be of fundamental concern to an operator of a smart space (as user roaming will be part of normal operation rather than an exceptional event).

[3] That is, examining the management requirements under the headings **F**ault Management; **C**onfiguration Management; **A**ccounting Management; **P**erformance Management and **S**ecurity Management.

Although it is not an exhaustive analysis, it is clear from the questions that are posed that it would be incorrect to treat smart spaces as just another distributed application. Each smart space concern is worthy of study from a management perspective in order to determine if existing paradigms and techniques used for an analogous purpose are potentially applicable, and if not what paradigm/technique research needs to be undertaken in order to address the management need.

	Management of Devices and Transmission	**Management of Computing**
Main Similarity	Management of heterogeneous devices and interconnection.	Management of computing, coping with a wide range of different interaction paradigms: Manager Agent, Event Based, RPC, Message Based, Signaling
Examples Telecoms/Enterprise	Heterogeneous routers, cables, fixed and wireless Networks, Intelligent Networks	CMIS, CMIP, SNMP, XML
Main questions for Smart Space Management Research overall	How to instrument sensors, smart appliances, ad hoc wireless networks, computing platforms, in for an integrated management framework?	Interaction paradigm: what can be reused from state of the art, what new lightweight approaches needed?
Main questions for Fault Management	What events and thresholds can be expected from diverse range of equipment?	Given dynamic composition of a Smart Space at any one time (including visiting resources), how can correlation of fault events be done efficiently?
Main questions for Configuration Management	How much configuration can be done externally and how should be self configuration, based on environment the device finds itself in?	Can a single framework cope with a mixture of self configuring and externally configurable resources?
Main questions for Accounting Management	What usage events to record? For example, in Smart Spaces will want to record resource usage attempts that fail due to resource being in use by another user?	Are the emerging flexible rating approaches sufficient for the diversity of events likely in a smart space environment?
Main questions for Performance Management	What are the smart space device and ad hoc network performance indicators that are needed?	Is an overall quality of service framework for a Smart Space possible that takes into account that the space is composed of fixed elements and dynamically introduced elements?
Main questions for Security Management	Trust and authentication are key issues for users and operators alike. With such a diverse range of devices and interconnections, how can this be achieved in a non intrusive way?	Can a lightweight security framework be introduced to allow freedom of movement between smart spaces?
Main questions for Interoperator Management	How will a Smart Space capabilities be discovered/interrogated?	How will the handover of user session information be achieved between smart spaces?

Table 1. Smart Space Management & Telecoms/Enterprise Management analysis

	Management of Information	Management of Service Components (by operators)	Management of Tasks (by users)
Main Similarity	Making information available about what current status and configuration is of users and their environment	Management of service components that provide the service functionality to users and which run on the interconnected devices and controlling the environment based on strategies as to what can be done when	User needs control over own tasks which are supported by the service components
Examples Telecoms/Enterprise	Shared Information Modeling (e.g. DMTF Common Information Model, TMF Shared Data Model)	Virtual Private Network service management	Intelligent Network user profile management
Main questions for Smart Space Management Research overall	How will user context information be represented and made available?	How will service components be represented so that they can be composed, adapted and execution policies enforced? How will tasks to service components mapping be achieved? Will a management service component be just another type of service component?	How can the user express management requirements in their task language?
Main questions for Fault Management	Are predictive models (based on fault histories of a space) possible, such that a user has an indication of potential problems which may affect task completion before and during task execution?	How will consequences of faults (in service components) upon user tasks be deduced?	How will faults be represented to the user in task terms, given the distance of user from technical details?
Main questions for Configuration Management	How will configuration metadata for devices, transmission, computing, applications, services & tasks be represented?	How can reconfiguration of Smart Space service components be achieved based on resources becoming available/unavailable on a frequent basis?	How are tasks and strategies configured by the user?
Main questions for Accounting Management	What strategies are needed to cope with the lifecycle of event recording, given the volume of events that are going to be generated?	Can the service component usage data be mined to support strategies for reconfiguration/reallocation of resources?	What is the users' model of resource usage as it relates to charges? Pay per resource usage? Charged per length of stay in a space?
Main questions for Performance Management	Given the complexity of the composition of a smart space, and the need to proactively manage expectation of the user, significant amounts of performance information needs to be gathered and processed from diverse sources. How can this be done in a lightweight manner?	How will service components be instrumented?	How do we design the equivalent of SLSs and SLAs at a Task level?
Main questions for Security Management	How to represent security policies and user information, and legal service compositions?	Given that service components can be composed in many different ways, how can security be specified?	How do we provided easy management of privacy profiles for the user? Concepts such as "friends and family" needed?
Main questions for Interoperator Management	What information is appropriate to share or exchange? Certainly user/task session information but what of resource usage? Performance usage? User task history?	If a user migrating across Smart Spaces has processing underway in the former Smart Space, will this be allowed to continue or will the service component processing underway need to be migrated too?	As migration between smart spaces cannot always be seamless for a user (as service adaptivity may be needed and require user confirmation), how can user task interruption be kept to a minimum?

Table 2. Smart Space Management & Telecoms/Enterprise Management analysis

2.3 Key Smart Space Management Requirements

Sections 2.1 and 2.2 have identified management requirements that need to be addressed in to order to empower smart spaces and smart space users. The key management requirements can be summarized as:

- Support for self management by devices within a highly intelligent/flexible smart space management architecture;
    - Support for management of context information, especially by smart space participants;
    - Support for dynamic service adaptation and composition from components;
- Support for non-technical users to play the roles of participant, programmer and operator of a smart space;
    - Support for participants' interactions with the smart space in terms of tasks, including representing management services as tasks;
    - Easy and dynamic composition of management services by non-technical operators;
    - Significant support for dynamic negotiation of migration of smart space participants and services;
    - Operators will require flexibility in applying management policies such that they are driven by what participants are in the space at the time, what resources are in use and operational goals and objectives.

Of the above key requirements, the authors believe that the most critical requirement is that of supporting non-technical users in operating the smart space. Unless this requirement is met, smart spaces are destined to remain within the realm of disjoint, non interoperable point applications and will not achieve the critical mass that is required to make the vision a reality. Taking as an analogy the success of the world wide web, the accessibility for the ordinary user in terms of using, programming and managing it proved a key factor in making its use ubiquitous.

3.0 Towards a managed Smart Space

3.1 Smart Space Management Framework

Until users can also "operate" a smart space in the same easy manner as they will be able to "programme" a smart space, the critical mass of smart space deployment that is needed to realize the ubiquitous computing vision will be difficult to achieve. Thus the user in an operator role will require the capability of dynamically composing/adapting management tasks within the smart space at design and runtime. This particular requirement provides the main motivation for the development of the Smart Space Management Framework. Due to the rapidly evolving and heterogeneous nature of smart spaces, this framework needs to incorporate a clear separation between technology neutral and technology specific parts.

The Smart Space Management Framework has been inspired by the work of the EU project called FORM [Form01]. The project has developed an Open Development

Framework (ODF) that takes a model-based approach to software development and enables the construction of management systems from reusable management components (termed Building Blocks). The ODF methodology comprises two guidelines. The first guideline is targeted at Building Block development. The second guideline provides a 'Business Process Driven' approach to management system construction from re-usable building blocks by explicitly modeling the required system processes and their constituent system activities. The guideline uses these system activities to determine the building block contracts needed to implement these processes.

FORM has validated the framework by constructing the management systems involved in the management of a web based application service on behalf of a customer. This included the management systems of the Customer, the Inter Enterprise Service Provider, the VPN Service Provider, the Application Service Provider and a Guaranteed Quality IP Service Provider. The kind of building blocks reused, for example, to construct the IESP management system included: Order Handling, SLA Management, Customer Reporting, Assurance Configuration, Performance Monitoring and Reporting.

The Smart Space Management Framework (SSMF) is structured into four portions (see Figure 1):

- **Logical Architecture:** The Logical Architecture describes the structural concepts of the Framework and their relationships in a manner independent of any implementation technology.
- **Development Methodology:** The Development Methodology provides the processes and notations needed to develop management components and assemble systems that conform to the framework.
- **Technology Architecture:** The Technology Architecture addresses how the concepts expressed in the Logical Architecture can be implemented using a range of technologies.
- **Reusable Elements:** This portion of the Framework is the repository for reusable products that result when the Framework is applied to a particular application domain.

For the purposes of this paper, focus is placed on outlining the logical architecture and development methodology portions of the SSMF, as it is too early to consider the technology architecture and reusable elements portions.

The logical architecture is described in terms of the following:

- A set of architecture concepts representing the main artifacts used in the SSMF;
- A set of models describing the main structural elements of the Logical Architecture and how they relate to each other;
- A set of abstract roles representing potential users of the SSMF at different points in the lifecycle.

The development methodologies portion of the SSMF specifies notations and processes for the development of models and software in accordance with the framework.

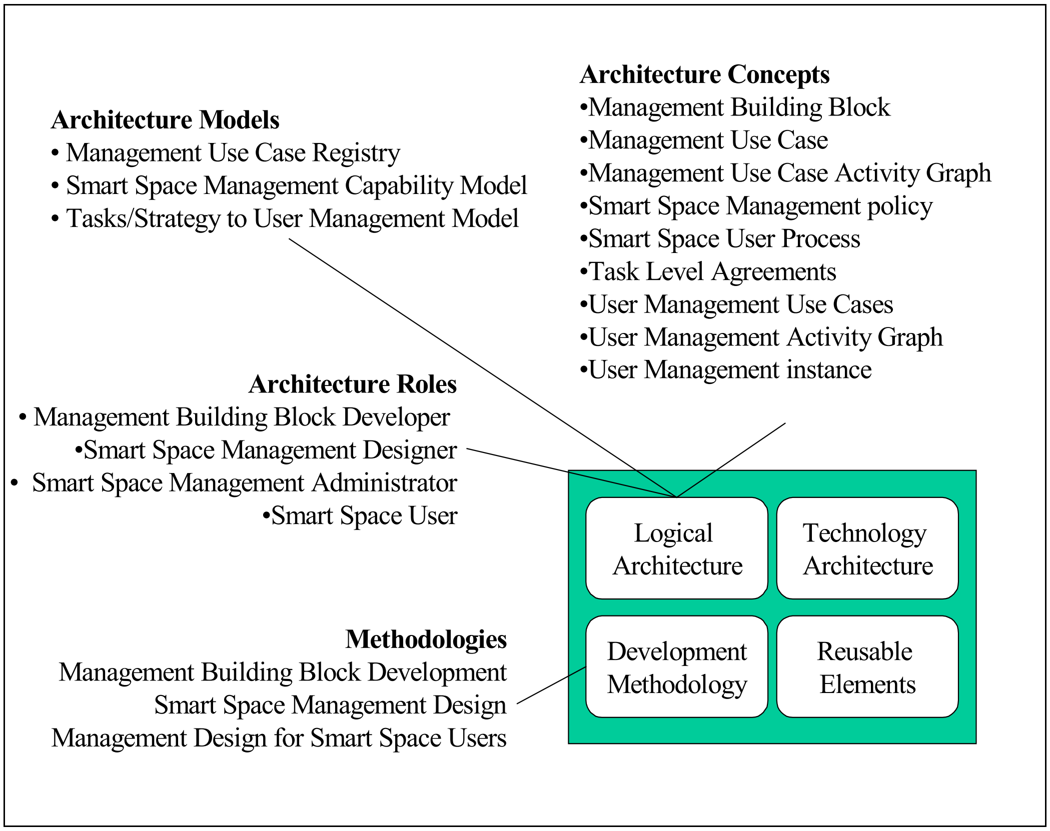

Figure 1. Smart Space Management Framework

3.1.1 Logical Architecture concepts

This section outlines each of the main architecture concepts used in the framework. In order to avoid inventing a new terminology set for engineering of smart space management systems, Unified Modeling Language (UML) terms [OMG01] have been used to represent a concept that is needed in the framework, whenever there is a good match.

The **management building block** concept represents the management service component discussed earlier. What functionality a building block offers can be described using UML use cases. In the logical architecture these are called **management use cases**, as they can be used to describe the management tasks that a building block can perform. Building block "operations" can be used to describe services on the building block as a whole that can be used independently of each other. Thus from an operator perspective, the management building block use cases will describe the capability of the software to undertake management tasks and the management system operations could be used for lifecycle management actions on the management service component (deploy, suspend etc.).

The **management use case activity graph** concept is used to describe possible patterns of management service component compositions that are allowed. In UML, an activity graph defines a computational process in terms of the control flow and object flow among its constituent actions. In the Logical Architecture, management use cases are modeled in the activity graph (rather than management building blocks) as from an operator's perspective it is ensuring valid management task combinations which is of most importance. A particular management use case can be offered by one or more different management building blocks. The **smart space management policy** concept is used to describe the policies that need to be implemented by a smart space management system. For example, to express management building block concurrent execution constraints.

The **smart space user process** concept is introduced to represent the tasks and strategies in which a smart space user wants to engage. In UML activities and activity graphs are used to represent business processes, and in the Logical Architecture activities and activity graphs are used to represent the desired user processes. Only those smart space user processes that involve management building blocks are of interest within the logical architecture. For this reason, the concepts of **user management use case** and **user management activity graph** have been developed. In addition, in order to represent user requirements for quality of task support, the concept of **task level agreement** is included.

The **user management instance** concept describes a runtime thread of execution of management for a user. It includes the management use case activity graph that has been instantiated and identification of the management building block instances that are being used to realize requeted user management use cases.

3.1.2 Logical Architecture models

The **Management Use Case Registry** is a repository of registered management use cases. The assumption here is that in order to enable dynamic composition of service components, the specification and semantics of management use cases needs to be standardized. Taking an analogy with what is happening in W3C at the moment, it is equivalent to saying that a UDDI like approach to interface registration combined with a Semantic web like approach to semantic definition will be needed. Thus vendors of management building blocks will be either able to add new use case definitions to the registry, or register that their building block implementation is defined by a set of management use cases already registered.

The **Smart Space Management Capability Model** describes the management use cases, management use case activity graphs, smart space management policies and quality of service that a Smart Space supports.

The **Tasks/Strategy to User Management Model** maps tasks, strategies and task level agreements as expressed in user terms to user management use cases, user management activity graphs, and quality of service specifications.

3.1.3 Logical Architecture roles

The **Management Building Block Developer** designs and develops management building blocks. The management use case(s) used to describe the functionality of the developed management building block are based upon (or added to) the use cases described in Management Use Case Registry.

The **Smart Space Management Designer** is responsible for taking the management building block(s) that have been acquired, and altering the smart space capability model to reflect the capabilities of the building block that are desired to be exposed (e.g. which management use cases, what management use case adaptivity allowed, etc.)

The **Smart Space Management Administrator** is responsible for the lifecycle management of all management building blocks.

The **Smart Space User** expresses what tasks they would like supported by the smart space, the strategies that they desire to be followed in executing those tasks, and the task level agreements that are desired. Some of the tasks/strategies require management system support. Task level Agreements require management system support, in terms of tools for expressing TLA requirements and mapping these into appropriate assurance mechanisms.

3.1.4 Development Methodologies

The **Management Building Block Development** methodology will be used by developers to ensure development of management building blocks is in accordance with the SSMF. This methodology is likely to be inspired by current best practice in component based development within IT (e.g. the use of UML, RUP) and within the management system community (e.g. EU project FORM).

Smart Space Management Designers will use the **Smart Space Management Design** methodology to guide the development and evolution of their smart space capability model.

Finally the **Management Design for Smart Space Users** methodology will differ from the other methodologies substantially as it will be primarily targeted at providing guidance to non-technical users. It is likely to build upon or integrate with the methodology that will emerge for programming of smart spaces by users.

3.2 Example Smart Space Management Scenario

In this scenario, there is an education provider who delivers educational services directly to the user and schedules tutorials for group work. The education provider can provide the educational content to the student in many different forms (e.g. web based browsing, audio streaming, video streaming) depending on the circumstances and smart space occupied. In this scenario, the education provider specializes in providing eLearning courses about UML.

Having identified the education provider, the student will use a management system tool that allows the student to express their preferences for service delivery and negotiate the quality/cost agreements that will be involved. The exact user interface for such a tool is for further study but must have the following properties: easy to use; terms used for expression appropriate to the tasks being negotiated and follows a normal bartering style. The essence of the dialogue is outlined in Table 1. Having negotiated terms and conditions for use of the service, the student is free to use at any time.

Dialogue	Information Captured
Student: I would like to learn about the *Unified Modeling Language (UML)*.	What learning student wants to undertake
Provider: Tell me what you know about UML or other development methodologies already.	
Student: I do not know UML but have learnt about Booch's OMT about a year ago.	Provides hint of background knowledge
Provider: How do you learn best?	
Student: I learn best as part of a group, I don't like a lot of reading, and enjoy the challenge of working out solutions to real world problems.	Provides hint of learning style
Provider: Where will you be undertaking your learning, for example at home, at our centre?	
Student: I would like to undertake individual learning at home or sometimes while I am on the move on my PDA, or sometimes in the local library learning rooms. I would prefer face to face group activities.	Provides information about smart spaces that could be used in delivery: home; PDA; local library; education centre rooms
Provider: We can provide videos of interviews with UML designers. At home you would need to access this via a medium quality communication link at a cost of X dollars per video or you can access for free in the local library learning room?	
Student: I will access the videos via the library room	
Loop: Negotiate details of service possibilities and usage	Preferred services composition and task level agreements, per smart space

Table 3. Essence of negotiation dialogue

The result of this dialogue has impact upon management systems in direct and indirect ways. For example, the mobility requirement of the student has impact directly on the various smart space management systems that will be involved. Whereas the application requirements expressed have indirect impact, in that the application logic places requirements on the management of the smart spaces. For example, the task level agreements will be translated into additional quality of service policies for the management systems to enforce.

If the student wants to begin a learning session at home and continue the session while travelling to the education center for a tutorial on the subject, the following kind of management system actions would be involved.

The first task of the student illustrated is to express “do not disturb” as the student wishes to have quality learning time. The home management system would accordingly alter the user’s context and reconfigure ongoing services accordingly (e.g. direct all non-urgent incoming calls to voicemail. This would involve configuration using the *context manager* and *ongoing tasks* management building blocks.

The student then requests the smart space to start the learning task using the most appropriate smart space appliance near the student (e.g. Television, PC etc.). This causes the home management system to alter the student’s context accordingly. The education provider’s *learning tracking* management building block is used by the education service software, in order to initiate the session appropriately. Similarly the *session configuration* management building block of the connectivity provider is involved in setting up the appropriate connection bandwidth. All three management systems will use *quality of service* management building blocks in cooperation so as to ensure successful delivery of the service.

As the student prepares to leave home to travel to the tutorial, the student first signals that he is willing to be disturbed on the journey by other tasks that may require his/her attention. The home management system acts accordingly by using the *context manager* and *ongoing task configuration* management building blocks. The student then asks the smart space to switch the learning session to his/her mobile device. Similar management building blocks are involved in setting up the new session as were involved in the original startup of the service. This is because the type of education content that can be streamed as part of the session might change, with consequences on connectivity etc. Assuming successful switching and adaptation of the learning session to the mobile device, the student continues the learning task whilst traveling to the tutorial.

Meanwhile, just before the tutorial is due to begin, the tutor seeks status of student learning from the *group summary* management component of the education provider management system. This management component uses the task and context information of individual users to gather a view as to the state of learning as a whole for the group. The tutor can then decide a strategy for the tutorial, whether to concentrate purely on working one to one with individual students or to proceed to work with the group as a whole. Depending on the strategy expressed by the tutor, the *tutorial planner* management building block will suggest what challenging tasks to set for students.

Once all the students arrive in the education room, the tutor proceeds and starts the tutorial. Depending on the policy set for the room or by the tutor, individual mobile devices may be prevented from operating during the tutorial upon any tasks that are not associated with the tutorial.

4.0 Conclusions and Future Work

From the analysis undertaken for this report, it is clear that the management of smart spaces in multiple application and multiple operator environments requires further research. Simply treating the management of smart spaces and their interworking as just a more diverse and scalable example of distributed systems management to which emergent management solutions can be applied, is insufficient. This paper has described the key management requirements that need to be addressed, and in particular identified support for non-technical operators of smart spaces as a prerequisite before the vision of interconnected smart spaces can be achieved.

Key to meeting this challenge is the availability of dynamically composable management services by non-technical operators. Based on the proven approach of the EU FORM project towards management system composability, the Smart Space Management Framework (SSMF) is outlined which enables this dynamic smart space management service composition.

It is expected that the Smart Space Management Framework will be developed further over the coming year through an Irish government funded project called *MZONES* (Management Zones).

References

[Abowd98] Abowd et al, *Ubiquitous Smart Spaces*, A white paper submitted to DARPA (in response to RFI), February 1998.

[Anind99] Anind et al, *Context Based Infrastructure for Smart Environments*, GeorgiaTech. Proceedings of MANSE '99

[Ark99] Ark, *A look at human interaction with pervasive computers*, IBM Systems Journal, Volume 38 Number 4, 1999.

[Bradley99] Bradley et al, *Augmented Reality through Wearable Computing*. Presence, Vol. 6, No. 4, August 1997, pp. 386-398.

[Cohen99] Cohen et al, *Meeting the Conceptual Needs of Intelligent Environments: The Metaglue System*, Managing Interactions in Smart Environments, Springer-Verlag, 1999, pp. 201-213.

[Dobson99] Dobson et al, An integrated system for managing intelligent buildings, Proceedings of the 1st International Workshop in Smart Environments (MANSE '99), Springer Verlag, 1999.

[Ducatel01] Ducatel et al, Scenarios for Ambient Intelligence 2010, IST Advisory Group, EU publication, 2001.

[Fishkin98] Harrison, Beverly L., P. Fishkin, Kenneth, Gujar, Anuj, Mochon, Carlos Want, Roy: *Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces; CHI* 98, Los Angeles CA, USA, 18-23. April (1998).

[Form01] EU Form project , *Deliverable 8 Annex C: A Logical Architecture for an Open Development Framework for Component-based Management Systems*, IST-1999-10357/LMD/WP4/0520_AnnexC, September 2001.

[Fox00] Armando Fox, Brad Johanson, Terry Winograd, Patrick Hanrahan, *Integrating Information Appliances Into an Interactive Workspace*,IEEE Computer Graphics & Applications (special issue: "Off the Desktop"), May/June 2000.

[Goodman00], Goodman, *Wireless Internet: Promises and Challenges*, Computer.org, IEEE Computer Society, August 2000.

[Grimm01], Grimm et al., Programming for Pervasive Computing Environments, University of Washington Technical Report UW-CSE-01-06-01, June 2001.

[Hong01], Hong et al., *Integrating Context Services through automatic path creation*, Berkeley, Proceedings CHI-2001 Workshop, 2001

[Kiciman00] Kiciman & Fox, *Using Dynamic Mediation to Integrate COTS Entities in a Ubiquitous Computing Environment*. Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K), Heidelberg, Germany: Springer Verlag. 2000.

[LeGal99] Le Gal et al. *SmartOffice: An Intelligent and Interactive Environment.* INRIA. Proceedings of MANSE '99

[McGinnes01] Mc Ginnes, *Accelerated Business Concept Modelling: Combining User Interface Design with Object Modelling,* Chapter 1, Object Modeling and Interface Design, Mark van Harmeln (ed.), Addison Wesley, 2001.

[Mills00] Mills K., *AirJava: Networking for Smart Spaces*, National Institute of Standards and Technology, US.

[Minar 99] Bradley J. Rhodes, Nelson Minar, and Josh Weaver. *Wearable Computing Meets Ubiquitous Computing: Reaping the Best of Both Worlds*. In Proceedings of the International Symposium on Wearable Computers (ISWC '99), October 1999.

[Moeslund01] T.B. Moeslund and E. Granum, *A Survey of Computer Vision-Based Human Motion Capture*. Computer Vision and Image Understanding, 2001. to appear

[Neti01] G. Potamianos, C. Neti, G. Iyengar, and E. Helmuth, *Large-vocabulary audio-visual speech recognition by machines and humans*, Proc. Eurospeech, Aalborg, 2001.

[Ngoss01] TeleManagement Forum, *The Next Generation Operations and Support Systems roadmap*, 2001.

[OMG01] Open Management Group, Unified Modeling Language version 1.4 specification, OMG document reference "formal/01-09-67", 2001.

[Pascoe98] Pascoe, "*Adding generic contextual capabilities to wearable computers*", in Proceedings of 2nd International Symposium on Wearable Computers, October 1998, pp. 92-99.

[Phillips01], Phillips et al, *Understanding Existing Smart Environments: A brief classification*, In Workshop on Ubiquitous Computing in Domestic Environments, pages 1--4. Web publication, September 2001. Lancaster University,

[Portolano] *Charting the new territory of invisible computing for knowledge work*, A proposal to DARPA in response to BAA 99-07, Xerox and University of Washington.

[Rosenthal00] L. Rosenthal, V. Stanford, *NIST Information Technology Laboratory Pervasive Computing Initiative*, IEEE Ninth International Workshops on Enabling

Technologies: Infrastructure for Collaborative Enterprises, June 14-16, 2000, NIST, USA.

[Schmidt99] Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., Van de Velde, W., *Advanced Interaction in Context*,. 1st International Symposium on Handheld and Ubiquitous Computing (HUC99), Germany, Sept. 1999.

[Streitz01] Streitz et al, ***Roomware****: Towards the Next Generation of Human-Computer Interaction based on an Integrated Design of Real and Virtual Worlds*. In J. A. Carroll (Ed.): Human-Computer Interaction in the New Millenium, Addison Wesley, pp. 551--576. 2001.

[Tennenhouse97] *A Survey of Active Network Research*. IEEE Communications Magazine,
Vol. 35, No. 1, pp80-86. January 1997.

[Wang00] Wang and Garlan, *Task Driven Computing,* Technical Report from CMU, CMU-CS-00-154, May 2000

[WashU00] *Invisible Computing for Knowledge Work*, A proposal to DARPA in response to BAA 99-07 , Washington University/ Xerox Parc, 2000.

[Weiser99] Weiser, Gold and Brown, *The origins of ubiquitous computing research in the late 1980s*, IBM Systems Journal, Volume 38 Number 4, 1999.