An architecture for User-centric Management of Intelligent Environments

Venet Osmani, Keara Barrett, Ray Carroll, Brendan Jennings, Sven van der Meer
Telecommunications Software & Systems Group
Waterford Institute of Technology
Cork Road, Waterford, Ireland
E-mail: {vosmani, kbarrett, rcarroll, bjennings, vdmeer}@tssg.org

Abstract – Ubiquitous computing environments are intelligent environments that have a wide array of embedded sensors and other artefacts that provide information regarding the current state of the physical environment. Utilisation of such information, known as context information has the potential to alter the natural surroundings of users in order to adapt the environment to suit the user's needs and assist in their tasks. These alterations in the environment should occur both transparently and seamlessly in order to minimise the impact on user's attention in such a way that the user's focus is not shifted from the task at hand. Satisfying user requirements in this manner, particularly in an environment that is so complex and dynamic raises a multitude of challenges. It is our belief that these challenges can be attributed to three distinct domains, namely management of personal information and management of context information combined with policy based management system. In the proposed paper we describe a management architecture for these intelligent environments that, through coordinated invocation of system and user policies triggered through the analysis of context information, taking into account business goals, can achieve a balance between user's desires and intentions while upholding system constraints.

I. INTRODUCTION

The main concept behind the ubiquitous computing initiative is to intertwine our physical surroundings with computing environments to provide non-intrusive, transparent computing ability anytime, anywhere. Research into ubiquitous computing, otherwise known as pervasive computing or ubicomp, aims to make computing capabilities so omnipresent and seamless that it is beneath the level of human attention [1]. This was the vision of Mark Weiser, the pioneer of pervasive computing, who began his seminal article, "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it" [2].

Advances in fields, such as distributed and mobile network, sensors, embedded systems, middleware and power consumption have made ubiquitous computing environments (UCE) a reasonable aspiration and many test-beds and prototypes exist today. While these advances bring the vision a step closer, their collaboration, essential for UCEs, creates new management challenges. The universal adoption of UCEs will not materialise without an *innovative intelligent management system*, despite this UCE management gets little attention. Most research carried out in this domain focuses on a narrow area, for example mobility management, trust management, context management or data management.

UCEs are inherently highly dynamic, since they are designed to respond to changes in the physical environment and to changing user intentions. The overall challenge from the intelligent management system perspective is how to respond to the continuum of changes in the environment and user behaviour, without compromising the underlying infrastructure that composes the UCE and the system requirements dictated by the organisational goals. Management for UCEs should be handled intelligently by controlling evolution and configuration changes without explicit user or administrator action. In other words, dynamic adaptive management is sought and methods that support such adaptable management are required.

Proposed in this paper is an intelligent management system, called the Ubiquitous Management Architecture (UMA). UMA comprises three major components, namely a Personal Information Manager (PIM), Context Manager (CM) and a Policy Controller (PC). These components and their collaboration attempt to address the challenges identified in this paper and in doing so address the outlined objectives of ubiquitous computing.

In this paper we first discuss related work in the areas that influence the development of the UMA. We then elaborate on our view of the key challenges in intelligently managing ubiquitous computing environments and the approaches required to meet them. In section IV we provide an overview of the architecture itself, describing the main components, namely PIM, CIM and a PC and how they interact. Finally, we outline the future work and conclude.

II. RELATED WORK

In terms of user information there are many systems that deal with some aspects of managing this information, such as vCard [3] and Learner Information Package [4]. These initiatives are typically limited to their own domain of interest. More recently the idea of identity management has gained recognition, describing the management of multiple identities across organisational/domain boundaries. A great deal of work has been done on the Liberty Alliance specification [5], which attempts to implement an identity management solution for internet-based transactions. However, the static nature of the internet is very different from the dynamic nature of ubiquitous computing. Users will require a much wider variety of services provided in a dynamic and personalised way. In particular ubiquitous computing environments have a much larger quantity of context information available, whereas the internet is typically based on limited context information such as an IP address. It is critical that this information is exploited to influence the usage of user information and to protect the users' privacy.

Context-awareness has been an important research area for many years with the Active Badge location system being one of the principal initiatives. However, context-awareness requires management of context information which according to Dey [6] may potentially include a vast amount of raw data. Therefore, a flexible and extensible system that manages context information effectively is of essential importance. While work has been done in this area, existing systems typically do not provide an adaptable solution in order to serve a wide variety of context-aware applications. Current approaches on context information management have a tendency to focus on a specific area, for example, modelling a meeting environment [7]. While in [7], a user can specify policies to deal with privacy issues, there is no observation for system-wide policies. With context-awareness, an important issue is to increase the level of abstraction of sensed context information, providing deductions that cannot be sensed directly from the environment. This is an important element highlighted in [8] which in general is lacking from surveyed systems for example in [9, 10]. Reasoning over sensed context information provides a means of ensuring that knowledge regarding the environment is consistent, which is typically neglected by the current approaches.

Policy Based Management (PBM) is a research topic that has received much attention as a flexible and adaptive concept for the management of networks and systems. Interest in PBM has been driven by the growing complexity inherent in the administration and management of present-day networks. Heterogeneous devices and network topologies, mobility, greater demands from users and the fluidity of the current and emerging distributed networks and systems, are but some of the aspects that make PBM an attractive alternative to other human resource-intensive management techniques. However, the dynamic nature of ubiquitous computing environments renders it impractical for an administrator to specify and update suitable device and technology-specific policies to be stored in the repository and used to manage and control the changing and/or maintaining of the state of one or more manage objects [11]. One possible approach for automatically generating enforceable, valid policies is to refine or translate policies, from a higher level of abstraction.

In 1999 researchers at Hewlett-Packard Laboratories published a report on a practical approach for policy refinement called the Policy Wizard Engine for Refinement or POWER prototype [12]. However, the POWER prototype requires large interaction and system knowledge from a human operator to set up the refinement templates for a specific system and to translate the policy from one level of abstraction to another. This refinement protocol does not include analysis capabilities and is more of a guide for policy refinement than an automated policy refiner, which is not suitable for ubiquitous computing environments. [13] describes a goal-based approach to policy refinement. The system uses event calculus in conjunction with abductive reasoning to derive the sequence of events that achieve the desired goal. However, the system expects the users to provide a representation of the system description, in terms of the properties and behaviour of the components and the goals that the system must satisfy. The user is also expected to define the pre and post conditions of the operations supported by components using state charts. Such interaction from a user must be avoided for policy refinement in ubiquitous computing environments.

While individually each of these areas has a considerable volume of work done, none completely address their own specific challenges encountered in ubiquitous environments. Furthermore little work has been undertaken in combining these to provide a holistic management solution.

III. MANAGEMENT CHALLENGES FOR UBIQUITOUS COMPUTING ENVIRONMENTS

Ubiquitous computing environments are intrinsically highly dynamic in nature, since they are designed to respond to changes in the physical environment and to changing user intentions. The overall challenge from a management perspective is how to intelligently reconfigure managed entities to respond to the continuum of changes in the environment and user behaviour, without compromising the computing infrastructure itself, or the business goals of the entity operating it. We contend that there are three main activities necessary to meet this challenge: (i) gathering and interpreting context information relating to the current state of the environment; (ii) capturing and using information relating to user attributes, desires and intentions; (iii) utilising both contextual and user information in a management decision making process that ensures business goals are achieved.

Context information, generally relating to characteristics of the physical environment within which a user interacts with a ubiquitous computing system, can be viewed as providing notification of events that require management decisions. However, collecting and processing data produced from the artefacts embedded in the environment is a challenging task that requires dominance of a flexible and scalable context management system. One of the main challenges stems from the nature of ubiquitous environments being inherently dynamic and heterogeneous, with multiple context sources providing diverse data at discreet intervals, adhering to different data models whilst using various communication protocols. Therefore, provided data typically will not be suitable to be exploited directly by context-aware entities, making it difficult to get a consistent view of the context. Enormous amount of raw data yielded by ubiquitous computing environments makes the process of data handling and consuming very difficult due to the sheer volume and diversity of the same. Acquired context data will need to be formatted, processed or combined with data from other context sources to generate meaningful, low-volume and pertinent context information.

Once the context data has been collected and processed, the next step is the process of reasoning. Typically, context information produced by the entities in the environment will be at a low point in the abstraction scale, resulting in information that has little or no value when utilised in isolation. The process of reasoning over context information employs methods of knowledge engineering which is a branch of science that is concerned with the creation, maintenance, understanding and use of information about a subject or collection of subjects of interest. Reasoning combines low-level information to produce logical conclusions, with the effect of elevating the abstraction level of context information by inferring conclusions that cannot be sensed directly from the environment. Thus a major challenge lies in creating a generic system in the sense that it would infer deduction across diverse domains.

As a result, a mechanism that will transform, process, aggregate and reason over context data to make it appropriate for consumption by context-aware applications is essential. We argue that context management systems for ubiquitous computing environments should incorporate processes for composition of context data sources and for aggregation/inference of that data into useful context information. Furthermore, these processes should be configured by the management system, so that only events of interest trigger the decision making process.

Of course many management decisions will be triggered directly as a result of user interaction with the ubiquitous computing system. To ensure the system is user-focused it is important that the management system has access to user information, in particular user's preferences to adapt the environment. A major challenge is therefore the aggregation of users' information into a commonly accessible repository. However this repository must provide more than just basic database functionality. Instead it must be a complete management system for user information governing how personal information is distributed and used within the ubiquitous computing environment, as well as how it is administered and controlled in general. We believe that such information should be kept under direct control of the user as much as possible. The user should retain knowledge of his/her preferences for interaction with the system, rather than having the system store and access information relating to how users interact with it.

Once context and user information are collected and processed, a mechanism that uses this information to change and/or maintain the state of environment is required. While the principal intent of the ubiquitous environments is to provide personalised services on demand, system requirements (which may contrast with or obstruct user requirements) must be upheld where applicable, raising an intricate challenge. The challenge presented is exacerbated further when system and user requirements change frequently and randomly, and when the environments onto which these requirements are enforced arbitrarily change. This is the case with today's evolving businesses, insistent users and unpredictable ubiquitous computing environment. Thus, the mechanism that employs context and user information to change or maintain the state of the environment must do so in an efficient manner. The mechanism must also be capable of dealing with the persistent changes that are discrete and continuous in nature originating from business, user or environmental alterations. The vast numbers of heterogeneous entities involved in ubiquitous environments introduces further challenges in defining the overall environmental state and controlling and managing a state change. We consider policy based approaches as the most attractive solution for complex systems and as a useful approach in addressing the unique challenges encountered in meeting user requirements within volatile ubiquitous environments. In particular, the vision of a policy continuum in which high-level organisational polices are automatically refined to produce low-level, device-specific policies that can be applied efficiently in highly dynamic environments appears an ideal solution to the ubiquitous computing management challenge.

IV. A UBIQUITOUS MANAGEMENT ARCHITECTURE

The proposed Ubiquitous Management Architecture (UMA) is depicted in Fig.1. It is a two level model, in which individual ubiquitous computing systems – *Smart Spaces* – employ a policy based management system to control discovery, monitoring and execution of services, with higher level management decisions being made at the *M-Zone* level. An M-Zone corresponds to an administrative domain, where individual Smart Spaces under its control are operated by a single entity.

At the Smart Space level management is realised via a Policy Decision Point (PDP) that controls a set of Policy Enforcement Points (PEP), which in turn control services and resources. Policies enforced by the PDP are stored in a Policy Repository and are simple in nature, for example white/black lists for access control. These Smart Space policies are re-configured and deployed at intervals by the PDP residing at the M-Zone level (the M-Zone PDP), thus allowing control over the operation of the Smart Space. The Smart Space also contains a Service Repository (SR), which maintains data relating to the state of all services available within the space.

At the M-Zone level there are three types of components: the Context Information Manager (CIM), the Policy Controller (PC) and Personal Information Managers (PIMs). There is one CIM and one PC for every M-Zone and one PIM for every user interacting with a Smart Space under control of an M-Zone. Users are expected to provide access to their PIM when they identify/authenticate themselves to an M-Zone. PIMs essentially contain a repository of a user's personal information and also provide considerable functionality in terms of distributing and restricting this information when appropriate. In the context of this system the usage of personal information can occur in a number of ways and for a number of reasons. Firstly personal information is of a sensitive nature and privacy must be ensured. Also in any given scenario only a certain quantity of user information is relevant, and so selective distribution and usage of this information is more efficient and secure. In terms of how information is used, this can occur in two ways. The PIM can, based on the user's current context and personal information such as preferences and history, deduce what service is required and actively request that service from the ubiquitous management system. Alternatively, in the case where a service or application requests user information from the PIM, this request must be evaluated and the appropriate information distributed. Personal information in this context will relate to attributes that influence the interaction of the user with Smart Space services, as well as "user policies" embodying preferences for the manner in which the user would like the Smart Space to react in certain circumstances. For, example user policies could represent a user's preferences for heating and lighting conditions whilst working at his/her desk, or specify privacy protection actions to take place in the presence of other individuals.

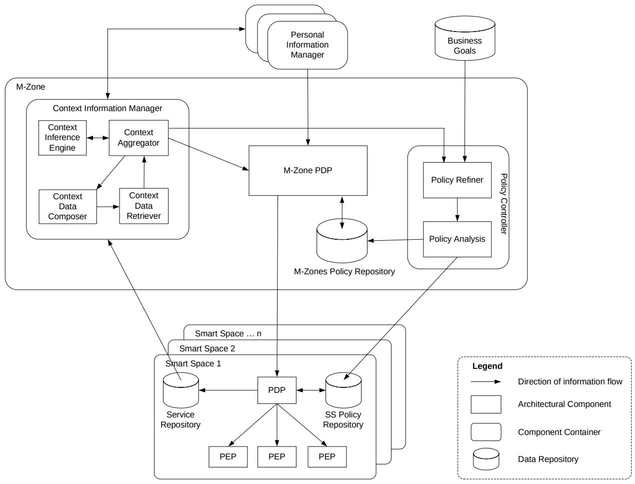

Fig.1 Ubiquitous Management Architecture (UMA)

The CIM collects and aggregates context information directly from Smart Space context sources. As part of its context analysis process it may be configured to access user information from the relevant PIM, so that the context information reported to the PC is semantically rich in nature. The PC plays a controlling role, in that it requests the CIM to notify it of particular events and it specifies what kind of user specific information to be incorporated in such notifications. This reduces the number of high-level context notifications, decreasing the overhead involved in the policy decision process.

The Policy Controller has overall responsibility for managing and controlling the changing and/or maintaining of the state of the environment, for example granting/denying access to managed video services to another managed object in the Smart Spaces. It contains policy refinement and analysis capabilities that allow it automatically translate high level business goals to progressively lower-level M-Zone and Smart Space policies, in a manner that accommodates user preferences to the highest degree possible. However, there is a significant gap between business goals and requirements (defined as high-level policies) and device/technology specific configuration rules (low-level policies) that must be bridged automatically to make policy-based management useful for UCE. Bridging this gap will be done through iterative transformation along a policy hierarchy or policy continuum with the aid of an information model. Analysis of the low-level policies will be performed to ensure that the policies are (i) correct – an accurate representation of the business goal (high level policy) at a different viewpoint, (ii) consistent – no conflicts, static or dynamic and (iii) valid – the refined policies can be executed on a tangible managed object in the environment [14]. The M-Zone PDP uses these refinement and analysis capabilities when events trigger policy decisions and/or re-configuration of policies in the system. These events can come from three sources: changes in business goals, changes in environmental context reported by the CIM and direct user interactions with the system notified by PIMs. Decisions relating to events take into account relevant user policies, but only to the extent that actions generated are not in conflict with M-Zone policies. Generally these actions result in re-configuration of policies deployed in one or more Smart Spaces, whose enforcement affects the manner in which users can interact with services, i.e. the state of a managed object is changed.

V. FUTURE WORK

The immediate goal of this work is to formally design and specify the system which in turn will lead to a prototype implementation. Following the implementation, this prototype will be tested using the Smart Home test-bed which has been designed and built for the evaluation of ubiquitous services and software. The test-bed incorporates a set of technologies that include an Ultra Wide Band (UWB) positioning system, a set of X.10 control modules, smart camera, a number of PDAs and Smart Phones, Multimedia Server, Tablet PC and various household appliances. Various test scenarios and metrics will be defined and used to evaluate the architecture to enable further refinement and possible modifications.

VI. CONCLUSION

It is widely accepted that the ability to anticipate user intentions will be a critical factor in the success of ubiquitous computing. However, user requirements are complex and changeable. In addition, the environments in which these requirements are fulfilled are also heterogeneous and unpredictable, making manual management unrealistic. This implies that an intelligent system to configure these environments towards the user requirements while taking into account system policies is necessary.

This paper has outlined the major challenges in building an adaptive ubiquitous environment that can adjust to users' goals. Existing work in this area has shown that management of these environments is generally confined to a subset of these challenges, rather than adopting a holistic view. The architecture presented in this paper attempts to address the challenges in a synergic manner, combining personal information, context information and policy based approaches. It is the interaction of these three distinct components that produces an innovative intelligent management system for governing ubiquitous computing environments, pushing technology into background while anticipating and responding to user intentions.

ACKNOWLEDGEMENTS

This work has been carried out under the M-Zones research programme, which is a 30+ person strong initiative funded by the Higher Education Authority of Ireland (www.m-zones.org). The M-Zones research programme is primarily interested in the development of novel management and control systems for integrated Smart Spaces. This includes the development of novel information and communications management technology to support dynamic integrated management for users, information appliances, and Smart Space infrastructure.

REFERENCES

[1] Brian Meyers, "Pervasive Computing: It's Everywhere", PC Conversations Vol. 12 Issue 6, 2001, Available: http://www.smartcomputing.com/editorial/article.asp?article=articles/2001/s1206/45s06web/45s06web.asp

[2] Mark Weiser, "The Computer for the 21st Century", Scientific American, pages 94-100, 1991, Available: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html

[3] T Howes F Dawson, "IETF RFC 2426", vCard MIME Directory Profile, 1998, Available: http://www.ietf.org/rfc/rfc2426.txt

[4] I.M.S. Corporation, "IMS learner information package (LIP) specification", 2004, Available: http://www.imsproject.org/profiles/index.cfm

[5] Liberty Alliance, "Liberty ID-FF Architecture Overview." 2004, Available: http://www.projectliberty.org/specs/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf

[6] Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications", Georgia Institute of Technology, 2000, Available: http://www.cc.gatech.edu/fce/ctk/pubs/dey-thesis.pdf

[7] Tim Finin Harry Chen, Anupam Joshi, "An Intelligent Broker for Context-Aware Systems", Adjunct Proceedings of Ubicomp 2003, Seattle, Washington, USA, 2003, Available: http://www.csee.umbc.edu/%7Ehchen4/paper/hc-ubicomp03-poster.pdf

[8] Apratim Purakayastha Norman Cohen, John Turek, Luke Wong, Danny Yeh, "Challenges in Flexible Aggregation of Pervasive Data", IBM Technical Report RC21942, 2001, Available: http://domino.watson.ibm.com/library/cyberdig.nsf/1e4115aea78b6e7c85256b360066f0d4/071dab93d3fe4a72852569de0056508a

[9] David Kotz Guanling Chen, "Context aggregation and dissemination in ubiquitous computing systems", In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), pages 105-114, Callicoon, New York, June 2002., 2002, Available: http://www.cs.dartmouth.edu/papers/chen_abstraction.pdf

[10] Apratim Purakayastha Norman H. Cohen, Luke Wong, Danny L. Yeh, "iQueue: A Pervasive Data Composition Framework", Proceedings of the Third International Conference on Mobile Data Management, 2002, Available: http://www.research.ibm.com/sync-msg/MDM2002.pdf

[11] John Strassner, *Policy-based Network Management: Solutions for the next Generation*: Morgan Kaufmann Publishers, 2004.

[12] A Baldwin M Casassa Mont, C Goh, "POWER Prototype: Towards Integrated Policy-Based Management", 1999, Available: www.hpl.hp.com/techreports/1999/HPL-1999-126.pdf

[13] Emil Lupu Arosha Bandara, Jonathan Moffett, Alessandra Russo, "A Goal-based Approach to Policy Refinement", 5th IEEE Workshop on Policies for Distributed Systems and Networks, 2004, Available: http://www.doc.ic.ac.uk/~bandara/research/BandaraA_GoalBasedPolicyRefinement.pdf

[14] A. van Lamsweerde R. Darimont, "Formal Refinement Patterns for Goal-Driven Requirements Elaboration", 4th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE4), 179-190, 1996, Available: ftp.info.ucl.ac.be/pub/publi/96/fse4.ps.gz