

# An Adaptive Architecture for Context-Aware Interaction in Pervasive Applications

Alan Colman, Minh Tran, Jun Han  
Centre for Information Technology Research (CITR)  
Swinburne University of Technology  
Melbourne, Victoria, Australia  
{acolman, mtran, jhan}@ict.swin.edu.au

## Abstract

*Adaptation is one of the key challenges in pervasive computing. Context-aware applications need to respond to dynamic physical and computing environments. In this paper, we propose an adaptive architecture for context-aware pervasive systems. The basis of our approach is that entities (e.g., user, device, application or group) are modelled as self-managed composites. Each composite maintains a representation of its relationships with other entities and any context conditions that affect those interactions. It then uses this context model to monitor and control its interactions. This loosely coupled architecture allows flexibility in configuring and monitoring components of pervasive systems, and thereby helps the systems to be more adaptable.*

## 1. Introduction

Pervasive or ubiquitous computing [23] is envisaged that computational power will be available everywhere and that heterogeneous computational devices can be integrated into the everyday life environment to support people's social activities and work [1, 11, 16, 17]. Pervasive computing radically changes the way people access information, perform tasks and interact with one another. People can move throughout environments, such as home and office, while having persistent access to information and services anytime and anywhere. For this vision to become a reality, pervasive computing systems must have the capability of adapting to highly open and dynamic environments—both physical and social [2]. To do so, pervasive systems need to be able to understand environments in which the systems operate and provide users with relevant support. Environments include other people, their locations, activities, tools, social relationships and so on.

Pervasive systems can mediate both tasks of individuals and cooperative tasks of a group. In both cases, systems need to understand and adapt to the changes of users' context. In the case of group interaction, systems may also need to provide awareness information that helps users to be aware of

other people in the group (e.g., awareness information about people's locations, presences, availability, activities, and so on). Research from related fields such as CSCW has shown that awareness information is valuable in group interaction because it helps a group to conduct natural and effective interaction.

To date, however, research on pervasive computing primarily focuses on examining systems that utilise context information to support tasks of individual users. Motivated by this and the potential of pervasive system in facilitating group interaction, this research **investigates an adaptive architecture of pervasive systems**, which are context aware and aims to enhance user experience in group interaction.

This paper is structured as follows. In the remainder of Section 1, we present a motivating scenario, identify some research problems illustrated in the scenario, and briefly describe our proposed approach to address those problems. Section 2 presents a discussion the proposed architecture. Section 3 discusses the deployment of context models in the architecture. Section 4 reviews related research. Finally, we conclude the paper with a summary and remarks on future research.

### 1.1 Motivating scenario

On the university Open Day, visitors including prospective students and their parents come to university to gather information about study courses and to explore the types of facilities and services offered by the university. Hence, it is a goal of the university to ensure that relevant information is accessible to visitors, and that they can speak with academics, course advisers and current students about study options and learning experience. To facilitate visitors' experience on the Open Day, the university has decided to develop an Adaptive Context-aware Pervasive (ACAP) computing environment. The ACAP system is *not* a monolithic system. It is rather a loosely-coupled and open set of related services (a service eco-system) that can be composed to support information delivery and collaborative interaction amongst individuals and groups.

The services that can be composed in various configurations using the ACAP system include identity

management systems for individual and groups, location-identity matching and navigation services, multi-media delivery services, media presentation, formatting and compression services, content-based information services that interface with the university systems, tools for supporting group interactions and awareness, smart-rooms, and so on.

The service compositions that can be developed to support interaction between heterogeneous devices and between multiple applications, as well as allowing co-located / dispersed users to interact with one another. Visitors are able to use their own mobile devices (e.g., PDAs, laptops and mobile phones) to access various sources of information and to exchange information with other devices allocated around the university (e.g., fixed PCs and big screens). In addition, visitors are also able to interact with other people who are co-located in the same campus or geographically dispersed over different campuses.

The ACAP system needs to be context-aware and adaptive. New services should be able to be dynamically added and integrated into the eco-system. The system should be able to cope with changing requirements of users, and changing levels of service provision from the constituent services. As each of the entities within the system may have different owners, the ACAP system also need to provide mechanisms by which those entities can control what is asked of them and what information they share.

The above scenario typifies the open and dynamic context-aware pervasive computing environments that might become popular and even essential in personal, organisational and social interactions of the future.

An ACAP system as illustrated in the above scenario is a composition of different **entities** (e.g., users, groups, devices, applications) and **relationships**. The relationships between these entities within the composition are **highly dynamic** as entities can join and leave the composition, and as devices and users move through the smart-spaces. Moreover, one entity can establish relationships and interact with multiple other entities of the same or different types. There is no fixed rule on how one entity is allowed to interact with another. Relationships between these entities are dynamically formed and diffused based on the context of which the entities interact. More specifically, users can be a group of individual users or collaborating groups of users interacting (possibly simultaneously) with a range of devices that are running a range of collaborating applications. These applications are in turn composed of a heterogeneous range of services. The relationships among users, devices, applications and smart-spaces are dynamically changing, many-to-many relationships.

## 1.2 Research problems

The above highly open and dynamic, context-aware pervasive computing scenario raises a number of research challenges, which need to be addressed to make such ACAP systems a reality.

Firstly, we need to address the problem of how to model dynamic relationships of groups and manage the mutual awareness between individuals and groups in pervasive systems. For pervasive systems to mediate group interaction, the systems need to understand social relationships between members of a group (referred to as “*social context*”) and the context of a situation in which interaction occurs (referred to as “*situational context*”). In addition, to achieve natural and effective interaction, members of a group need to be aware of each other (e.g., other people’s locations, identities, activities, current tasks at hand, and so on). Furthermore, in open systems members need to have control over their own relationships, and therefore need to have control over how they represent their own contexts. An adaptive architecture should be able to model relationships from various participants’ perspectives.

Secondly, an ACAP architecture needs to integrate entities (individuals and groups) that consume services into existing service eco-systems that provide services to those users. An architecture that has a consistent way representing both service consumers and service providers is therefore desirable. Furthermore, these entities may be running on heterogeneous infrastructures. The architecture ideally would be able to work across a range of such infrastructures with minimal modification.

Finally, there is no widely accepted ontology or formalism for representing context. Although there have been attempts to broadly classify different types of context (e.g. [12]), the description of context is inherently open-end and multi-faceted. An adaptive architecture needs to work with various (arbitrary) ontologies for representing context. It needs to be able to sense changes in context and take appropriate action to adapt to those changes.

## 1.3 Our proposed solution

To address the above problems we propose an architecture for context-aware pervasive systems. The architecture is based on ROAD [9, 10] which focuses on *relationships* between entities of a system rather than on the entities themselves. We argue that from an entity’s point of view, a relevant *social context* is always defined in terms of its relationships with other entities. The proposed architecture ‘wraps’ with interaction-structures (composites) the entities that need to adapt. These composites represent the entities’

social contexts. Each composite has an **internal** view of itself, and **external** participatory views of other composites.

Our ROAD-based architecture creates small runtime models that are encapsulated into self-managed composites. The model/composites contain no functional entities in themselves, but are rather interaction-structures that define a set of functional relationships (referred to as **contracts**) between entity roles. These contracts:

- define permissible interactions over those functional relationships,
- define non-functional requirements related to those interactions along with any context conditions that affect those requirements, and
- monitor the performance of entities with respect to those interactions.

Externally, these composites present themselves as services. Indeed, the ROAD architecture treats all entities within the system (individuals, groups, devices, applications, third party services, composites) as services. Developing pervasive systems is, therefore, considered as a dynamic service modelling and composition problem.

The proposed architecture will enable each entity to control its own social context (and thus access to itself by others) by selecting and regulating its own external relationships taking into account environmental context information (location, technical environment, etc.)

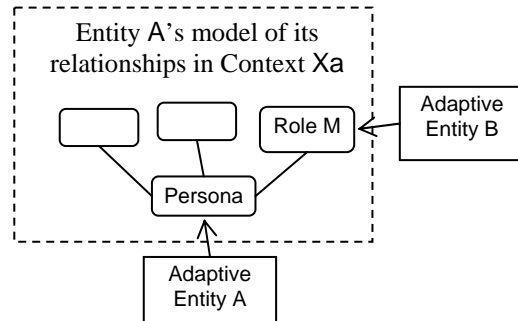
## 2. Modelling Contexts

ROAD composites represent the set of relations an individual entity has with other entities *within* a particular context. They are both a model of those relationships, and can be implemented as self-managed composites (SMCs) using the ROAD framework. In this section we briefly describe the application of ROAD models to the domain of pervasive computing. Firstly, we describe how an entity's relationships are modelled within a particular context and, secondly, how 'propositions' (correct or otherwise) about the context are represented.

### 2.1 Modelling relationships between individuals, devices, groups and services

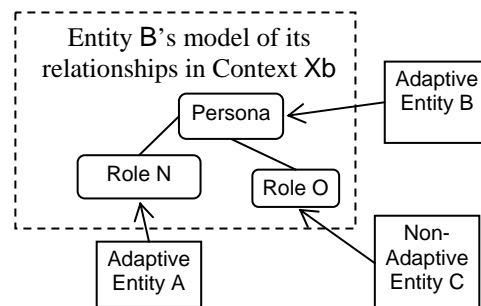
In an open dynamic system in which each interacting entity may have a different owner and interests, there may be no global or consistent representation of context to which each entity can confer. To the extent that an entity is deliberately *adaptive* it needs to maintain its own representation of its own context – that is, *its relationships and the conditions that affect those relationships*. In Figure 1,

an Entity A maintains a model of its relationships within a particular context Xa. In this model it represents itself as a role (Persona)<sup>1</sup> as well as the other entities with which it interacts in that particular context, for example, Entity B which plays Role M with respect to A.



**Figure 1. Entity A's model of its relationships in Context Xa**

An entity's model of a context is *subjective*. Figure 2 illustrates Entity B's model of its relationship with A in Context X. Note that B's view of Context X (Xb) is different from A's view. In this 'post-modernist' approach, there is no globally 'correct' view of a context<sup>2</sup>. While entities may share their perception of 'propositions' about a context with other entities, each participating entity needs to maintain its own construction of its context.



**Figure 2. Entity B's model of its relationships in Context Xb**

<sup>1</sup> An entity may have a number other contexts in which it participates and for which it maintains separate models. In each of these different contexts, these self-roles may be defined differently, that is, the entity may have different 'personas'.

<sup>2</sup> In this approach then, a CSCW systems primary function would be to facilitate the sharing on context information and, perhaps, providing sensory input to context 'propositions', rather than providing a complete global construction of a context.

It follows that each entity sees its relationships with other entities through a context whose representation it constructs. In ROAD, this constructed model is reified as an interaction structure through which all interactions are mediated. In the case where two self-managed adaptive ROAD entities (e.g. A and B in the above figures), these entities interact through the contexts they create and maintain. Entities only need to create and maintain such context *to the extent* that they need to manage and adapt their behaviour. If an entity's behaviour is invariant and it does not need to control or regulate its relationships (e.g. a resource Entity C in Figure 2), then there is no point modelling its context. In such a case, while non-adaptive entity participates in other entities contexts, it does not maintain its own context.

Let us now illustrate this approach to modelling using our motivating scenario. For example, at the open day a prospective student registers an interest in a particular course run by a Faculty. The Faculty provides interactive multi-media sessions in various rooms and to remote student devices. These rooms and devices have varying capabilities to present media in different formats. In this scenario, there are a number of services that need to be combined to provide this composed service. An identity management system needs associate the individual user and their preferences with faculty and courses. A multi-media delivery system needs to work with the identity management system to ensure correct format is streamed to the device. A location-tracking system senses whether or not the student is in the presentation room. If not, the system will attempt to deliver the presentation to the student's device rather than the video screen in the room. The faculty also wishes to record on its information system if the student has had access to the presentation.

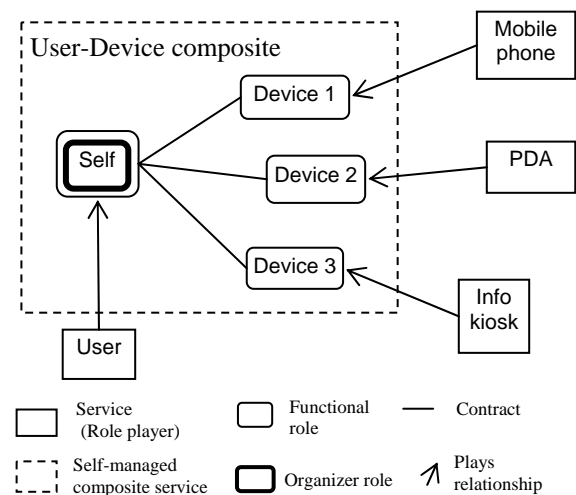
An individual attending the open day with one or more devices will want to control the interactions those devices have with other systems, say, in restricting access to personal information. Likewise, the university's system will need to control access and interactions to the various systems to which it provides and interface. In the context of our scenario, the different 'owners' in will have different set of relationships they may want to model and control. Some of the models required might include:

- the individual's view of their relationships to various devices including their preference for each device;
- the identity management system's view of the relationship between faculties, groups and individuals, and the access privileges associated with those relationships; and
- a multi-media delivery system's model of the various devices, their presentation capabilities,

the format source media, available compression services and bandwidth capabilities of available communication channels.

Figure 3 shows a schematic of a user's relationship to a number of devices with which he/she interacts. The individual will have a set of general preferences on how he/she likes to interact with devices in general (a user model). These preferences will be translated into specific contract properties that take account of the specifics of the relationship between the user role and a particular device. These specifics may relate to the device's capability (e.g. a user may have a preference for large fonts), or relate other quality attributes such the user's level of trust of the device (e.g. a user may not be happy for personal information to be stored on the device).

As the composite in Figure 3 represents the user's view of his/her relationships, the user role is not just a proxy for the user within the composite. The user role is also an *organizer* of the relationships within the composite (denoted by the bold rounded rectangle in the Figure). The organizer role is discussed in more detail in the Section 3.

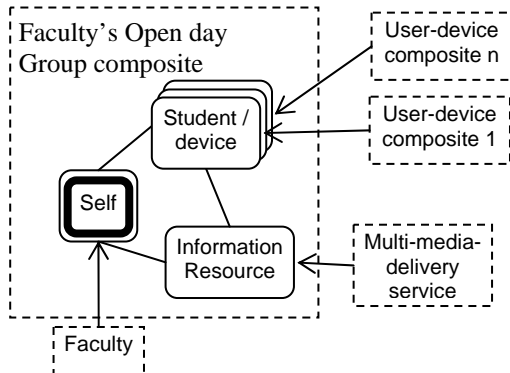


**Figure 3<sup>3</sup>. An individual's relationships to his/her device context**

Figure 4 shows another example of representation of relationships from our scenario. The Faculty creates a composition that allows information presentation to be delivered to prospective students in various locations. Note that the services playing the roles in this Faculty's composite are themselves composites. The User-device composite service that plays a student/device role in Figure 4 aggregates information

<sup>3</sup> The notation in Figures 3 and 4 is a simplified version of the notation used for ROAD concepts the semantics of which is described in [9].

about a user and devices as shown in Figure 3. The Faculty service (internal composition not shown) aggregates an Identity Management service, and the Faculty information system; and the Multi-media delivery service uses location tracking service, a compression service and information provided by the Faculty.



**Figure 4. A Faculty's composite representing relationships between individuals, group and resource services**

In a ROAD model, the contracts between roles terms define the mutual obligations of the participant roles in the given context. They define the interactions that are permissible or required by the participant roles. Contracts can also set arbitrary non-functional requirements on their roles' interactions, and how conformance to those requirements will be measured. Next, we show how context can be represented in contracts.

## 2.2 Representing context

As pointed out in the section on related work below, there are a number of approaches to modelling context. We take a distributed approach where each entity maintains its own representation of the context relevant to it. Context is relevant to an entity (i.e. needs to be taken account of by the entity) *to the extent* that it affects its relationships with other entities. From this view, *a situational context proposition is the object of a predicate condition that affects one or more interactions that a part of a social context of an entity*. An entity's store of situational context propositions can be related various types of context (e.g. computing context, user context or physical context) and can be statements about any aspects of that context (e.g. identity, time, state, etc.). These statements may be accord with the real-world or otherwise. The context for an individual is the sum of those propositions.

For example, in Figure 3 the user entity (the owner/organizer of the context representation) may

have a preference not to use their mobile cell phone in a lecture theatre. The context proposition is that the user and their phone are in the lecture theatre (this may be true or not). This proposition affects to relationship between the user and the phone because the user as owner of the context relationships has set a rule that the phone will not signal the user when there is an incoming call. In ROAD, context is represented as conditions are written as arbitrary contract terms that govern the relationships between roles played by services bound to the composition. In this case, the contract between the user role and the phone role define permissible user-phone interactions. The rule that the phone will not signal the user is written as a term of that contract. However, it is worth reiterating that these contracts are only the entity's own representation of its context. The entity on the other side of the relationship needs to conform to the terms of the contract in order for the terms of the contract to be fulfilled. Where the other entity is owned by a separate business this may require some form of external agreement (e.g. a service level agreement). In our example however, the user owns the phone, so provided the phone can conform with the contract term (e.g. it has a silent mode that can be triggered externally) the phone's behaviour will be able to be made context sensitive.

## 3. Deploying Context Models

In this section we outline how the above context models can be deployed as self-managed composites using the ROAD framework [9, 10]. The ROAD framework provides a small set of abstract classes that can be extended to create adaptive applications. These classes realize the concepts introduced in the previous section: self-managed composite, contract, functional role and organizer.

### 3.1 Context models as service composites

ROAD composites are self-managed interaction structures that not only model the relationships between roles in a particular context, but manage and control those interactions over those relationships using contract objects. These composites are deployed as services. As services they *externally* present rich descriptions of one or more interfaces with which other services can interact. They also can be deployed on any compatible Web accessible server.

*Internally*, a ROAD composite is a self-managed role structure bound together by contracts. *Contracts* perform three functions in a ROAD composite role structure: composition, interaction control, and performance monitoring. As such, ROAD contract instances are dynamic and rich connectors between

roles. A contract stores the mutual obligations of the parties bound it and monitors interactions between the parties for compliance to those requirements. In this way contracts perform a similar function to interceptors in conventional middleware.

Each composite has an organizer role that tries to ensure the relationships within the composite, and the relationships with other entities, remain viable and continue to fulfil the composite obligations. *Organisers* are autonomic managers that create and destroy roles. They also make and break the bindings between composite roles and services (service selection), and create and revoke the contracts between the roles. They can thereby create various configurations of roles and services. Organisers set performance requirements for the contracts they control, and receive performance information from those contracts. They also maintain and update the store of context propositions used to evaluate the context relevant to the interactions that flow through its contracts. The interested reader is referred to [9] for a more details description of the ROAD framework and its adaptive capabilities.

### 3.2 Acquiring and communicating context

In order to monitor compliance to contract terms, contracts can have domain-specific utility objects attached to their terms. The evaluation methods of these objects are triggered as interactions pass through the contract. This same mechanism can be used to evaluate a context relevant to an interaction. At runtime, these utility objects provide the sensing and evaluation mechanisms to determine the state of the context, evaluate that state with respect to rules stored in the contract, and execute any subsequent action (e.g. stop the phone ringing).

As described in the previous section, each entity controls the perception of its own context. Perception of context between entities can therefore vary. However, if two adaptive entities are to effectively interact within a given set of ‘objective’ context propositions, the entities may need communicate and agree on the context of the interaction. In ROAD, the organiser of composites present management interfaces over which requirements and capacity requirements can pass. It is proposed that the composite organiser use this same interface this context sharing. This management interface has not yet been implemented in the ROAD framework.

### 3.3 Adapting to changes in context

In context-aware pervasive applications, the structure and relationships with the composition may have to continually change as entities join and leave

the composition, and as changes in context affect the relationships between the entities. The ROAD framework has been designed to support adaptation resulting from both changes in requirements and changing levels of service provision. This adaptation can take the form of *regulation* (the terms of existing contracts are change in response to a mismatch between requirement and performance) or *restructuring* (where the structure of the composite is changed by adding or removing new roles or by selecting different role players). Composite organisers are provided with decision making strategies to execute such adaptation.

Likewise, in adaptive context-sensitive applications the need for adaptation is also triggered by a mismatch between monitored performance and requirements as defined in the contracts. In the case of context-sensitive applications, however, the change in requirements is triggered by a change in context as determined by rules written into the contract terms (as described in Section 2.2 above). Apart from this different source of perturbation, the strategies for adaptation are the same as for a standard service oriented application.

## 4. Related Work

Context-aware computing focuses on studying how information about people context is modelled and used. Particularly, researcher efforts have been in two areas:

- (1) understanding and modelling the context of pervasive systems,
- (2) developing architectures for context management.

### 4.1 Modelling context

Context has been defined as “information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and applications themselves” [12, p. 106]. Researchers have been investigated ways of modelling different aspects of context, such as user modelling [15, 21], location modelling (e.g., a symbolic model and a geometric model) [6], and device and sensor modelling [8], etc. In addition, different approaches to structure context information, including key value data structure [20], mark-up scheme model [7], ontology based model [15], aspect-oriented model [18], and so on. This paper does not attempt to address such semantic and formal aspects of context representation. Our approach is complementary to such concerns in the sense that we address *where* context should be represented in an

application architecture – that is, context is seen as a property that affects interactions,

In relation to user's awareness of group, previous studies from CSCW have modelled various aspects of groups and group awareness [14, 22]. Particularly, different aspects of group awareness have been examined, including conversational awareness (user awareness of communication), workspace awareness (user awareness of tasks, people, artefacts) and organisational awareness (user awareness of group/organisation structure). Again, this paper does not address the semantics of group contexts or how to present such context information to the user. Rather, our concern is to develop of meta-model that will allow group of entities and their relationships to be modelled in a software system. Such an architecture allows the relationships within the system to mirror the relationships between entities in the real-world.

## 4.2 Adaptive architectures for context-aware pervasive systems

A number of architectures have been proposed to support management of context information. Commonly, the architectures decouple the context sensing task and the pervasive application. This loosely coupled design aims to reduce the overhead of acquiring low-level data from sensors, representing it to the standardised format and using it in a useful manner. Context server can be *centralized* or *distributed*. In the centralized architecture, there is a centralized context server that provides context information to applications. Examples of the centralized architecture are shown in [12, 24]. On the other hand, in the distributed architecture context is acquired and managed at different servers, the applications need to search for context information that is useful to the applications, as shown in [5]. While this paper does not specify how context is acquired, the approach proposed here is inherently decentralised in that each entity is responsible for maintaining the representation of its own context (albeit with the option of using centralised if desired).

In addition, there have been also different approaches to developing infrastructures that adapt to dynamic environments. In particular, two of these approaches have focused on adaptive pervasive systems.

In the Aura project, Garlan et al. [13] define an adaptive framework for the construction of "task-aware" pervasive systems. Task models capture and model user goals and intent, and represent quality attributes of the services that perform those tasks. These quality goals can be conflicting. The adaptive system needs to find the optimal balance of qualities to suit the user's goals. To do this it evaluates and

consolidates multi-dimensional utility functions. The Aura project introduces an important concept to control-oriented adaptation in software; that is, control loops are needed to adapt to both the external environment (user, system context, problem domain), and the computational/network environment (bandwidth, CPU, memory etc.). The multi-dimensional nature of quality attributes is recognised, and formalisms (albeit crudely modelled) are introduced to calculate optimal utility [19].

Another middleware approach to adaptive pervasive computing is Peer-to-Peer Pervasive Computing (3PC) which consists of both a middleware (BASE) [4] and a component model (PCOM) [3]. In 3PC, an application is viewed as a tree of components. Adaptation is seen as switching of sub-trees using search and selection based on the description of component interfaces and requirements. These contracts define offered functionality (both functional and non-functional qualities), required functionality, and required resources. 3PC applications are monolithic in the sense that a single application attempts to compose passive resources and services. .

## 5. Conclusion

In this paper, we propose an adaptive architecture for context-aware pervasive systems. The proposed architecture focuses on modelling entities as well as relationships within and between entities. To support adaptation, our proposed architecture 'wraps' the entities that need to adaptive in interaction-structures that represent the entities' contexts. Each entity maintains the context(s) that is (are) relevant its own interactions but adding/modifying context proposition-action rules to the terms of its contracts. These loosely-coupled composites are self-managed in that they can alter their internal structure and relationships in response to changes in context. Externally, each composite is considered as a service composite, which has a standardized interface allowing interaction/composition with other composites.

The ROAD framework has been developed as a proof-of-concept prototype in which self-managed composites can dynamically alter their contracts and composition in response to changing requirements and component performance. However, it remains future work to instrument an application built on the framework with context-sensitive mechanisms. Interfaces for sharing and negotiating context information also need to be developed.

## 6. References

- [1] G. D. Abowd and E. D. Mynatt, "Charting past, present, and future research in ubiquitous

- computing," *ACM Transactions of Computer Human Interaction*, vol. 7(1), pp. 29-58, 2000.
- [2] G. Banavar and A. Bernstein, "Software Infrastructure and Design Challenges for Ubiquitous Computing Applications," *Communications of ACM*, vol. 45(12), pp. 92-96, 2002.
- [3] C. Becker, M. Handte, G. Schiele, and K. Rothermel, "PCOM - a component system for pervasive computing," *Proceedings of the 2<sup>nd</sup> IEEE Conference on Pervasive Computing and Communications PerCom'04*, 2004, pp. 67-76.
- [4] C. Becker, G. Schiele, H. Gubbels, and K. Rothermel, "BASE: A Micro-Broker-Based Middleware for Pervasive Computing," *Proceedings of the 1st IEEE international Conference on Pervasive Computing and Communications*, 2003, pp. 443- 451.
- [5] W. Beer, V. Christian, A. Ferscha, and L. Mehrmann, "Modeling Context-aware Behavior by Interpreted ECA Rules," *Proceedings of the International Conference on Parallel and Distributed Computing (EUROPAR'03)*, Klagenfurt, Austria, 2003, pp. 1064-1073.
- [6] M. Beigl, T. Zimmer, and C. Decker, "A Location Model for Communicating and Processing of Context," *Personal and Ubiquitous Computing*, vol. 6(5-6), pp. 341-357, 2002.
- [7] P. J. Brown, J. D. Bovey, and X. Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications*, vol. 4(5), pp. 58-64, 1997.
- [8] D. J. Carmichael, J. Kay, and B. Kummerfeld, "Consistent Modelling of Users, Devices and Sensors in a Ubiquitous Computing Environment," *User Modeling and User-Adapted Interaction*, vol. 15(3-4), pp. 197-234, 2005.
- [9] A. Colman, "Role-Oriented Adaptive Design," PhD Thesis, *Faculty of Information and Communication Technologies* Melbourne, Australia: Swinburne University of Technology, 2006.
- [10] A. Colman and J. Han, "Using Role-based Coordination to Achieve Software Adaptability," *Science of Computer Programming*, vol. 64(2), pp. 223-245, 2007.
- [11] G. B. Davis, "Anytime/Anywhere Computing and the Future of Knowledge Work," *Communications of the ACM*, vol. 45(12), pp. 67-73, 2002.
- [12] A. K. Dey, D. Salber, and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human Computer Interaction*, vol. 16(2-4), pp. 97-166, 2001.
- [13] D. Garlan, V. Poladian, B. Schmerl, and J. P. Sousa, "Task-based self-adaptation," *Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems*, 2004, pp. 54-57.
- [14] C. Gutwin and S. Greenberg, "A Descriptive Framework of Workspace Awareness for Real-Time Groupware," *Computer Supported Cooperative Work*, vol. 11(3-4), pp. 411-446, 2002.
- [15] M. Hatala and R. Wakkary, "Ontology-Based User Modeling in an Augmented Audio Reality System for Museums," *User Modeling and User-Adapted Interaction*, vol. 15(3-4), pp. 339-380, 2005.
- [16] L. M. Jessup and D. Robey, "The Relevance of Social Issues in Ubiquitous Computing Environments," *Communications of the ACM*, vol. 45(12), pp. 88-91, 2002.
- [17] K. Lyytinen and Y. Yoo, "Issues and Challenges in Ubiquitous Computing," *Communications of the ACM*, vol. 45(12), pp. 63-65, 2002.
- [18] J. Munnely, S. Fritsch, and S. Clarke, "An Aspect-Oriented Approach to the Modularisation of Context," *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications PerCom'07*, White Plains, New York, USA, 2007, pp. 114-124.
- [19] V. Poladian, J. P. Sousa, D. Garlan, and M. Shaw, "Dynamic Configuration of Resource-Aware Services," *Proceedings of the 26th International Conference on Software Engineering ICSE'04*, 2004, pp. 604-613.
- [20] B. N. Schilit, M. M. Theimer, and B. B. Welch, "Customizing mobile applications," *Proceedings of USENIX Mobile & Location-Independent Computing Symposium*, Cambridge, Massachusetts, 1993, pp. 129-138.
- [21] O. Stock, M. Zancanaro, P. Busetta, C. Callaway, A. Krüger, M. Kruppa, T. Kuflik, E. Not, and C. Rocchi, "Adaptive, intelligent presentation of information for the museum visitor in PEACH," *User Modeling and User-Adapted Interaction*, vol. 17(3), pp. 257-304, 2007.
- [22] M. H. Tran, Y. Yang, and G. K. Raikundalia, "F@: A Framework of Group Awareness in Real-time Distributed Groupware," *Proceedings of the Eighth Asia Pacific Web Conference APWeb'06*, Harbin, China, 2006, pp. 461-473.
- [23] M. Weiser, "Computer for the 21st century," *Scientific American*, vol. 256(3), pp. 94-104, 1991.
- [24] T. Winograd, "Architecture for Context," *Human-Computer Interaction*, vol. 16(2-4), pp. 401-419, 2001.