

Architecture Support for Context-aware Adaptation of Rich Sensing Smartphone Applications

Zhaozong Meng

School of Computing and Engineering, University of Huddersfield
Huddersfield HD1 3DH - UK

[e-mail: zhaozong.meng@outlook.com]

*Corresponding author: Zhaozong Meng

*Received February 18, 2017; revised May 14, 2017; accepted July 10, 2017;
published January 31, 2018*

Abstract

The performance of smartphone applications are usually constrained in user interactions due to resource limitation and it promises great opportunities to improve the performance by exploring the smartphone built-in and embedded sensing techniques. However, heterogeneity in techniques, semantic gap between sensor data and usable context, and complexity of contextual situations keep the techniques from seamless integration. Relevant studies mainly focus on feasibility demonstration of emerging sensing techniques, which rarely address both general architectures and comprehensive technical solutions. Based on a proposed functional model, this investigation provides a general architecture to deal with the dynamic context for context-aware automation and decision support. In order to take advantage of the built-in sensors to improve the performance of mobile applications, an ontology-based method is employed for context modelling, linguistic variables are used for heterogeneous context presentation, and semantic distance-based rule matching is employed to customise functions to the contextual situations. A case study on mobile application authentication is conducted with smartphone built-in hardware modules. The results demonstrate the feasibility of the proposed solutions and their effectiveness in improving operational efficiency.

Keywords: Mobile applications, context-awareness, function customisation, semantic distance, built-in sensors

1. Introduction

The smartphone today has become an indispensable information assistant for a large population. It plays an important role in people's daily routines for information retrieval, navigation, entertainment, learning, health monitoring, social activities, etc. [1-3] However, the smartphone devices are handheld with wireless connectivity, which are suitable for light-weight computing tasks. The computation, communication, and user interaction capabilities are limited due to the resource limitation. Meanwhile, in order to attract users' interest and maximise the benefit, smartphone producers continuously make progress in built-in hardware modules and a lot of novel sensing techniques are integrated into the mobile devices, such as accelerometer, brightness sensor, proximity sensor, digital compass, gyroscope, GPS, NFC, microphone, camera, etc. [1, 4] In addition, some embedded and wearable sensing techniques can be used to characterise the ambient environment and user behaviours, which can also be integrated to improve the performance of the applications [5-6]. Therefore, it promises great opportunities to take advantage of the built-in hardware modules and embedded sensing techniques to gather the contextual information related to the computing tasks to improve the performance of mobile applications.

This context-aware computing paradigm can be tracked down to the days when Mark Weiser published the paper "*The Computer for the 21st Century*" [7]. Many research articles addressed and defined context, such as [8-11]. The definition by Dey [9]: "*Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application including the user and application themselves*" is widely accepted by the recent research publications. In a context-aware computing system, the relevant contexts are utilised as implicit input to characterise the user and physical environment to provide computing service that is customised to the contextual situations. Although it is widely recognised as a promising way to use the sensor data as context variables to promote efficiency of user interaction, it still faces a lot of technical challenges such as heterogeneity in techniques [12], semantic gap between sensor data and applicable context [13], and appropriate decision making method for context-aware adaptation. Since people's expectations in software and computing service increase with the continual progress of technologies, it is significant to provide a comprehensive solution employing the novel sensing techniques to effectively improve the smart devices based computing systems.

The work of this investigation is motivated by the restriction of mobile applications due to resource limitation and the technical progress of smartphone built-in hardware modules and emerging novel sensing techniques. In this investigation, a general architecture is designed to integrate the heterogeneous sensing techniques and handle the context information for computing service provisioning toward context-aware automation and decision optimisation in the mobile applications. With this system architecture, an ontology based context model is employed to bridge the gap between the sensor data and applicable context, and a rule-based decision making strategy is used for the context-aware adaptation in mobile applications with high level supervision. To prove the feasibility of the proposed methods, a case study on mobile application user authentication is implemented. The results demonstrate the feasibility of the employed methods and show great potential of the hardware driven method for mobile applications.

The remainder paper is structured as follows: Section 2 presents a review of the related work. Section 3 and section 4 provide a general context-aware computing architecture and the

technical solutions, including context modelling method and context-aware function customisation strategy. Then, Section 5 implements the methods in a case study with the results and discussions. Finally, conclusions are drawn and future work is suggested in Section 6.

2. Background

2.1 Related Work

The performance of mobile computing systems is mainly restricted by two factors: 1) limited computation and communication power, and 2) restricted user interaction capability. The mobile devices become open owing to the various information interfaces and built-in sensors. Thus, how to take advantage of the various sensing techniques to effectively improve performance of the mobile applications becomes a common concern. The technical issues focus on the following topics: 1) feasibility study of the emerging technologies in promoting mobile applications; 2) system architecture design for context-aware computing on mobile devices; 3) how to provide users with the appropriate functions.

2.1.1 User interaction of the mobile devices

Since integrated, light, and handy mobile devices are inevitably limited in computation power, memory size, communication, display, and especially user interaction, various built-in or embedded sensing techniques are employed to compensate the user interaction of mobile devices. Therefore, Human Mobile Interaction (HMI) becomes a newly coined term for user interaction of computing systems based on mobile devices.

Hardy et al. [14] used the NFC technique on the mobile phones for interaction, which allows a user to dynamically display with their mobile devices or a large public screen in greater detail. With this method, the information on the mobile device screen can be displayed on large screens. An interaction technique is presented by Toye et al. [15] allowing users to access mobile service using their camera-phones, public information display, and visual tags. This interaction technique is mainly for the control of site-specific mobile services, and a public information display is used to overcome the limitation of mobile devices' small keypads and screens. The EyePhone system by Miluzzo et al. [16] uses the mobile front camera to track the user's eye and map its current position on the display to a function or application. This approach allows the users to launch an application by blinking at an application instead of clicking the mouse. The EyePhone consists of four steps fulfilling a task: eye detection, open eye template, eye tracking, and blink detection. Implementation on a Nokia N710 tablet has been done and results indicate that Eyephone is a promising hand-free manner to drive a mobile application. A scalable framework named SoundSense is proposed by Lu et al. [17] for sound events modelling on mobile phones. SoundSense is implemented on the Apple iPhone and is specially designed for resource limited devices. It uses a combination of supervised and unsupervised learning techniques to classify general sound and discover sound events for individual users. Implementation and evaluation of two proof-of-concept applications demonstrate that SoundSense can recognise meaningful sound events of a user's everyday life. The neural signal is used to control mobile phones for hands-free, silent, and effortless human-mobile interaction by Campbell et al. [18]. The NeuroPhone detects neural signals by mobile phone applications on the iPhone using off-the-shelf wireless electroencephalography (EEG) headsets. A case study on brain-controlled address book dialing up was conducted. The prophase implementation produced promising results for a

limited set of cases. Tung and Shin [19] introduced touch force to expand user-phone interactions, which estimated the user-applied force by using the structure-borne sound propagation. Forcephone is a very promising technology for force-sensitive applications.

Table 1. Example HMI Studies

Projects	Techniques	Platform	Purpose
Touch & interact [14]	NFC	Nokia 6131	Interaction with large dynamic display
Camera-phone [15]	Camera	Nokia 3650	Site-specific mobile services
EyePhone [16]	Camera	Nokia N710	Hand-free manner to drive mobile application
SoundSense [17]	Microphone	iPhone	Discover sound events for individual users
NeuroPhone [18]	EEG headset	iPhone	Brain-controlled operations
ForcePhone [19]	Speaker, microphone, and touch screen	Galaxy Note 4	For use in force-sensitive applications

From the above investigations, various sensing techniques are promising to overcoming the limitation of the mobile devices in user interaction. However, most of the methods are still in lab experiment stage which are not mature for practical use. The probable reason is that sensor modules are susceptible to environmental interferences and they may also affect usability and efficiency of mobile applications. The interaction methods need to be reliable, robust, and lightweight to guarantee the performance of the applications.

2.1.2 Architecture support for context-aware mobile computing

The proper architectural support is an indispensable foundation for the gathering, modelling, storing, and distributing of context data. In order to facilitate mobile-based context-aware application development, a number of design strategies and frameworks are created accordingly focusing on either frontend context acquisition or backend data management and service customisation.

Early in 2001, a noticeable example framework named Context Toolkit was developed [20]. Context Toolkit is a basic architecture for platform-independent supply of context information. The Context Toolkit can facilitate the basic levels of context-aware mobile application development, which is suitable for IF-THEN logic-based computing services. A Context Broker Architecture (CoBrA) is proposed by Chen [21] to reduce difficulty and cost in building context-aware systems. In CoBrA, Web Ontology Language (OWL) is employed for context modelling and privacy policies, Java Expert System Shell (JESS) is used to build a hybrid reasoning mechanism. Chen [22] presents a context fusion framework named Solar focusing on fusion of the distributed context data in dynamic computational environment. It is built with a scalable and self-organised service, which allows applications to select distributed data sources and compose them with customised data-fusion operators. This system has taken into account high-level understanding of its execution context, buffer overflow, environment dynamics, and some common failures, and provided corresponding solutions. A Service Oriented Context-Aware Middleware (SOCAM) architecture is proposed by Gu et al. [23] for the rapid prototyping of context-aware services, which consists of five independent service components: context provider, context interpreter, context database, context-aware services, and service locating to efficiently support acquiring, discovering, interpreting, and assessing various contexts to build context-aware systems.

A flexible, scalable, and reasoning oriented framework ECORA (Extensible Context Oriented Reasoning Architecture) is presented by Padovitz et al. [24] for context-aware computing based on a functional component library implementing concepts and algorithms of

a context space approach. The ECORA is designed focusing on the reasoning of the uncertain context considering the issues of heterogeneity, scalability, communication, and usability. Therefore, it provides powerful reasoning capability, and flexibility and robustness as well. A context-aware middleware infrastructure for pervasive healthcare named Context-Aware Middleware for Pervasive Home (CAMP) is presented by Pung et al. [2], which allows timely and accurate delivery of health or medical information among the patients, doctors, and healthcare workers through widespread deployment of wireless sensors and mobile devices. The main strength of the CAMP is its capability to support a context-aware service running over multiple physical spaces.

A robust mobile SOA framework is proposed by Shrestha [25] for building and operating lightweight and flexible context-aware mobile applications (CAMA). The framework supports different attributes of context information and dynamic integration of context in real-time. Tesoriero et al. [26] defined a Model Driven Architecture (MDA) – CAUCE, a model consisting of three layers for context-aware application development: the first layer defines how to build the task, space, and social view of the system; the second layer defines how to build the referential space, the information flow, and the entity context; and the third layer defines the deployment environment according to the second layer. In order to clarify the macro-components and their interaction of context distribution, Bellavista et al. [27] presents a data-centric architecture model consisting of three principal factors: context data source, context data sink, and context distribution function. The context distribution function contains three main facilities organised in two horizontal layers: (1) context data management and context data delivery, and (2) a cross-layer facility to support runtime adaptation. This architecture model is clearly defined for context description and distribution. In addition, Moazzami et al. [28] present ORBIT, a smartphone-based platform for data-intensive embedded sensing applications, a tiered architecture to suit rich processing, multi-modal sensing, and versatile networking capabilities of smartphone applications.

From the above discussion, it is not difficult to find that dispersion of context source and heterogeneity in techniques are common problems facing architecture design of context-aware applications. On the other hand, the system architectures lack further exploration of the history context for high-level supervision. Another problem facing context-aware system architecture design is the handling of context data of different semantic levels. In addition, the awareness of quality of context and potential use of history context also needs architecture support in context-aware application development.

2.1.3 Decision making for context-aware adaptation

With the applicable context data, the system can then provide users with the functions that are customised to the contextual situations. As the ultimate goal of context-aware systems is to promote the efficiency and usability of the applications and provide the user with expected computing services, the method utilising context data to create service customised promise great opportunities to the mobile computing systems. This sub-section will provide a brief discussion of these theories and methods in dealing with context-aware adaptation in the mobile applications.

Traditional context-aware mobile systems are based largely on IF-THEN logic [29] and their capability in dealing with complex contextual situations is limited. Thereby, some mathematical models are presented accordingly. As the IF-THEN rules are independent of each other, the context change must be specified for each action for context-aware application. With the emergence of rich sensing devices, the increase in quantity of context and the complexity of its relationship requires more powerful approaches dealing with the new

situations. For the new context-aware computing systems, some theories are introduced in this area for context-aware service decision making, such as Fuzzy Multi-Attribute Decision Making (FMADM) [30], Multi-facet Item based method [29, 31], Multi-Attribute Utility Theory (MAUT) [32], Adaptive Neuro-Fuzzy Inference System (ANFIS) [33], Rough-Fuzzy method [34], and probability-based model [35]. These mathematical models are more powerful in handling the diversity of context data in for different use scenarios.

Although much progress has been made, there are remaining problems deserve further investigations. Firstly, it is still a challenge to manage the complicated structure of context data and sophisticated relationship between the entities in context-aware systems. Secondly, the existing methods mainly focus on function adaptation with contexts of interest, the following aspects are not sufficiently addressed: (1) how to unobtrusively determine the context items of interest; (2) how the context items contribute the decision making in context-aware adaptation; and (3) whether the methods employed are lightweight enough for resource limited mobile platforms.

2.2 Problems Identified

From the above discussion, it is evident that the context-aware computing paradigm has already been recognised as an important approach to improve the performance of resource constrained computing scenarios. However, since the context-aware systems may involve techniques from low-level electronics or sensor acquisition to high level data modelling and processing, it faces plenty technical challenges to effectively take advantage of the sensing techniques. The principal technical challenges identified in this investigation are summarised as follows:

- (1) It lacks general architecture to integrate heterogeneous sensing techniques and take advantage of context data
- (2) There is a semantic gap between sensor data and context data that is applicable for the applications
- (3) The complicated relationship between the entities and complex structure of context data for context-aware adaptation

This investigation will focus on the above concerns and provide the approaches to effectively make use of the sensing modules and provide functions customised to the contextual situations, including the architecture support, context modelling, and decision making strategy for context-aware adaptation. A case study on mobile application user authentication involving the built-in hardware modules is implemented to demonstrate the feasibility of the proposed methods.

3. A Hierarchical Context-aware Computing Architecture

To effectively utilise the dynamic context in the physical environment, the sensor enabled context-aware system is described with a functional model to show the role that context plays in user interaction of the applications. And then, a hierarchical system architecture consists of physical layer, context handling layer, context-aware service layer, and knowledge discovery layer is designed to guide the context-aware application development.

3.1 Fundamentals and Model

Compared with the traditional computing paradigm, a significant strength of the context-aware system is its capability in unobtrusively gathering the contextual information as implicit input to customise the applications. Thus, the essential function of the context-aware

mobile application is its capability to gain users' purpose and proactively make a decision to provide the appropriate functions accordingly. The iHCI model by Schmidt [36] has clarified the concept of implicit human computer interaction. The interaction model of traditional computing paradigm and context-aware system paradigm can be described with Fig. 1 and Fig. 2.

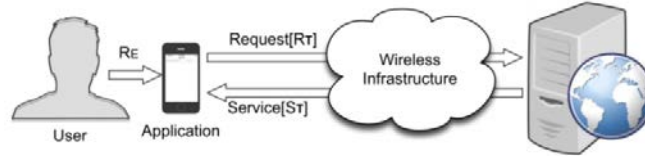


Fig. 1. Traditional Computing Paradigm

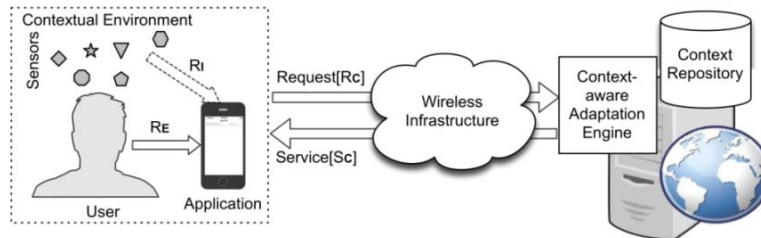


Fig. 2. Context-aware Computing Paradigm

In Fig. 1, the traditional computing systems provide the computing service S_T according to user's explicit request R_E , which can be obtained by users' operation with keyboard, mouse, or touch screen.

$$S_T = F(R_T) = F(R_E) \quad (1)$$

In contrast, as shown in Fig. 2, the context-aware systems provide context-aware computing service S_C according to users' explicit request R_E and context information C related to the computing tasks.

$$S_C = F(R_C) = F(R_E, C) \quad (2)$$

where $C = \{C_w, C_d, C_n, C_e, C_h, \dots\}$, and $C_w, C_d, C_n, C_e, C_h, \dots$ are the contextual situation of computing tasks such as user context, device context, network context, environment context, and history context. With the implicit input C , the computing function S_c is customised to the contextual environment. Therefore, the operational efficiency and user experience is promoted if the context information C is obtained with sensors in an unobtrusive way.

Since most of the smartphone mobile devices ship various built-in sensing techniques, this creates great chances to take advantage of the technologies to reduce human effort while providing users with the appropriate data or functions. In the development of context-aware computing systems, the context information plays a role that is as important as users' explicit input. The sensed context is some underlying information about the users which can be explored to characterise user's requests, such as location, ambient temperature and brightness, devices used, and user preferences. Admittedly, the handling of the context information is a critical task in the design and implementation of context-aware systems. Therefore, a hierarchical system architecture for context gathering, context management, context distribution and service adaptation is proposed for context-aware system development.

3.2 A Hierarchical Context-aware System Architecture - Modules and Components

A distinctive feature of context-aware computing system is that it involves techniques of several different areas, such as sensor network, embedded electronics, wireless communication, and web service, etc. Thus, in order to efficiently integrate the context information into the computation, the data and techniques of the system architecture should be loose coupling for the sake of flexibility and reusability. In this investigation, a hierarchical system architecture is proposed from the standpoint of context information handling flow of the system as shown in Fig. 3. The system architecture is divided into four layers: physical layer, context handling layer, context-aware service layer, and knowledge discovery layer.

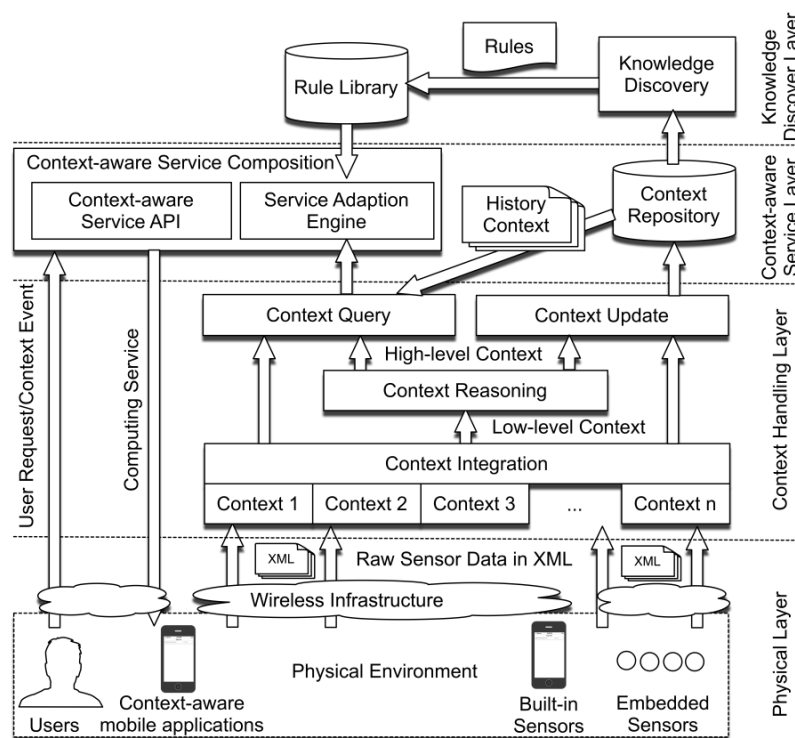


Fig. 3. Diagram of Context-aware Computing Architecture

In contrast to the systems that are based on specific middleware technologies such as DPWS [37, 38] and OSGi [39, 40] the proposed system architecture gives a clearer view of the acquisition, distribution, and utilisation of context data. The knowledge discovery layer utilising history data of context information and user interactions to discover new rules to supervise the decision making is a highlight of this architecture. Then, a brief introduction to each layer is given as follows:

(1) *Physical Layer (PHL)* - The physical layer behaves as an interface between physical world and the computing system. It consists of the low-level hardware electronics perceiving the context and executing the customised functions. Entities in this layer are human, environment, sensors, and devices. In this layer, some embedded electronics are required to convert the logic, do some pre-processing, and constitute the formalised data. Devices like Analogue to Digital Converter (ADC), Micro-Controller Unit (MCU), and wireless adaptors are needed to convert the physical variables, collect sensor data, and transfer data respectively.

(2) *Context Handling Layer (CHL)* - The context handling layer is responsible for the effective acquisition and sharing of context data. This layer behaves as an intermediate layer which connects the physical layer with the context service layer. The functional modules in this layer may include: context integration, context update, and context distribution. Due to the heterogeneity of the techniques and devices used in the physical layer for acquisition and communication, efficiency and interoperability of the modules in this layer may largely affect the performance of the system.

(3) *Context-aware Service Layer (CSL)* - The context-aware service layer achieves the main function of the composition of functions customised to the dynamic context. It makes use of the context data in either the service end or the mobile client end, and provides the service with the implicit request. Thus, it reduces the user interaction and increases the degree of automation of an application. The functional modules in this layer are context-aware APIs, service adaptation engine, context repository, and client-end application. This layer handles the generation of context-aware function with the context information according to adaptation rules in the rule library.

(4) *Knowledge Discovery Layer (KDL)* - The knowledge discovery layer consists of modules for high-level supervision of the context-aware adaptation. It employs the Knowledge Discovery in Database (KDD) techniques to extract new rules or dynamically revise the parameters of the existing rules for context-aware adaptation. These rules created automatically are used to supervise the context-aware adaptation. This layer enables the system to be able to derive high-level rules and knowledge for intelligent optimisation. Thus, it can potentially become smarter with the learning capability of the system.

The standard cross-platform protocols such as TCP and HTTP, and light-weight data wrapping techniques, such as XML, JSON are used to transfer context information between the devices of different layers. These cross-platform and light-weight techniques can effectively deal with the heterogeneity in technologies. The strength of the hierarchical system architecture is its loose coupling structure between the functional modules. This system architecture is convenient to extend and maintain because of the explicit relationship of modules in the structure. The explicit structure also simplifies the context management. Since this investigation is meant to make full use of context information to enhance the usability and efficiency of the resource limited mobile device based applications, the efficiency of the architecture in context handling and context-aware adaptation are significant to this research. The method modelling the context and creating the context-aware adaptation decisions with context data and rules are discussed in details in the following section.

4. Context Modelling and Context-aware Adaptation Decision Making

Since the context data is heterogeneous in nature and difficult to handle in computation, only with the system architecture is not enough to prototype the context-aware mobile applications. This section gives the key enabling technologies to provide users' expected functions in the mobile device based applications with the hierarchical architecture given in section 3 –context modelling and context-aware adaptation decision making.

4.1 Context Modelling

The context model organizes the context data logically with respect to the diverse applications according to its attributes. This section describes how the context items as physical environments or user profile are presented as a computable model in the context-aware mobile systems.

4.1.1 Context modelling for context-aware mobile applications

For the implementation of the context-aware system, structured context information should be provided to adapt the functions or usability features to the user's context [41]. It puts forward requirements for the context model of being able to handle application adaption, resource awareness, mobile service, and user interface adaptation. It is evident that the update of sensor perceived context data and the utilisation of context by relevant context-aware applications are based on operation of context data in context repository normalised with a context model.

The context model transforms the variables describing the physical world to the form of high level data model in computing systems that supports the distribution of context information. It is evident that the context model affects the efficiency of context updating, context reasoning, and context-aware adaptation. Thus, the context modelling should be based on the in-depth analysis of attributes and the inter-relationship of context items and its flexibility in context presentation and context distribution. In this section, the context modelling approach is presented in order to conduct efficient collaboration of the low-layer context acquisition, mid-layer context management, and high-layer context based adaptation.

4.1.2 Classification and modelling of context data

Peer investigations on context modelling favour the ontology based context modelling in the several context modelling approaches [13]. Compared with the other alternatives, ontology-based models of context are more expressive and computable especially for complex context domain. It provides formal semantics for a context domain about the objects, relationships and domain constraints, which supports the sharing and integration of structured context information [42]. Hence, ontology based modelling is a good candidate to express meanings of a context domain in a pervasive environment.

Usually, a user request or computing task of some applications may not be related to so many context items. However, a holistic context model that incorporates the relevant context items covering most of the use scenarios and for different application domains is really in need for general use. In this investigation, the context is categorised and modelled considering both the user requests and application domains. The typical attributes indicating the performance of the context model may be the generality, flexibility, and interoperability. That is to say, the context model: 1) can satisfy most of the use scenarios, 2) is easy to modify and cut, and 3) easy to represent for heterogeneous platforms. This investigation regards the context as the information that is used to characterise the relevant objects and entities that affect the computing tasks. To this end, this investigation categorises context data into the following dimensions:

(1) *User context (UC)* – The user preference, user activities, user emotion, and user social context that affect users' intention in the applications.

(2) *Computation context (CC)* - The device hardware and software platform that is used to request or execute the context-aware functions.

(3) *Network context (NC)* - The attributes of the wireless connection for the mobile devices that affects the performance of applications.

(4) *Environment context (EC)* - The ambient environment that the computing task is executed.

(5) *Location context (LC)* - The position where the context-aware computing task is executed.

(6) *Time context (TC)* - At what time the context-aware computing task is executed.

(7) *History context (HC)* - Existing context information in the context repository of the system.

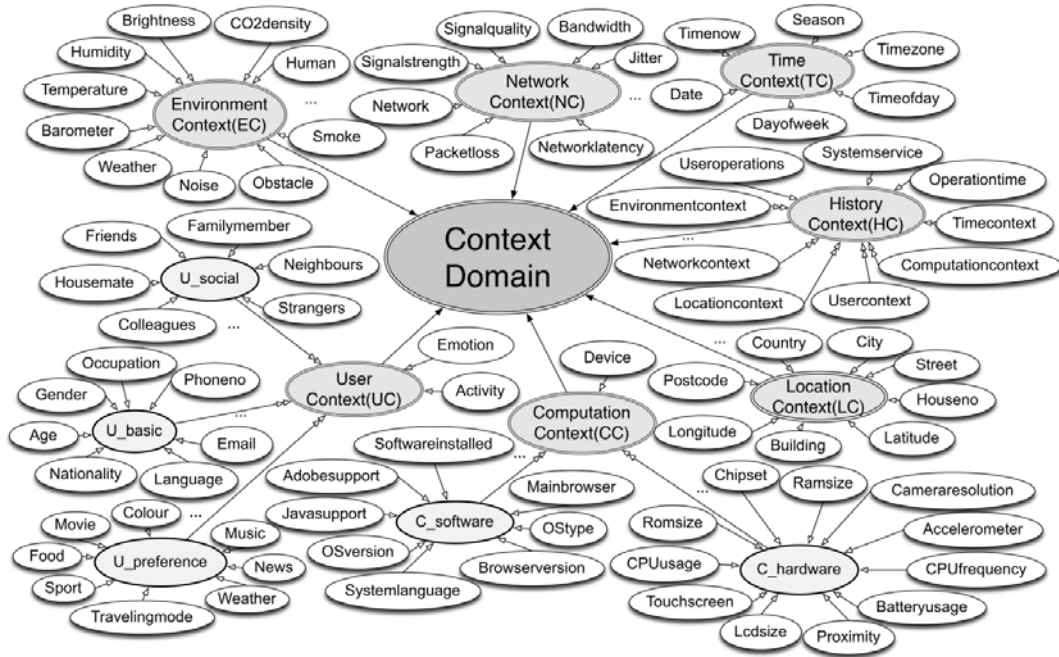


Fig. 4. Graphical Representation of Context Model Centre Ontology

The graphical representation of the context model centre ontology is given as shown in Fig. 4. The above seven context classes constitute the high-level ontology of the model, and context source modules such as physical sensors, soft sensors, and the user interaction modules that provide raw context data constitute the low-level ontology. Each of the above classes consists of the sub-classes that are considered to be critical in the context-aware service. The relationship between the classes and sub-classes is built based on the concepts and practical facts. For different use case, the context model in Fig. 4 can be tailored to fit the application scenarios according to the domain knowledge of specific applications.

4.2 Semantic Distance based Rule Matching for Decision Making

In the context-aware service layer, the context data are complex in structure and of different data types. In this investigation, the semantic distance is used for the decision making of context-aware adaptation. The rule matching method compares the current context and context-aware adaptation rules and makes decisions according to the semantic distance between them.

Since the context can be digits or linguistic text. The fuzzy theory is used to convert the digits into linguistic variables to fit into the context model as the method described by Krause [43]. For non-numeric, the difference between the values can be described with semantic distance, and researchers have already proposed many different methods measuring the semantic distance [44]. In this investigation, the Generalised Cosine-Similarity Measure (GCSM) by Ganesan et al. [45] is employed to compute the semantic distance between context sets, which is defined as follows:

Definition 1 Lowest Common Ancestor (LCA) LCA denotes the common ancestor of the maximum depth of two concepts in the ontology.

Definition 2 Generalised Cosine-Similarity Measure If c_1 and c_2 are two concepts in a tree-hierarchy of indexing terms and $depth(c_1)$ and $depth(c_2)$ are their depth in the hierarchy,

the GCSM similarity between them is:

$$GCSM(c_1, c_2) = \frac{2 \times depth(LCA(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{3}$$

For non-numeric data, (3) can be used to calculate the distance of context values which is between 0 and 1 ($0 < GCSM(c_1, c_2) < 1$). Meanwhile, the value of $GCSM(c_1, c_2)$ denotes the approximation of c_1 and c_2 . Then,

$$dist(c_1, c_2) = 1 - GCSM(c_1, c_2) \tag{4}$$

can be used to present the semantic distance between concepts c_1 and c_2 . Therefore, the formula calculating the distance of a context vector and a rule vector is:

$$dist(V, R) = \sum_{i=1}^n w_i \cdot dist(v_i, r_i) \tag{5}$$

where w_i is the weight of context item c_i , v_i and r_i are the elements of context vector V and rule vector R on attribute c_i , and $dist(V, R)$ is the distance between vector V and vector R .

With the semantic distance based rule matching method GCSM to deal with the tree-hierarchy of indexing terms in the ontology model, the heterogeneous context data can be treated with rule matching to derive the appropriate functions.

5. Mobile Application User Authentication – A Case Study

In this section, a case study on mobile application user authentication is designed and implemented for a European Commission Funded project - Mobile Exam System (MES), which employs university students' mobile devices for classroom examination. In addition to touch screen manual input, this application allows students to login by scanning the barcodes or NFC tags on their student ID cards. With the relevant context such as student ID card barcode, NFC tag ID, exam time and exam location, a context-aware decision system is implemented to improve the user interaction of MES user authentication with the proposed solutions.

5.1 Design and Implementation

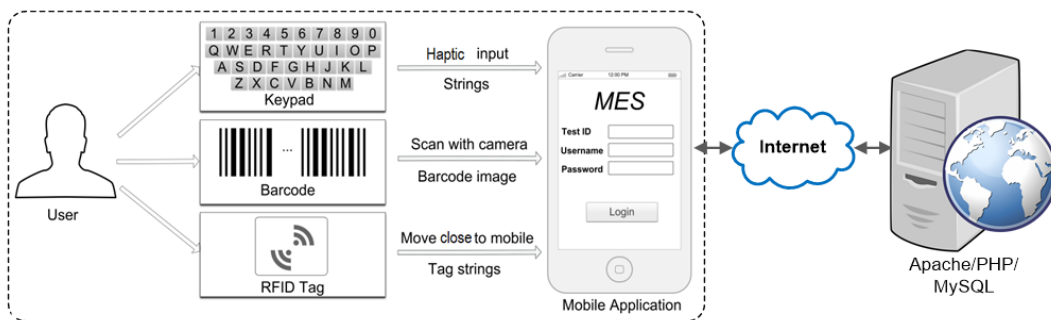


Fig. 5. User Interactions in MES Authentication

The diagram in Fig. 5 describes the design of the MES application system. It includes the MES mobile application and service application, which are designed and implemented with selected platforms and technologies. The mobile applications are developed for the

mainstream operating systems iOS, Android, Windows Phone, and Symbian. The devices used for test are listed in [Table 2](#). The Apache/PHP/MySQL is selected for the fast prototyping of database and server-end applications.

Table 2. Mobile Devices Used

Devices	OS Version	Camera (Pixels)	Screen Size(Inches)
Apple iPhone 4S	iOS 5.0	5M	3.5
Samsung Nexus S	Android 4.1.1	5M	4.65
HTC HD2	Windows Phone 7.1	5M	4.3
Nokia N8	Symbian^3	12M	3.5

5.1.1 Context definition and context modelling

For traditional mobile applications, the user account (username and password) is used for user authentication. In order to guarantee the security of the system and keep malicious access from the system, some other attributes about the users and device that are relevant to the exam can be involved as context information to assist the authentication. In the MES application, the following information are regarded as the context items: ID card barcode, NFC tag ID, IMEI (International Mobile Equipment Identity) number, phone number, user location, and exam time. Therefore, the context model that is tailored to the MES authentication use case is given in [Fig. 6](#). All the context information involved can be provided with the built-in modules of the mobile device or by users' manual operations.

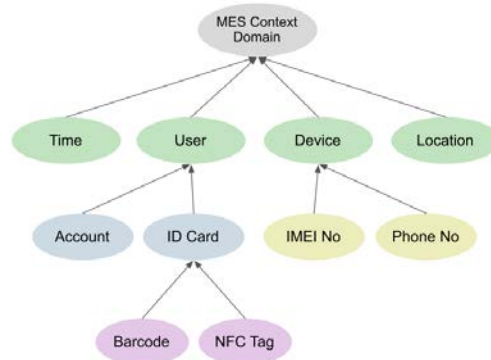


Fig. 6. Context Model for MES User Authentication

5.1.2 Context acquisition

Normally, the context data should be obtained by the mobile devices in an unobtrusive way, such as IMEI, phone number, current time, and user location. The devices read the corresponding system module to access the data in real-time.

For barcode scanning and NFC tag recognition, the users need to align the barcode to camera vision or move ID card close to the phone. For barcode scanning, the open source library ZXing is used, and the decoding algorithm for student ID card barcode format 'Telepen' is developed and added to the library for different mobile platforms. For NFC tag recognition, some mobile software development kits allows the functions in software development.

5.1.3 Rules definition

Before the rules, the condition attributes and decision attributes should be defined first. The condition attributes vector A of the MES system includes barcode ($C_{barcode}$), NFC tag ID (C_{nfctag}), IMEI (C_{imei}), time (C_{time}), and location ($C_{location}$), which can be denoted as below:

$$A = \{C_{barcode}, C_{nfctag}, C_{imei}, C_{time}, C_{location}\},$$

The states of the context items are True, when matches users' data accessed from database, or False, when it does not match.

The decision attribute vector D for user authentication are authorised to login ($D_{authorised}$), password required ($D_{password}$), and login refused ($D_{refused}$), which is presented as below:

$$D = \{D_{authorised}, D_{password}, D_{refused}\},$$

After the decision making, one of the above attribute is selected and executed with the mobile applications.

In the context-aware service layer, the decision making of context-aware adaptation is accomplished by calculating the semantic distance between context samples and the rules. The predefined rules for the MES context-based authentication decision making are given in **Table 3**. Since IMEI and phone number are usually combined, only IMEI is used in this system.

Table 3. Pre-defined Rules

ID	RULES (Conditions and Decisions)
Rule1	$(C_{barcode} \text{ is True or } C_{nfctag} \text{ is True}) \wedge (C_{imei} \text{ is True}) \wedge (C_{time} \text{ is True}) \wedge (C_{location} \text{ is True}) \rightarrow D_{authorised}$
Rule2	$(C_{barcode} \text{ is False or } C_{nfctag} \text{ is False}) \wedge (C_{imei} \text{ is X}) \wedge (C_{time} \text{ is X}) \wedge (C_{location} \text{ is X}) \rightarrow D_{authorised}$
Rule3	$(C_{barcode} \text{ is True or } C_{nfctag} \text{ is True}) \wedge (C_{imei} \text{ is False}) \wedge (C_{time} \text{ is True}) \wedge (C_{location} \text{ is True}) \rightarrow D_{password}$
Rule4	$(C_{barcode} \text{ is X or } C_{nfctag} \text{ is X}) \wedge (C_{imei} \text{ is X}) \wedge (C_{time} \text{ is False}) \wedge (C_{location} \text{ is X}) \rightarrow D_{authorised}$
Rule5	$(C_{barcode} \text{ is X or } C_{nfctag} \text{ is X}) \wedge (C_{imei} \text{ is X}) \wedge (C_{time} \text{ is X}) \wedge (C_{location} \text{ is False}) \rightarrow D_{authorised}$

(*X in Table 3 denotes any value)

5.1.4 Decision making for function customisation

The user authentication decision making is completed according to the context model in **Fig. 6** and rules defined in **Table 3** using the method presented in section 4.2. In order to demonstrate how decision making works in MES authentication, a set of context data in database of the prototype system is selected as the current context:

$$CV_i = \{\text{True}, X, \text{False}, \text{True}, \text{True}\}$$

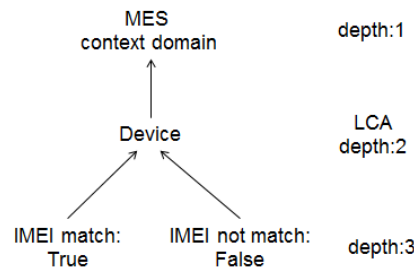


Fig. 7. Tree Hierarchy Structure of IMEI in the MES Context Domain

The $dist(CV_i, Rule_i)$ denotes the semantic distance between the context sample CV_i and rule $Rule_i$. According to **Fig. 6**, the barcode in the tree hierarchy indexing structure in the context

model can be represented with the diagrams in **Fig. 7**.

Then, according to rule matching presented in section 4.2, the normalised semantic distance between the context sample CV_i and the rules in **Table 3** can be calculated.

For example, in the sample data $CV_i[C_{imei}] = \text{False}$. According to the tree hierarchy structure, $\text{depth}(\text{LCA}(\text{IMEI Match:True}, \text{IMEI Not Match:False})) = 2$, $\text{depth}(\text{IMEI Match:True}) = 3$, $\text{depth}(\text{IMEI Not Match:False}) = 3$. According to formula (3),

$$\text{dist}(CV_i[C_{imei}], \text{Rule1}[C_{imei}]) = 1 - 2*2/(3+3) = 0.33$$

With the same method:

$$\text{dist}(CV_i[C_{barcode}], \text{Rule1}[C_{barcode}]) = 1 - 2*4/(4+4) = 0$$

Then, according to formula (5),

$$\text{dist}(CV_i, \text{Rule1}) = 0.25*(\text{dist}(CV_i[C_{barcode}], \text{Rule1}[C_{barcode}]) + \text{dist}(CV_i[C_{imei}], \text{Rule1}[C_{imei}]) + \text{dist}(CV_i[C_{time}], \text{Rule1}[C_{time}]) + \text{dist}(CV_i[C_{location}], \text{Rule1}[C_{location}])) = 0.0825$$

where 0.25 is the weight of the four relevant context items. So, the results of the rule matching are given as follows:

- (1) $\text{dist}(CV_i, \text{Rule1}) = 0.25*(0+0.33+0+0)=0.0825$
- (2) $\text{dist}(CV_i, \text{Rule2}) = 1*0.25=0.25$
- (3) $\text{dist}(CV_i, \text{Rule3}) = 0.25*(0+0+0+0)=0$
- (4) $\text{dist}(CV_i, \text{Rule4}) = 1*0.5=0.5$
- (5) $\text{dist}(CV_i, \text{Rule5}) = 1*0.5=0.5$

In this system, the weight of the attributes in the rules are equal, the weight is $1/n$ when there are n attributes in a rule. It is clear that the minimum semantic distance between CV_i and the pre-defined rules is $\text{dist}(CV_i, \text{Rule3})$. Thus, *Rule3* is considered to be the appropriate rule for user authentication, and the decision ' $D_{password}$ ' is executed, namely display user account automatically and request password to confirm.

5.2 Experiments and Results

With the system and methods presented in section 5.1, experiments are carried out and the results are obtained. The experiments are expected to determine the strengths of the hardware involved method compared with the traditional touch screen manual input method. In order to guarantee the results are objective and convincing, the experiments are performed as follows:

- (1) The same participant tests the system on the different devices.
- (2) Test of each method is repeated 20 times and the average is used as the final value.
- (3) The timing of login operation is from opening of software to login result is given.
- (4) The experiments should be done in the same physical environment.

With the above principles, two experiments are carried out: (1) testing the 3 user interaction methods on the same platform; (2) test the same methods on the different platforms. Since the NFC module is not widely shipped on the mobile platforms, only the touch screen input and barcode scanning methods are tested in experiment (2). The length of user name and password are set from 8 to 20 letters. The operation time of touch screen input, barcode scanning, and NFC recognition are tested and results are as shown in **Fig. 8**. In order to further analyse the experimental data, the standard deviation (STDEV) is calculated and presented in **Fig. 9**.

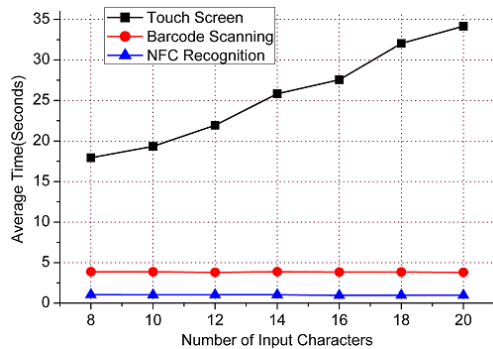


Fig. 8. Mean value of operation time with the 3 methods on Nexus S

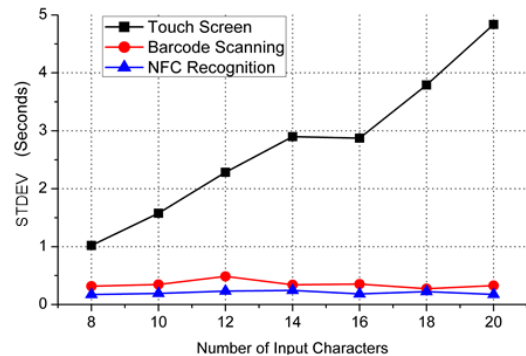


Fig. 9. STDEV of operation time with the 3 methods on Nexus S

From Fig. 8, it is easy to find that the operation time of touch screen is much longer and it increases with the length of information to input, while the barcode scanning and NFC recognition methods are much faster and they keep constant. On the other hand, the operation of touch screen input method is boring and error prone. By the barcode scanning and NFC recognition methods, users just need to put the student card in the camera vision and put the student card close to the smartphone. With the assistant of built-in camera and NFC module, the user operation is largely simplified and efficiency and accurate rate are also promoted. From Fig. 9, it is evident that the standard deviation of operation time with touch screen input is much larger than the barcode scanning and NFC recognition. The STDEV values of barcode scanning operation time vary slightly, and STDEV values of NFC recognition keep almost constant. That is to say, the performance of touch screen input method is not very stable, and the NFC recognition method is the best in stability.

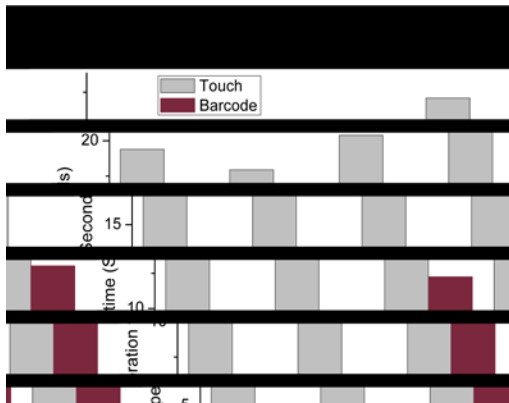


Fig. 10. Operation time of touch screen and barcode method on different platforms

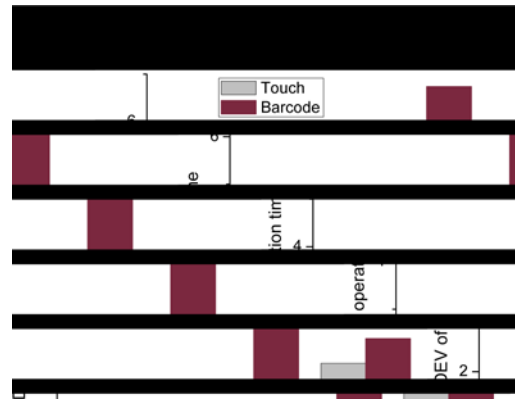


Fig. 11. Operation time STDEV of touch screen and barcode method on different platforms

The operation time and STDEV of touch screen input method and barcode scanning method on different platforms are given in Fig. 10 and Fig. 11. From Fig. 10, we find the performance of the user operation differs on the different mobile platforms. The touch screen input methods vary slight between the different platforms. The operations on the device with larger screen are faster and operations on the devices on the smaller screens are slower. On the other hand, the operation time of barcode scanning also varies, which is faster on iPhone 4S and Nexus S and slower on HD2 and N8. The operation time STDEV of HD2 given in Fig. 11

is significantly greater than the other platforms, and those of iPhone 4S and Nexus S are very small.

5.3 Discussion

In the case study, the user interaction methods using the built-in camera and NFC module have effectively promoted the operational efficiency of the mobile application. The system architecture and methods proposed in this investigation can be considered to be feasible. However, it also reveals some underlying problems in applying the built-in hardware modules to enhance the performance of software applications. Through the experiments, it is found that the hardware based methods may be easily influenced by the ambient environment. For example, the camera based barcode scanning is vulnerable to the ambient brightness and light reflection of the student ID card surface. It is also found that the heterogeneity in hardware system influences the performance of the application to a large extent. The devices with efficient camera (iPhone 4S and Nexus S) are faster in barcode scanning, and the performance is much more stable. But the devices with camera that are not efficient enough are slower and the performance is not stable. Since the performance of the applications is influenced by the mobile platforms due to the heterogeneity of the hardware modules, the quality of the hardware modules should be evaluated when they are used. In addition, the effect on performance for regular working of smartphones due to active use of sensors for context sensing needs to be investigated as well.

The hierarchical system architecture divides the techniques and functions into four layers which are flexible and loose coupling. It simplifies the construction and maintenance of the system. Since the knowledge discovery layer enables the system to determine the context items that make greater contribution of some computing tasks and generates new rules, the system is self-motivated to provide the appropriate decision for different people and different use scenarios. The ontology based context modelling method bridges the semantic gap between raw sensor data and usable context. However, it needs further investigation to accurately characterise the users and physical environment with the various context items using domain knowledge for specific applications. In addition, the complexity of ontological context model may also challenge the resource limited devices. The context models are expected to be tailored to the application scenarios for lightweight computation. The context-aware adaptation decision making method is competent in dealing with the heterogeneous context data in context-aware computing systems. Since all the context data and decisions are recorded in database, some data analysis methods such as Bayesian Network, Artificial Neural Network (ANN), etc. can be potentially applied to discover underlying rules with the history data. The efficient and effective rule generation methods will be promising topics in the future investigations.

Compared with the context-aware system architectures mentioned in section 2.1.1, this investigation provides a systematic and comprehensive solution for context handling and context-aware adaptation. It firstly gives a clear classification of context handling method according to the attributes of context at different levels. From electronics signal level at physical layer to information level at context handling layer, data level at service customisation layer, and knowledge at knowledge discovery layer. Then, the context modelling bridges the physical environment and contextual variables and gives a computable presentation of context items. And finally, the semantic distance based decision making offers solutions to pursue the suitable decisions for specific contextual situations.

In the case study, by integrating the context as implicit inputs to the mobile applications, the user operations are simplified and the operation efficiency are therefore improved.

6. Conclusion and Future Work

This investigation provides a hierarchical architecture with the relevant methods utilising the context information to customise the functions of smartphone applications. From the results of the case study, it can be regarded as a feasible strategy to employ the hardware modules and sensing techniques to effectively improve the performance the applications on the resource limited smart devices.

(1) The proposed architecture is general and flexible to integrate the heterogeneous technologies to gather and distribute the context data. In addition, it highlights the further use of context by employing history context to generate rules to supervise the function customisation.

(2) The ontology based context modelling is expressive and computable to bridge the gap between raw sensor data and usable context items. It is open and extensible for the prototyping of complicated context-aware applications

(3) The decision making strategy is competent in dealing with normalised context data of different data types and structures, which is well suited for complex contextual situations.

Although various new techniques are continuously emerging to strengthen the technical solutions, the challenges facing context-aware computing systems are still the design principles and fundamental methods and theories. To put it briefly, the focal objectives of the future works are:

(1) Methods to evaluate quality of context and to deal with the vagueness and uncertainty of context data,

(2) Method to comprehensively and accurately characterise the users and physical environment with the various sensing techniques with the domain knowledge of specific applications,

(3) Method to determine the underlying knowledge from history context with efficient KDD methods to supervise the function adaptation of mobile applications.

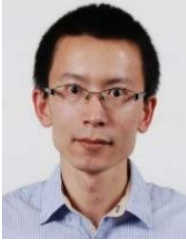
Since more and more novel sensing techniques are integrated to the handheld mobile devices, the architecture and methods presented in this investigation has the potential to revolutionise user interaction performance of applications of the resource limited mobile devices.

References

- [1] A. K. Triantafyllidis, C. Velardo, D. Salvi, S. A. Shah, V. G. Koutkias, and L. Tarassenko, "A survey of mobile phone sensing, self-reporting, and social sharing for pervasive healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp.218-227, 2017. [Article \(CrossRef Link\)](#)
- [2] H. K. Pung, T. Gu, W. Xue, P. P. Palmes, J. Zhu, W. L. Ng, C. W. Tang, N. H. Chung, "Context-aware middleware for pervasive elderly homecare," *IEEE Journal on Selected Area in Communications*, vol. 27, no. 4, pp. 510-524, 2009. [Article \(CrossRef Link\)](#)
- [3] Y. Chen, C. Shen, "Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition," *IEEE Access*, vol. 8, pp. 3095-3110, 2017. [Article \(CrossRef Link\)](#)
- [4] W. Z. Khan., Y. Xiang, M. Y. Aalsalem, Q. Arshad, "Mobile phone sensing systems: a survey," *IEEE Communications Survey & Tutorials*, vol. 15, no. 1, pp. 402-427, 2013. [Article \(CrossRef Link\)](#)
- [5] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp.386-403, 2017. [Article \(CrossRef Link\)](#)
- [6] R. K. Ganti, F. Ye, H. Lei, "Mobile Crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32-39, 2011. [Article \(CrossRef Link\)](#)

- [7] M. Weiser, "The computer for 21st century," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 3, o. 3, pp. 3-11, 1999. [Article \(CrossRef Link\)](#)
- [8] B. Schilit, M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22-32, 1994. [Article \(CrossRef Link\)](#)
- [9] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001. [Article \(CrossRef Link\)](#)
- [10] A. Zimmermann, A. Lorenz, R. Oppermann, "An operational definition of context," *Lecture Notes in Computer Science*, vol. 4635, pp.558-571, 2007. [Article \(CrossRef Link\)](#)
- [11] C. Bolchini, C. A. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A. Schreiber, L. Tanca, "And what can context do for data?" *Communications of the ACM*, vol. 52, no. 11, pp. 136-140, 2009. [Article \(CrossRef Link\)](#)
- [12] L. Nachabe, M. Girod-Genet, and B. E. Hassan, "Unified Data Model for Wireless Sensor Network," *IEEE Sensor Journal*, vol. 15, no. 7, pp. 3657-3667, 2015. [Article \(CrossRef Link\)](#)
- [13] C. Hoareau, I. Satoh, "Modeling and processing information for context-aware computing: a survey," *New Generation Computing*, vol. 27, pp. 177-196, 2009. [Article \(CrossRef Link\)](#)
- [14] R. Hardy, E. Rukzio, M. Wagner, M. Paolucci, "Exploring expressive NFC-based mobile phone interaction with large dynamic displays," in *Proc. of 2009 1st International Workshop on Near Field Communications*, Hagenberg, Austria, pp. 36-41, 2009. [Article \(CrossRef Link\)](#)
- [15] E. Toye, A. Madhavapeddy, R. Sharp, D. Scott, A. Blackwell, E. Upton, "Using camera-phones to interactive with context-aware mobile service," *Technical reports published by the University of Cambridge Computer Laboratory*, vol. 609, pp. 1-23, 2004. [Article \(CrossRef Link\)](#)
- [16] E. Miluzzo, T. Wang, A. T. Campbell, "EyePhone: activating mobile phones with your eyes," in *Proc. of the Second ACM SIGCOMM Workshop on Networking Systems, and Applications, on Mobile Handhelds*, 30 August, New Delhi, India, pp. 15-20, 2011. [Article \(CrossRef Link\)](#)
- [17] H. Lu, W. Pan, N. D. Lane, T. Choudhury, A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in *Proc. of the 7th International Conference on Mobile systems, Applications, and Services*, 22-25 June, Wroclaw, Poland, pp. 165-178, 2009. [Article \(CrossRef Link\)](#)
- [18] A. T. Campbell, T. Choudhury, S. Hu, H. Lu, M. K. Mukerjee, M. Rabbi, R. D. Raizada, "NeuroPhone: brain-mobile phone interface using a wireless EEG headset," in *Proc. of the Second ACM SIGCOMM Workshop on Networking, System, and Applications on Mobile Handhelds*, 30 August, Delhi, India, pp. 15-20, 2010. [Article \(CrossRef Link\)](#)
- [19] Y.-C. Tung, K. G. Shin, "ForcePhone: Software Lets Smartphones Sense Touch Force," *IEEE Pervasive Computing*, vol. 15, no. 4: 20-25, 2016. [Article \(CrossRef Link\)](#)
- [20] Q. K. Dey, G. D. Abowd, D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-Computer Interaction*, vol. 16, pp. 97-166, 2001. [Article \(CrossRef Link\)](#)
- [21] H. Chen, "An intelligent broker architecture for context-aware systems," PhD Thesis, University of Maryland, 2003.
- [22] G. Chen, "Solar: Building a context fusion network for pervasive computing," PhD Thesis, Dartmouth College, 2004.
- [23] T. Gu, H. K. Pung, D. Q. Zhang, "A service-oriented middleware for building context-aware service," *Journal of Network and Computer Applications*, vol. 28, pp. 1-18, 2005. [Article \(CrossRef Link\)](#)
- [24] A. Padovitz, S. W. Loke, A. Zaslavsky, "The ECORA framework: A hybrid architecture for context-oriented pervasive computing," *Pervasive and Mobile Computing*, vol. 4, pp. 182-215, 2008. [Article \(CrossRef Link\)](#)
- [25] A. Shrestha, "MobileSOA framework for context-aware mobile applications," in *Proc. of 7th International Conference on Mobile Data Management*, 10-12 May, Nara Japan, pp. 297-298, 2010. [Article \(CrossRef Link\)](#)
- [26] R. Tesoriero, J. A. Gallud, M. D. Lozano, V. M. R. Penichet, "CAUCE: Model-driven development of context-aware applications for ubiquitous computing environments," *Journal of Universal Computer Science*, vol. 16, no. 15, pp. 2111-2138, 2010. [Article \(CrossRef Link\)](#)

- [27] P. Bellavista, A. Corradi, M. Fanelli, L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM computing Survey*, vol. 44, no. 4, pp. 24: 1-45, 2013. [Article \(CrossRef Link\)](#)
- [28] M. Moazzami, D. E. Philips, R. Tan, G. Xing, "ORBIT: A platform for smartphone-based data-intensive sensing platform," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 801-815, 2017. [Article \(CrossRef Link\)](#)
- [29] A. Rarău, K. Pusztai, I. Salomine, "MultiFacet item based context-aware applications," *International Journal of Computing & Information Science*, vol. 3, no. 2, pp. 10-18, 2005. [Article \(CrossRef Link\)](#)
- [30] P. TalebiFard, V. C. M. Leung, "A data fusion approach to context-aware service delivery in heterogeneous network environments," *Procedia Computer Science*, vol. 5, pp: 312-319, 2011. [Article \(CrossRef Link\)](#)
- [31] A. Rarău, "Multifacet paradigm – support for context-aware application development," *Dissertation*, Technical University of Cluj-Napoca, 2006.
- [32] R. Schäfer, "Rules for using Multi-Attribute Utility Theory estimating a user's interests," in *Proc. of Workshop Adaptivity and User Modelling*, 2001. [Article \(CrossRef Link\)](#)
- [33] A. Al-Hmouz, J. Shen, R. Al-Hmouz, J. Yan, "Modeling and simulation of an Adaptive Neuro-Fuzzy Interface System (ANFIS) for mobile learning," *IEEE Transaction on Learning Technologies*, vol. 5, no. 3, pp. 226-237, 2012. [Article \(CrossRef Link\)](#)
- [34] Q. Duan, D. Miao, H. Zhang, J. Zheng, "Personalised Web retrieval based on Rough-Fuzzy method," *Journal of Computational Information System*, vol. 3, no. 2, pp. 203-208, 2007. [Article \(CrossRef Link\)](#)
- [35] X. Wang, D. Rosenblum, Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proc. of the 20th ACM International Conference on Multimedia*, 29 Oct. - 2 Nov., Nara Japan, pp. 99-108, 2012. [Article \(CrossRef Link\)](#)
- [36] A. Schmidt, "Ubiquitous computing—computing in context," *Dissertation*, Lancaster University, 2012.
- [37] J. Parra, M. A. Hossain, A. Uribarren, E. Jacob, A. E. Saddik, "Flexible smart home architecture using device profile for Web service: a Peer-to-Peer approach," *International Journal of Smart Home*, vol. 3, no. 2, pp. 39-56, 2009. [Article \(CrossRef Link\)](#)
- [38] P. Chen, X. Zhi, "Smart home architecture based on event-driven DPWS," *Journal of Shanghai University*, vol. 15, no. 5, pp. 386-390, 2011. [Article \(CrossRef Link\)](#)
- [39] T. Gu, H. K. Pung, D. Q. Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 66-74, 2004. [Article \(CrossRef Link\)](#)
- [40] R.-T. Lin, C.-S. Hsu, T. Y. Chun, S.-T. Cheng, "OSGi-based smart home architecture for heterogeneous network," in *Proc. of 3rd International Conference on Sensing Technology*, 30 Nov.-3 Dec., Tainan, Taiwan, pp. 527-532, 2008. [Article \(CrossRef Link\)](#)
- [41] A. Casals, A. A. F. Brandão, "Modeling a mobile learning context data ontology," In: 2017 IEEE World Engineering Education Conference (EDUNINE), pp. 71-75, 2017. [Article \(CrossRef Link\)](#)
- [42] M. Poveda-Villalon, et al., "A context ontology for mobile environments," in *Proc. of Workshop on Context, Information and Ontologies Co-located with EKAW*, 2010. [Article \(CrossRef Link\)](#)
- [43] E. F. Krause, "Taxicab Geometry: An Adventure in Non-Euclidean Geometry," *Dover Publications*, New York, 1987.
- [44] S. Benabderrahmane, M. Smail-Tabbone, O. Poch, A. Napoli, M.-D. Devignes, "IntelliGO: a new vector-based semantic similarity measure including annotation origin," *BMC Bioinformatics*, vol. 11, pp. 1-16, 2010. [Article \(CrossRef Link\)](#)
- [45] P. Ganesan, H. Garcia-Molina, J. Widom, "Exploiting hierarchical domain structure to compute similarity," *ACM Transactions on Information System*, vol. 21, pp. 64-93, 2013. [Article \(CrossRef Link\)](#)



Zhaozong Meng received the B.Sc. degree in instrumentation and measurement from Sichuan University, Chengdu, China, in 2006, the M.Sc. degree in instrumentation and measurement from Beihang University, Beijing, China, in 2009, and the Ph.D. degree in computer science from the University of Huddersfield, Huddersfield, U.K., in 2014. He worked in The University of Manchester, Manchester, U.K., as a Research Associate from 2014 to 2016. He is now a Research Fellow in the University of Southampton, Southampton, UK. His research interests include sensors, instrumentation and measurement, wireless sensor networks (WSN), pervasive and mobile computing, and Internet of Things (IoT).