Journal of Electrical Engineering 3 (2015) 60-69
doi: 10.17265/2328-2223/2015.02.002

Enforcement of CA-UCON Model

Abdulgader Almutairi[1] and Francois Siewe[2]

1. Computer Science College, Al Qassim University, Burydah 51412, Kingdom of Saudi Arabia

2. Software Technology Research Laboratory, Faculty of Technology, De Montfort University, Leicester, LEI 9BH, United Kingdom

Abstract: A CA-UCON (Context-Aware Usage CONtrol) model is an extension of the traditional UCON (Usage CONtrol) model which enables adaptation to environmental changes in the aim of preserving continuity of access. When the authorisations and obligations requirements are met by the subject and the object, and the conditions requirements fail due to changes in the environment or the system context, CA-UCON model triggers specific actions to adapt to the new situation. Besides the data protection, CA-UCON model also enhances the quality of services, striving to keep explicit interactions with the user at a minimum. In this paper, we propose an architecture of the reference monitor for the CA-UCON model and investigate a variety of enforcement approaches in ubiquitous computing systems; whether centralised, distributed or hybrid; depending on applications.

Key words: Pervasive system, context-aware, usage control, enforcement.

1. Introduction

One of the most rapidly developing areas in ICT (information and communications technology) is known as “ubiquitous computing”, first introduced by Weiser [1]. It refers to the ever increasing phenomenon of integrating and embedding ICT tools in people’s daily lives and in the situations or environment in which they live [2-4]. This has been made possible by the ever improving developments in the manufacturing of microprocessors, which now have built-in communication functions and other amenities. There are many applications available for ubiquitous computing, ranging from health-care, home-care, environmental monitoring, intelligent transport systems management and monitoring, etc. In order to function correctly, ubiquitous computing systems collect and share a great deal of information about the users, their mobile devices and environment. Securing the use of this information is one of the most important challenges in ubiquitous computing, paramount for the widespread acceptance of such a technology.

The CA-UCON (Context-Aware Usage CONtrol) model [5] is the latest major enhancement of the traditional access control models which enables (i) mutability of subject and object attributes; (ii) continuity of control on usage of resources; and (iii) adaptation to environmental changes in the aim of preserving continuity of access despite changes in user context. The concept of mutability refers to the fact that attributes are not static but do change intermittently. Continuity of access decision ensures that decision to permit and allow access to an object is made constantly before and during the access to an object. This access decision is based on three key factors: authorisation, obligation and condition requirements. When the authorisation and obligation requirements are met by the subject and the object, and the condition requirements fail due to changes in the environment or the system context, specific actions are automatically triggered to adapt to the new situation, so as to minimise disruption to the users. These make CA-UCON a suitable usage control model for ubiquitous computing systems [5].

This paper addresses the enforcement of the CA-UCON model in a ubiquitous computing environment. The contributions of the paper are twofold:

Corresponding author: Abdulgader Almutairi, assistant professor, Ph.D., research fields: context-aware computing, pervasive systems, cyber security, access control, and formal methods.

(1) An architecture of the CA-UCON RM (CA-UCON Reference Monitor) is proposed (Section 3); its innovative features include context-awareness and dynamic adaptation to changing environmental situations;

(2) The deployment of CA-UCON RM is investigated based on three different approaches: centralised, distributed and hybrid; depending on ubiquitous computing applications (Section 4).

2. Overview of CA-UCON Model

A CA-UCON model is an extension of the traditional UCON model [6] that enables adaptation to environmental changes in the aim of preserving continuity of access. Indeed, when the authorisations and obligations requirements are met by the subject and the object, and the conditions requirements fail due to changes in the environmental or the system context, CA-UCON model triggers specific actions to adapt to the new situation. Besides the data protection, CA-UCON model also enhances the quality of services, keeping explicit interactions with the user at a minimum.

The behaviour of the CA-UCON model can be described using a FSM (finite state machine) depicting how a subject's request to access an object is handled in the CA-UCON model. The FSM is shown in Fig. 1, where nodes are called states and edges are called transitions. The initial state, labelled initial, corresponds to the state when the system is waiting for a subject to submit a request.

There are three final states: *end*, when the access has successfully terminated; *denied*, when the access request has been denied; and *revoked*, when access permission has been revoked during access and hence the access stopped. The intuitive meaning of the remaining states of the FSM can be summarised as follows: *requesting*, denotes when the access request is being processed; *accessing*, represents the state when the actual access is taking place; *preadapting*, is the state when the system is trying to adapt to the environmental context prior to access; and finally *onadapting*, is when the system is trying to adapt to the environmental context during access.

The transitions of the FSM are labelled with the events (or actions) that fire them. The event *tryaccess* occurs when a subject sends an access request (e.g. by clicking a menu button). This event forces the FSM to enter the requesting state to process that access request. While in this state, the system can perform updates on the subject's and object's attributes through *preupdate* events. If the authorisations, obligations and conditions

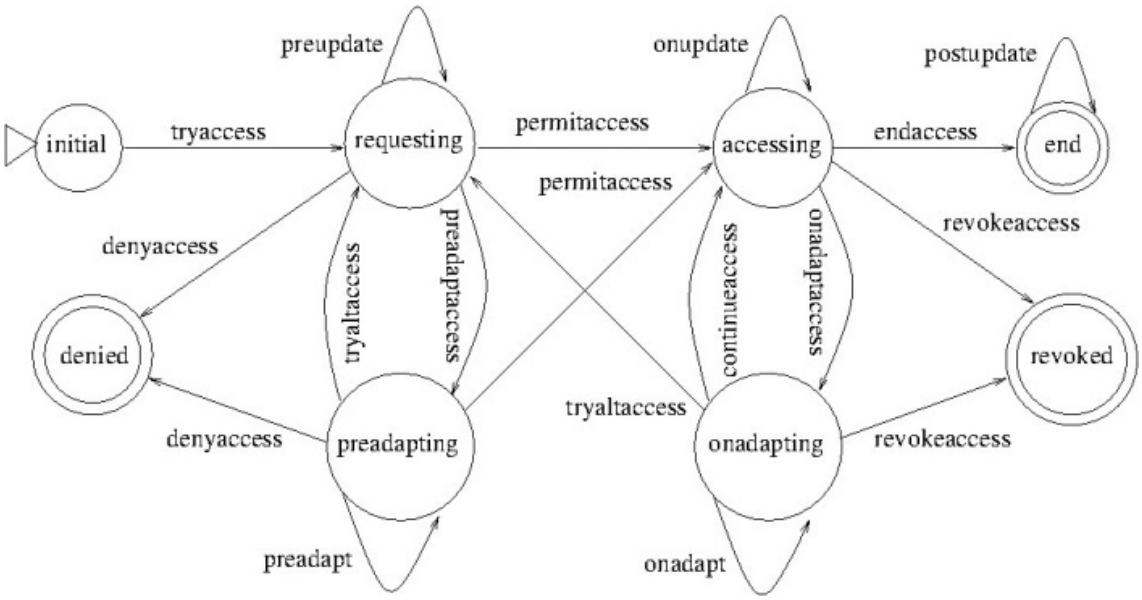

Fig. 1 Execution of an access request in the CA-UCON model.

requirements are all met, the system emits the *permitaccess* event and moves into the accessing state. If for some reasons either the authorisations requirement or the obligations requirement is not met, the system emits the event *denyaccess* and terminates in the denied state.

However, if both the authorisations requirement and obligations requirement are met, but the conditions requirement is not satisfied, the system emits the *preadaptaccess* event and moves into the *preadapting* state. In this state, specific adaptation actions, denoted by the preadapt events, are performed in an attempt to meet the conditions requirement. If the adaptation is successful, the *permitaccess* event is raised and the system transitions into the accessing state. In addition, a new request to access a specified alternative object, denoted by the *tryaltaccess* event, may be issued automatically by the system if the adaptation actions fail. Otherwise the access request is simply denied when no adaptation is possible.

When access permission is granted (see *permitaccess* event), the system transitions into the accessing state in which the actual access takes place. During access the system can perform updates on subject's and object's attributes via *onupdate* events. If during access either the authorisations requirement or the obligations requirement is not met, the system emits the event *revokeaccess* and terminates in the revoked state. However, if both the authorisations requirement and obligations requirement are continuously met, but the conditions requirement fails, the system raises the *onadaptaccess* event and moves into the *onadapting* state. In this state, specific adaptation actions, denoted by the *onadapt* events, are performed in an attempt to meet the conditions requirement. If the adaptation is successful, the *continueaccess* event is raised and the system moves back into the accessing state. In the effort to enhance the quality of service even further, the system might issue an implicit request to access a specified alternative object through the *tryaltaccess* event, when the adaptation actions fail. In the worst case when no adaptation is possible, the access permission is simply revoked and the access stopped at once.

When an access terminates successfully via the *endaccess* event, the system moves into the end state and eventually performs updates on the subject's and object's attributes through *postupdate* events. Further details of the model can be found in Ref. [5].

3. Architecture of CA-UCON Reference Monitor

In this section, we propose an architecture of the CA-UCON RM (CA-UCON reference monitor) as depicted in Fig. 2, where computing components are represented in a rectangular shape, data storage in a cylindrical shape, and arrows indicate interactions between components. In the following subsections, we explain in detail each component along with its role in this architecture.

3.1 Enforcement Point

The role of this component is to receive requests from the subject that wishes to access an object and to issue decision requests to the decision point component—in particular to the UD (usage decision) component—and to wait for the response from it, whether PERMIT or DENY. It then lets the subject access the service if the reply from decision point component is PERMIT; otherwise, it denies access to the subject. This behaviour is typical to many access control models [7-9].

3.2 Decision Point

Unlike in other access control models [8-10], this component consists of two parts, UD (usage decision) component and AD (adaptation decision) component. The responsibility of the UD component is the evaluation of each access request that receives from enforcement point component. UD checks the authorization, obligation and condition requirements of

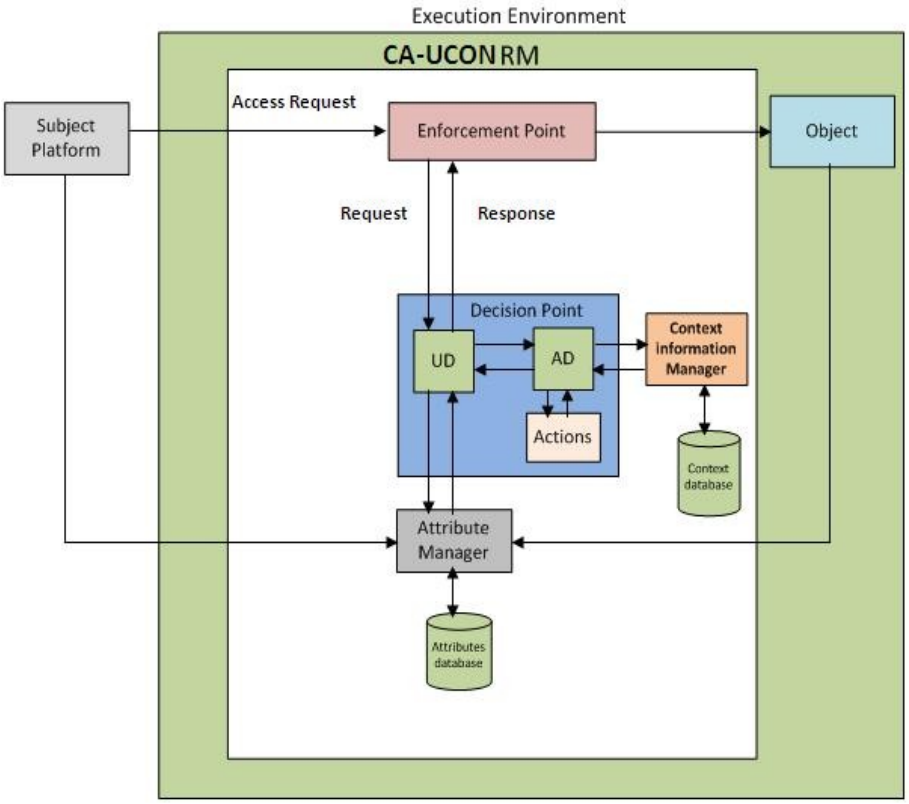

Fig. 2 Architecture of the CA-UCON reference monitor.

each request in order to permit or deny the access request. To do so, UD communicates with the attribute manager component in order to acquire the values of the attributes of the subject, object and environment. If all the requirements are met, UD responds by PERMIT to the enforcement point component to allow the access to take place.

However, if the authorization and obligation requirements are met, but the condition requirement is not fulfilled by the environment, UD communicates with the AD component in order to adapt to the new situation. AD interacts with the context information manager component to identify the current context of the subject, object and environment. It then performs appropriate actions in an attempt to make the environment meet the condition requirement. This attempt will last for a specified period of time after which the adaptation will be deemed successful or unsuccessful. If successful, the access request will be granted, otherwise it will be denied.

3.3 Attribute Manager

The responsibility to this component is to provide access to the database of subject and object attributes in order to use them in access decision. All subject and object attributes will be stored in a database and updated as they change.

3.4 Context Information Manager

This component is responsible for monitoring and evaluating the context information such as subject context, object context and environment context. It uses a variety of sensors to sense different types of contexts. It communicates with AD component in order to provide the current context of the subject, object and environment.

4. Enforcement Architecture of CA-UCON Model

In this section, we present three different enforcement approaches in order to demonstrate how

the CA-UCON model can be implemented in real world situations. The CA-UCON model can be enforced in a number of ways depending on the application. These enforcement approaches are known as the centralized approach, the distributed approach and the hybrid approach. The following subsection will explain each approach in detail to show the difference between these approaches, their usages, and possible benefits and limitations.

4.1 Centralized Enforcement Architecture

A centralized architecture is dependent upon one node being designated as the computer node or server. This node executes and runs the complete application locally and all users share this central system. Hence control and failure are concentrated in a single place, the server. The CA-UCON RM will be implemented in the server side only. This entails control of the services being the responsibility of the server alone as depicted in Fig. 3.

How does this model support adaptation? The server is installed with a variety of sensors (physical and logical) which are able to record the context and behaviour of the subjects, objects and environment. The adaptation to the new situation and sending of the appropriate service to the user will occur based on these contexts and behaviours. Thus, adaptation required for any service will be initiated by the server. This is a strict method of enforcement since all controls and adaptations are carried out centrally on the server where CA-UCON RM is implemented.

The centralized architecture presents a number of benefits. Firstly, avoidance of duplication: a centralised approach facilitates having a single version of any information system for the entire organisation. They also help to ensure that every piece of data is stored only once. Secondly, sharing resources is easy since data are stored in a single place, accessible to all legitimate users. This makes the system maintenance and (security) administration more efficient. Central

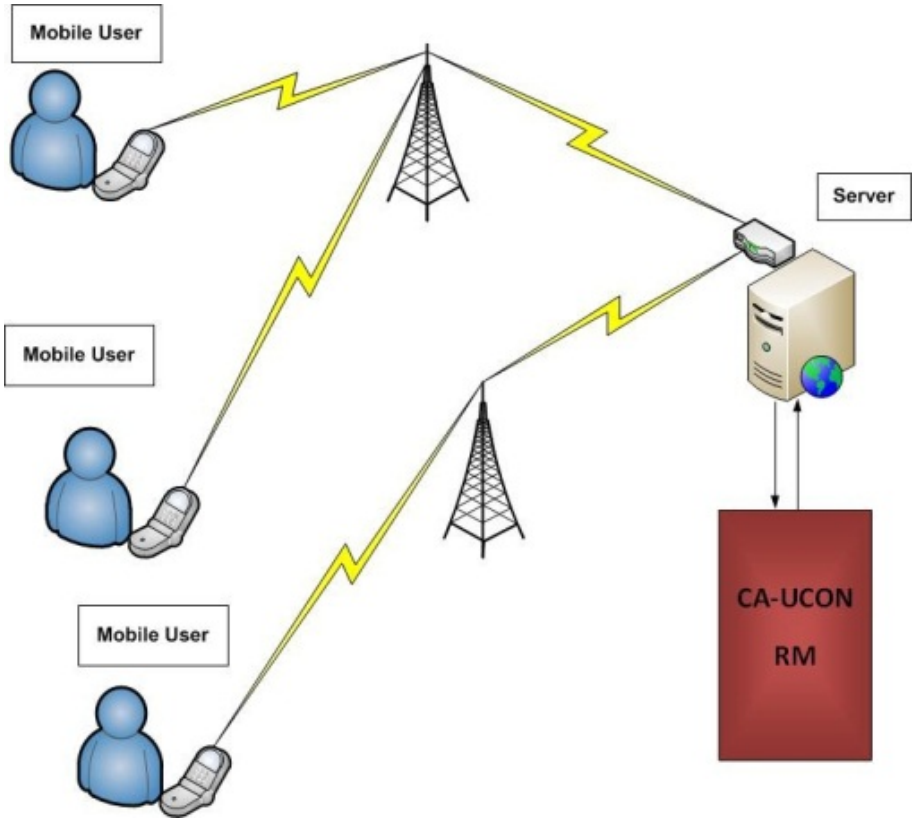

Fig. 3 Centralised enforcement architecture.

process and planning also permits well-suited technology and skills to be established. The sharing of resources other than data is simplified. The transfer between units of hardware, software and employees becomes easier. Finally, scale economies are attained, since centralised approaches permit most activities to be carried out with lower unit cost.

From amongst the disadvantages of the centralized architecture is heavy time consumption, since carrying out actions and decisions centrally is additional time consuming than non-centralised approach. This is due to the extra time it takes to assemble information from various different distributed places as input to centralised system decisions. Furthermore, when the central computer or database system fails, then the system is inaccessible to anyone until the server/database recovers. Additionally, rigidity and increased dependence and susceptibility are inherent to the centralized architecture.

Example 1: Suppose a ubiquitous learning (u-learning) system that provides u-lectures in three different formats: video, audio and text. The access to each format is restricted by specific requirement on context of the user. If the user requests a u-lecture in the video format when she/he is driving, the system will not deliver it to her/him in the video format, but it will deliver it in the audio format which is suitable to the user current context (driving). In the centralised approach, the server may use a location tracking technique to sense the context of the user (whether driving or not) and then decide in what format the u-lecture must be delivered.

4.2 Distributed Enforcement Architecture

The distributed architecture can be understood in two distinct ways. It can be defined by the physical components or defined from the angle of the user or computation. These are known as the physical view and the logical view respectively. A distributed system is a set of nodes (computers or portable devices) connected by a communication network. The nodes in the network do not share their memory and are loosely coupled. The nodes in the system communicate via passing messages over the communication network using communication protocols.

The logical model is the view that an application has of the system. It includes a set of simultaneous processes and communication routes between them. The centre network is treated as completely linked. Communication of processes is done by transferring messages to each other.

In either view, each node in the distributed system is responsible for its own security as depicted in Fig. 4. The CA-UCON RM will be installed on each node (device) which will be responsible for handling access request in the manner described in Section 3. The enforcement of distributed architecture exhibits a number of benefits. Prominent amongst these is greater compatibility between systems and local needs, since users can develop their own information systems. These are more likely to correspond with their needs than those developed by someone else. Another benefit is greater system usage, as users show increased motivation by such approaches and are thus more enthusiastic to adopt computing when it directly bolsters their own interests, benefits and work. Finally, it supports quicker system development, since the less the organisational distance between system user and system developer, the swifter the development of that system is likely to be.

On the other hand certain negatives are displayed. This approach places barriers to sharing data. Distributed approaches can produce information systems in individual work units that are incompatible with each other and asynchronous. Further, effort is duplicated, since units will often duplicate what others are doing. Therefore distributed approaches tend to be very costly. Distribution can also lead to a failure to achieve scale economies as activities are not pooled.

Example 2: Following up from Example 1, suppose each u-lecture format requires some minimum amount of free memory available on the user’s mobile device,

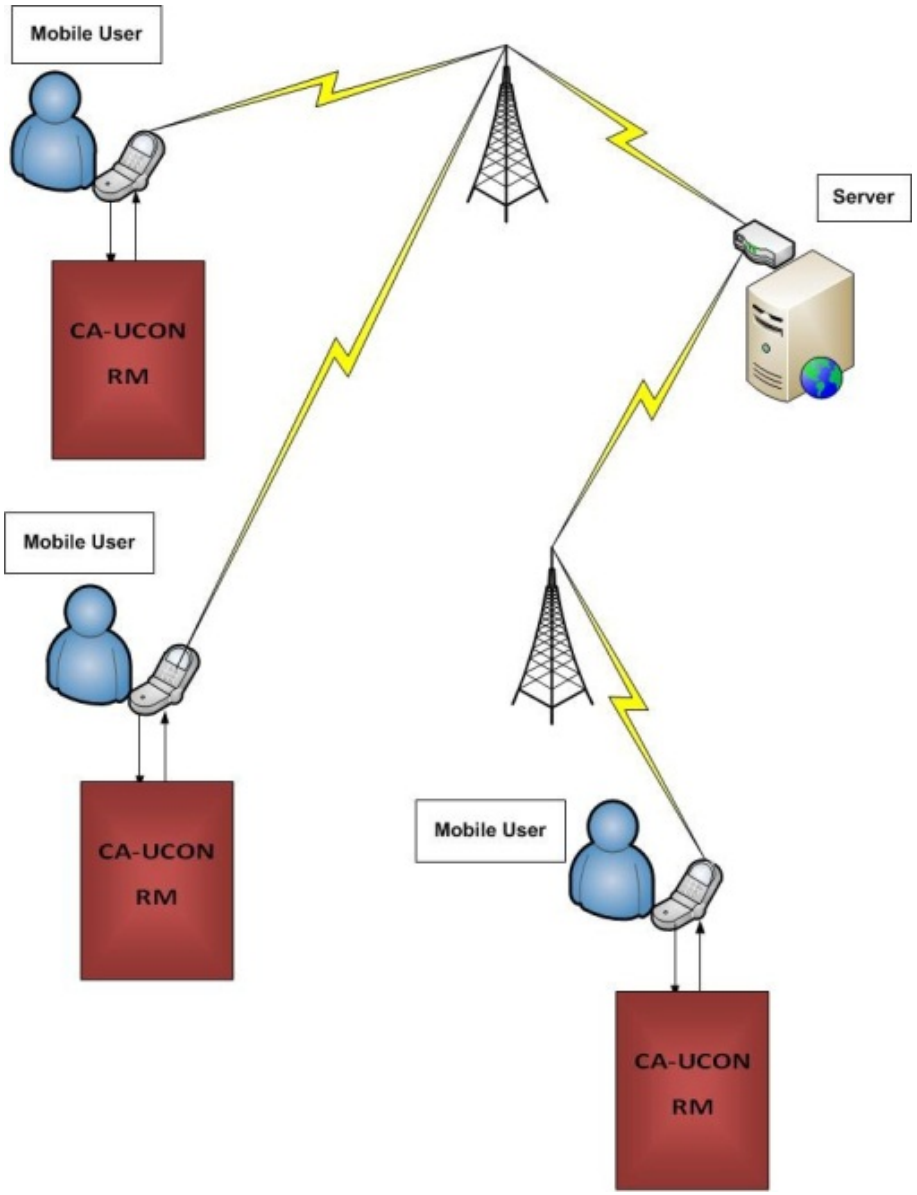

Fig. 4 Distributed enforcement architecture.

e.g.: 5 MB for a video, 2 MB for an audio, and 1MB for a text format. If a user requests a u-lecture in video format and the available memory on her/his mobile device is less than 5 MB, the system needs to adapt to this new situation. In a distributed enforcement architecture where CA-UCON RM is installed on each mobile device, the adaptation can be done e.g. by turning on a garbage collector software on the mobile device (client side). This will eventually increase the size of available memory on the mobile device. If enough memory has been freed and the current available memory on the mobile device is greater than 5 MB, then the adaptation is successful and access will be granted.

4.3 Hybrid Enforcement Architecture

Hybrid enforcement architecture is an interesting evolutionary architecture that has combined features of centralised enforcement architecture and distributed enforcement architecture. The purpose is to circumvent the disadvantages displayed by these architectures whilst maintaining many of their advantages. In this

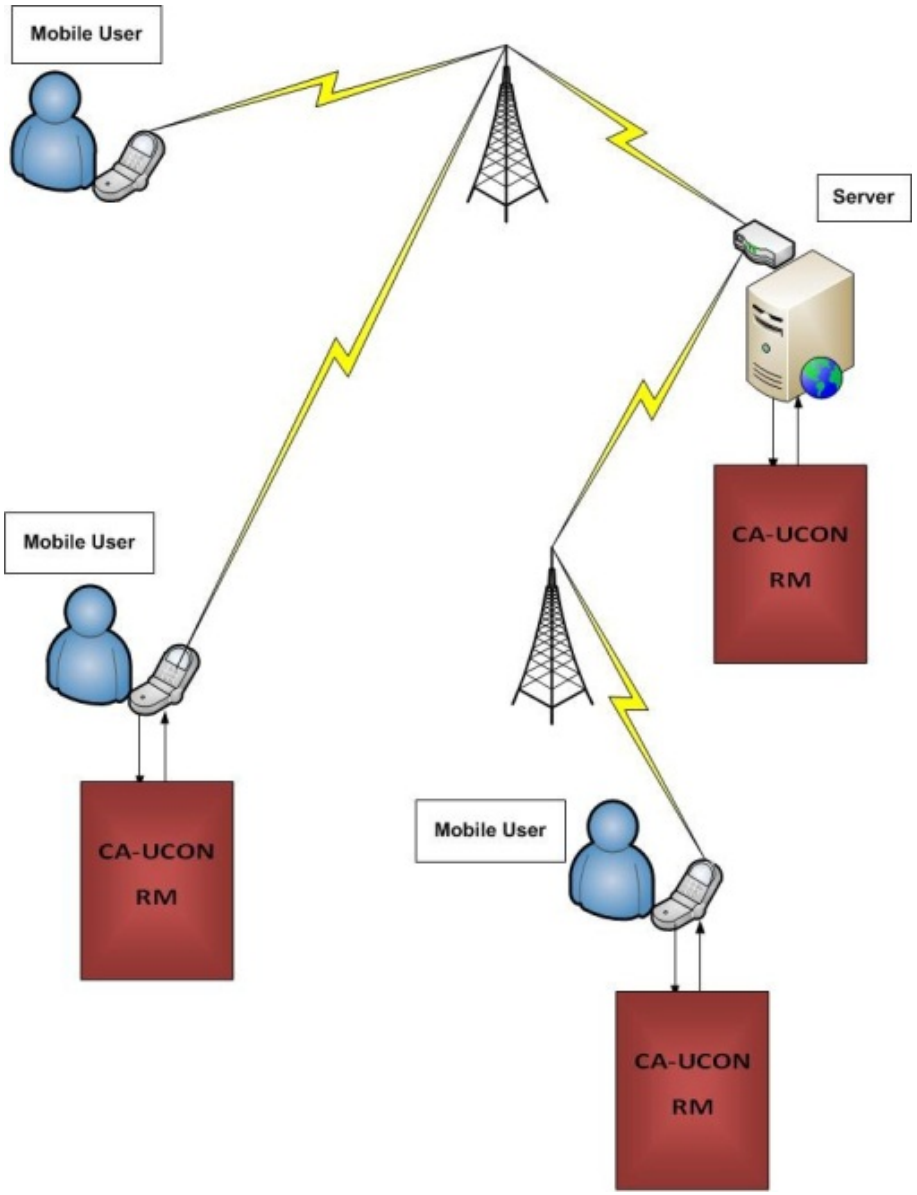

Fig. 5 Hybrid enforcement architecture.

architecture the CA-UCON RM is enforced in both the client side and the server side as can be seen in Fig. 5. Evaluation of the requested access right will take place on the mobile device side or on the server side, or on both. This enables the controlling of service or adapting to a new situation to occur at any time anywhere without restrictions. Some requested service accesses cannot be undertaken without adaptations from both the server and client side. This is difficult to do in the previously discussed architectures. On the other hand, the hybrid approach facilitates the implementation of the CA-UCON RM in the real world on both server and client sides, enabling more flexible adaptations. Hybrid architecture is a comprehensive and versatile approach. By combining two different architectures, a third, more tailored architecture for the enforcement of the CA-UCON RM has been evolved. Mobile devices and servers should be fitted with a variety of sensors in order to monitor all contexts related to the subject, object and environment. This is done so as to adapt to new situations and deliver a suitable service for the majority of contexts.

The advantages of the hybrid architecture include reliability, because a fault detected in one part of the system can be isolated from the rest. Therefore the necessary corrective measures can be carried out, without disturbing the operating of any other part of the system. Furthermore, hybrid architecture displays more flexibility than other architectures since it integrates the benefits of both centralized and distributed architectures optimizing the existing resources whilst avoiding many drawbacks. On the other hand, the hybrid architecture involves a larger scaleof work and therefore is normally more costly.

Example 3: In the u-learning system described in Examples 1 and 2 above, if the user requests a u-lecture in video format when she/he is driving and the available memory size in her/his mobile device is less than 2 MB (i.e., 2 megabytes), the system in this case needs to adapt to the new situation by performing actions on both sides (server side and client side). So, in the server side the system has to deliver the audio format instead of video format based on the user context which is in this case, driving. Meanwhile, the system has to check the amount of free memory available on the mobile device prior to delivering the service, which in this case is less than 2 MB, i.e., lower than the minimum memory size requirement for accessing a u-lecture in audio format. In this case the client side (mobile device) has to adapt as well, e.g., by executing a garbage collector software in order to free more memory space. If both adaptation actions are successful, then access will be granted; otherwise access will be denied. In this scenario, CA-UCON RM is installed on both the server and the user mobile device (client).

5. Related Work

There have been some works conducted in term of usage control enforcement. In spite of the short period of time from the time when the notion of usage control was introduced, there are a lot of substantial efforts in order to create an appropriate enforcement mechanism. For instance, Sandhu et al. [11] proposed a new approach known as a TRM (trusted reference monitor) which is placed in client side in order to enforce policies. So, the protocol that is used between TRM and application, as well as between different TRMs is relied on challenge-response. Any access request from receiver is followed by a challenge. The requester consequently confirms the application, platform or environment throughout the means of a digital signature.

Another approach was proposed by Abie et al. [12] which is similar strategy to the above approach, it is based on a hardware-based TPM (trusted platform module) in order to enforce usage policies on digital objects. It is known as SEOs (self-enforcing objects) which is used as secure container for transferring objects and policies and also has the ability to enforce the attached policies on any trusted platform autonomously. Moreover, a similar enforcement approach was proposed by Nauman et al. [13] to explain the notion of platform attestation. The system ensures via a WS-Attestation procedure that receiving platform must act correctly before the information is released. HUE (hardware UCON engine) is another enforcement approach proposed by Matson et al. [14] which is a different from the above mentioned approaches. HUE is considered as secure co-processor with a designated software stack in order to offer integration with the operating system. Next, the Right-Enforcer was proposed by Alam et al. [15] known as a product that is used to enforce simple usage control restrictions, i.e., limiting the capability to view, copy, print and store. This approach is incorporated in an e-mail user, so whenever a usage controlled object is delivered to a receiver, the Right-Enforcer encrypts the content and then sends the term of use to a centralised Right-Server. Then the receiver is forced to use the Right-Enforcer in order to be able to decrypt the content and so the policy is always enforced.

However, majority of these enforcement approaches have been done in usage control model (UCON).

Unlike the above UCON enforcement approaches, CA-UCON model enables adaptation to environmental changes in the aim of preserving continuity of access by triggering specific actions to adapt to new situations. In addition to data protection, CA-UCON model enhances the quality of services, striving to keep explicit interactions with the user at a minimum. Therefore, the need for enforcement approach for an adaptive usage control such as CA-UCON to control the service in ubiquitous environment is significant.

6. Conclusions

In this paper, we have presented the architecture of the CA-UCON RM and explained the responsibility for each component and the interactions among them. We then, proposed three types of enforcement architectures of CA-UCON model, namely: the centralised enforcement architecture, the distributed enforcement architecture, and the hybrid enforcement architecture.

In future works, we will investigate actual implementation of the proposed architectures in a real-world ubiquitous computing application, such as a u-learning system.

References

[1] Weiser, M. 1991. "The Computer for the 21st Century." *Scientific American* 265 (3): 94-104.

[2] Horvath, I., and Peck, D. 2009. "Survey of Advanced Learning Solutions from Methodological and Technological Perspectives." In *Proceedings of 24th ASCILITE Conference*, Singapore.

[3] Cardelli, L., and Gordon, A. 2000. "Mobile Ambients." *Theoretical Computer Science* 240: 177-213.

[4] Weiser, M. 1993. "Some Computer Science Issues in Ubiquitous Computing." *Communication of the ACM* 36 (7): 75-84.

[5] Almutairi, A., and Siewe, F. 2011. "CA-UCON: A Context-Aware Usage Control Model." In *Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems (CASEMANS, 11)*, 34-8.

[6] Jaehong, P., and Ravi, S. 2004. "The UCON: Usage Control Model." *Journal of ACM Transactions on Information and System Security* 7(1): 128-74.

[7] Nyre, A. A. 2011. "Usage Control Enforcement—A Survey." In *Proceedings of the IFIP WG 8.4/8.9 International Cross Domain Conference on Availability, Reliability and Security for Business, Enterprise and Health Information Systems.*

[8] Samarati, P., and Vimercati, S. 2000. "Access Control: Policies, Models, and Mechanisms." In *Foundations of Security Analysis and Design.* Springer-Verlag, 137-96.

[9] Sandhu, R., Coyne, E. J., Feinstein, H. L., and Youman, C. E. 1996. "Role-Based Access Control Models." *IEEE Computer* 29(2): 38-47.

[10] Almutairi, S., Aldabbas, H., and Abu-Samaha, A. 2012. "Review on the Security Related Issues in Context Aware System." *International Journal of Wireless & Mobile Networks (IJWMN)* 4(3): 195-204.

[11] Sandhu, R., Zhang, X., Ranganathan, K., and Covington, M. J. 2006. "Client-side Access Control Enforcement Using Trusted Computing and PEI Models." *Journal of High Speed Network* 15(3): 229-45.

[12] Abie, H., Spilling, P., and Foyn, B. 2004. "A Distributed Digital Rights Management Model for Secure Information-Distribution Systems." *International Journal of Information Security* 3(2): 113-28.

[13] Nauman, M., and Ail, T. 2008. "A Hardware UCON Engine for Fine-Grained Continuous Usage Control." In *The IEEE International Multitopic Conference*, 59-64.

[14] Matson, M., and Ulieru, M. 2006. "The How and Why of Persistent Information Security." In *Proceedings of the International Conference on Privacy, Security and Trust*, 1-4.

[15] Alam, M., Seifert, J., and Li, Q. 2008. "Usage Control Platformization via Trustworthy SELinux." In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, 245-8.