Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **5**, September **2015**

Future Classroom with the Internet of Things A Service-Oriented Framework

Feng-Cheng Chang and Duen-Kai Chen

Department of Innovative Information and Technology
Tamkang University
180 Linwei Road, Jiaosi Township, Yilan County, 262, Taiwan
135170@mail.tku.edu.tw, dkchen@mail.tku.edu.tw

Hsiang-Cheh Huang*

Department of Electrical Engineering
National University of Kaohsiung
700 University Road, Kaohsiung, 811, Taiwan
*Corresponding author
hchuang@nuk.edu.tw

Received September, 2014; revised June, 2015

Abstract. *The future classroom emerges as an approach to enhance pedagogy. Classroom participants can become more involved in their learning by interaction and collaboration. This becomes practical due to the extensive availability of modern electronic devices and networks. To implement an environment that supports the future classroom, an integration mechanism for various types of devices is required. In this paper, a framework design is proposed using the concept of the Internet of Things (IoT). The IoT aims to model an application domain as a set of things that are associated with the Internet. By analyzing the project requirements of future classroom applications, the Extensible Messaging and Presence Protocol (XMPP) is selected as the communication method for the things. Two examples are also provided to demonstrate the effectiveness of the design.*

Keywords: Internet of Things, Future classroom, Extensible messaging and presence protocol

1. **Introduction.** The Internet and various types of information appliances are employed throughout a modern campus. Therefore, these items can be employed to provide improved pedagogy that has previously been proven impractical. The concept that is referred to as the future classroom emphasizes interaction and collaboration in the classroom. A straightforward idea is to incorporate the computers, the peripherals, and the network as part of the infrastructure of the classroom. We need a mechanism to efficiently construct a future classroom application on the infrastructure.

The concept of the Internet of Things (IoT) emerges as a technology to integrate almost everything on Earth with the Internet. The original idea is to make physical objects identifiable by attaching electronic tags. Thus, an object can be tracked by reading the identity, processing the associated information, and saving the results in a database. Over time, the concept of IoT has been evolved to include physical objects, sensors/actuators, smart devices, and software services. Due to the variety of things and signaling/communication

methods, the development of a "universal" IoT framework for all types of applications is not practical because an IoT framework is domain-specific.

Combining these concepts, the future classroom could be an interesting application domain of the IoT. To implement this type of classroom on a campus, the framework should be flexible for integrating the devices on the Internet and making them available to the applications. This paper is organized as follows: In Sec. 2, the important concepts and technologies are provided as the background information. In Sec. 3, the application domain is analyzed and the framework design is described. The effectiveness of the framework is illustrated by two examples in Sec. 4. Some issues regarding the implementation experiences are discussed in Sec. 5. The conclusions are provided in Sec. 6.

2. **Background.**

The related information about a future classroom with the Internet of Things (IoT) concept is provided in this section. The properties of a future classroom are described in Sec. 2.1. The fundamental concepts of the Internet of things are briefly introduced in Sec. 2.2. In our design, the Extensible Messaging and Presence Protocol (XMPP) network is the infrastructure for the services. This protocol is briefly introduced in Sec. 2.3.

2.1. **Future classroom.**

The idea of the "future classroom" has attracted significant attention from academia and practitioners. The future classroom is a concept that is distinguished from the traditional classroom by the addition and enhancement of some physical components to enrich students' psychological experiences while learning. Although there is no standard definition of the future classroom, multiple studies, e.g., [1], have outlined what characterizes a future classroom. The following common properties are considered to define the future classroom:

- **Interaction**: The teaching material is designed to guide the learning progress via interactions. New information is adaptively revealed according to a student's response to the previous information.
- **Collaboration**: The teaching material is designed to encourage interactions among students. Regardless of how well the teaching content is designed, it is relatively static compared with human thought. Encouraging students to interact with each other is a suitable approach to inspiring diverse and deep thinking.
- **Flexibility**: This property is related to the physical configuration of the learning environment. To support the previous two properties, a flexible and optimal arrangement of the facilities inside the classroom is required. For example, we should be able to easily move the tables to form discussion groups, access digital media, and receive feedback.

To create a classroom with these properties, the pedagogy, materials, and supporting facilities should be revised. Modern information appliances and infrastructure could be the fundamental components of novel classroom facilities. A few application scenarios were discussed in our research group; we discovered that the variety of education-centric facilities is dependent on the application scenario. Therefore, an integration mechanism is needed to associate the users, application, and facilities.

2.2. **The Internet of things.**

Computers and various information appliances are connected by the Internet. As the number of digital devices increases, demands to develop the "Future Internet" to support the information flows will become evident. Although the definition of the Future Internet is divergent, some of the properties are recognized. For example, the available addresses should be sufficiently large to identify every device on Earth and the devices should communicate with each other to accomplish a certain task.

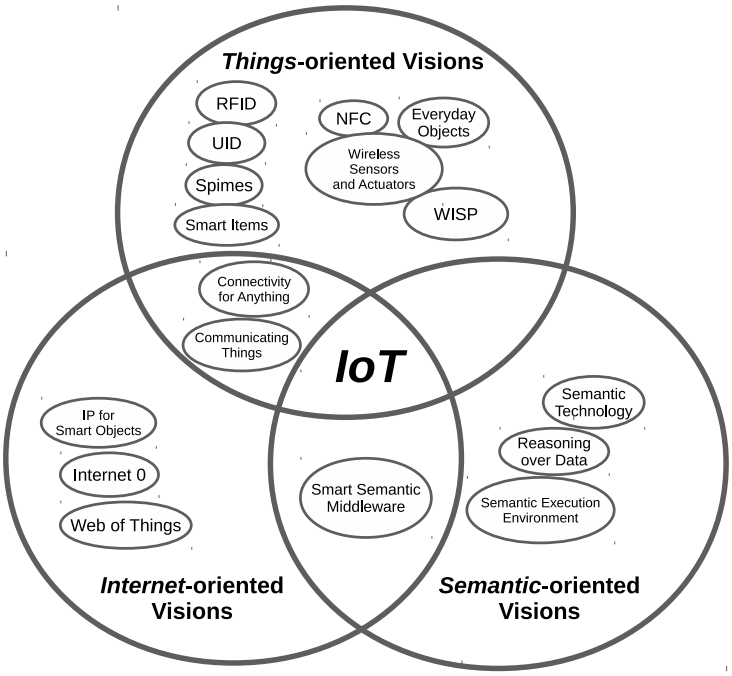


FIGURE 1. Perspectives of the Internet of Things [11]

The Internet of things (IoT) was initially introduced by the MIT Auto-ID Labs. The original concept is to identify an object by a unique digital identity to ensure that its associated information can be retrieved and processed. The electronic product code (EPC) and the unique identifier (UID) are the popular standards that specify the format of the identity. With the RFID tag as the carrier, we can attach a tag to an object, read the identity of the object, send the identity of the object to the Internet, and process the associated information. Thus, the RFID tag and the reader can bridge the physical world to the digital world.

Because additional IoT applications and architectures have been proposed, the concept of the IoT has evolved to include technologies other than the RFID tag [2, 3], such as the sensor network [4, 5, 6, 7], service-oriented architecture [8], and the web service. Many people consider the IoT to be one of the structures of the Future Internet [9, 10]. Because no consensus on the Future Internet has been reached, the definition of the IoT is manifold [11, 12, 13]. The IoT problems may be viewed from three perspectives as shown in Fig. 1.

When considering the IoT as things with information that is available on the Internet, the main objective is to make a physical object identifiable. Therefore, the identity formats (e.g., the EPS and UID) and the carriers (e.g., RFID tags and QR codes) are designed. A lookup mechanism is required to associate the identity and the information of the thing. The implementation ranges from an embedded database to a sophisticated resolution service, for instance, the Oracle Object Name Service (ONS).

When considering the IoT as a network of connected smart devices (network-enabled things), the main objective is to invoke the functions of a device across the network. To implement the IoT from this viewpoint, the service-oriented architecture (SOA) addresses most of the fundamental technology. The missing piece is the way to uniquely identify a thing. Two approaches are employed to identify a thing [14]: based on the physical device and based on the functionality of the device. For the former category, the mapping between the identity and the address is straightforward. For the latter category, the endpoint of communication is the service provided by the device. The addressable entity

is the service (i.e., a group of functions). Therefore, a device may provide several services and may be associated with a few identities (referred to as *virtual identities*).

When considering the IoT as an orchestra of several entities (physical object and/or smart devices), the main objective is to manage the entities. Therefore, identity resolution, service discovery, messaging protocols, and functional structure [15] should be determined according to the problem domain.

2.3. **Extensible messaging and presence protocol.** The Extensible Messaging and Presence Protocol (XMPP) [16] is a technology specified by the IETF for real-time communication. It is based on the XML format for exchanging information. One of the features of the XMPP is that it transmits small fragments of XML from one entity to another by an efficient stream-like method. Therefore, the XMPP can be adopted as the high-level communication protocol for a software service.

A typical XMPP network is organized as a federation of servers. Each client is assigned a Jabber ID (JID) that has a format similar to an e-mail address[16]. Presence detection and instant messaging are the two fundamental functions that the XMPP is designed to achieve. An authenticated user can subscribe to another user's presence status. Whenever a presence change occurs, the servers implicitly propagate the change to the related subscribers. The messaging is achieved by one-way *normal* messages or bi-directional *chat* messages. Multi-party communication is supported by using a *groupchat* message (or the more advanced multi-user chat (MUC) extension).

Many extensions (collectively named the XMPP Extension Protocol, XEP) have been defined to enhance the applicability of the XMPP. For example, the service discovery extension [17] provides the necessary service registry mechanism and is available in the majority of the free XMPP server implementations. With the service discovery XEP, the XMPP is suitable for integrating high-level services in a dynamic environment. When small amount of binary data is needed to transfer, the Bits of Binary (BOB) [18] or In-Band Bytestreams (IBB) [19] could be used. For bulk data, using an out-of-band protocol extension, such as Jingle [20], is recommended to efficiently encode and transfer the stream. Support for binary data transfer is important not only for efficient transmission but also for wrapping proprietary products.

According to our development experiences [21, 22, 23, 24, 25], the following useful properties are observed when implementing a service with the XMPP:

- According to the specifications, the service discovery extension of the XMPP is useful for locating a *server component* (a service bound to the server and identified as a subdomain inside the server). When locating a client-based service (bound to an XMPP user account), we have an additional option: we can directly identify the necessary service using its JID. The decision regarding whether to use the discovery mechanism or the JID is dependent on the service resolution method in the application.
- A campus-wide service network is under the control of the IT department. The allowed applications can also be certified. Enforcing a consistent discovery method is not difficult. Sometimes locating a service based on the JID is sufficient. Even if a discovery-based approach is used in the application, we may assume some prior knowledge to reduce the complexity. Although a sophisticated service definition language is applicable to an extensive variety of service descriptions, a set of simple strings is sufficient for identifying a service in this controlled environment.

3. **Framework Design.** In this section, the motivation and goals of the framework is described in Sec. 3.1. This work is an evolution of our previous design. The mapping of

the IoT concepts to the framework is provided in Sec. 3.2. Then, the details of the newly added access control service are described in Sec. 3.3.

3.1. **Revisions of the framework.** The framework proposed in this section is derived from our previous designs. In [21], the project to establish a budget-limited future classroom (e.g., for remote-area schools) is initiated. The XMPP is used as the infrastructure for integrating the projector and the tablet computer in a low-cost interactive whiteboard application. In [22], the framework-layer protocol and the fundamental service is defined. The service-discovery XEP is enforced in the framework. Thus, the design can be used in a dynamic environment. In [24], the framework design is applied to a telecare application for discovering and designing more basic services. The scenario is to integrate the software/hardware components at home. The notification service and schedule service are designed to work with the notification receivers.

Based on a few additional implementation experiences, we found that the design is not enough for the scenario of future classrooms. The point is that the services in a classroom are often connected as peers. The scheme is similar to the IoT. The things work collaboratively and the role of client/server is not obvious. As mentioned in Sec. 2.2, the SOA addresses most of the technology in IoT. It is possible to adapt an SOA-based design to IoT-based. The issue of the adaptation is the management of the things. Firstly, the peer-based service composition relies on distributed service discovery. Because our previous SOA-based design hosts a service on XMPP with discovery XEP, it is almost ready to meet the requirement. Secondly, when the scale of the application is single-classroom or home-based, most of the services are physically protected by the network infrastructure such as the firewall. Therefore, access control can be performed at the application level. However, to make the framework practical in a campus-wide configuration, the service-level access control mechanism should be considered. In the next section, the access control service (ACS) is integrated as part of the framework. Due to the ACS design, the original *Fundamental Service* is deprecated. The XMPP messaging functions, the service-discovery functions, and the lifecycle control functions are moved to the communication module. The revision from the Fundamental Service to the communication module also makes the design cleaner in terms of API encapsulation.

3.2. **Modeling IoT in the framework.** As previously mentioned, the IoT is domain-specific and should be designed based on the applications. A future classroom enables interactive and collaborative applications. A modern classroom contains electronic devices such as a projector, a card reader, an audio-visual device, and/or a personal computer. These items are the physical things that provide services on the network by the controlling software. In addition to the physical devices and network, pure software services are also allowed. They provide the functionality for work flow and data processing. Therefore, a future classroom application integrates several services to facilitate the real-time communication and resource-sharing among the participants and the services.

Because the IoT environment is considered as a set of services, the *digital shadow* concept [14] is employed in modeling the IoT. An IoT application is composed of a few things and the communication channels. A thing may be a physical object or a software entity; it is identified by one or more identities. When a thing refers to a piece of data, such as the description of a product, the identity is the value for looking up the associated data. When a thing provides services on the Internet, it has one or more digital representations, which are the aspects of a thing's functionality and the endpoints of communication to the thing. Each digital representation is associated with an identity. To distinguish the data-oriented identities and the service-oriented identities, we refer the latter as a *virtual identity*. Therefore, the application accomplishes the task by obtaining input

values and identities and exchanging messages among the services for processing the information. To support the configuration of the application, a mechanism for discovering the representations is needed. In addition, a resolving method is required to map a virtual identity to the address of the service.

According to our previous study [22, 24], the XMPP fits the requirements. Each entity in the XMPP network is identified by a JID. The JID format satisfies both the identity and the addressing. Therefore, JID is employed as the virtual identity. The XMPP entity that connects to the network is the software service or the pseudo representation of a smart device. The message encloses a piece of data or a remote procedure invocation request/response. The service description and discovery mechanism is achieved by enabling the service-discovery XEP. The status of a service is announced by the presence. The controlling is accomplished with the command XEP.

3.3. **Access control service.** The XMPP core and extension provide the basic functions to implement a service. To organize a things' orchestra, we need a service-level management framework. Our design is inspired by [15], especially the social-oriented approach. In a campus, the classrooms are located in one or more buildings, and the types of classrooms are limited. Therefore, the management policies may apply to a single classroom, a building of classrooms, a type of classroom, and/or all classrooms. Our design includes the access control services (ACSs) for specifying access policies (as shown in Fig. 2). According to the scope of the policy, the following terms are defined:

Local ACS: specifies the access control policies for a classroom. If a local ACS cannot determine the permission, it forwards the request to the regional ACS and the functional ACS. The responses from the regional ACS and the functional ACS are AND'ed as the result.

Regional ACS: specifies the access control policies for a group of physically related classrooms. If a regional ACS cannot determine the permission, it forwards the request to the global ACS.

Functional ACS: specifies the access control policies for a type of classroom. If a functional ACS cannot determine the permission, it forwards the request to the global ACS.

Global ACS: specifies the access control policies for all classrooms. This service should include a catch-all policy to ensure that it can respond with a default value.

This mechanism consists of a three-level policy chain from the local level (the lowest level), regional and functional level, to the global level (the highest level). It is flexible because we can specify generic rules in a higher level and override them in a lower level. In our reference implementation, we choose the white-list approach. If a grant is not explicitly specified in a lower level ACS, the result is determined by the higher level ACS. The catch-all rule in the global ACS is to reject the access.

Figure 3 shows the relationships of an IoT application and the associated services. In terms of implementation, three types of services are available: (1) normal service, (2) local, functional, and regional ACSs, and (3) global ACS. The modules required to implement the services are shown in Fig. 4. Based on these discussions, a normal service is implemented with the communication module, the ACS module, and the service-specific module. The first two modules are provided by the framework. The communication module contains the functions for basic XMPP communication, service discovery, and service life-cycle control. This module requires the following configuration data: the login credentials and the service descriptions. The ACS module provides the functions for granting or rejecting access to the service. The required configuration parameters

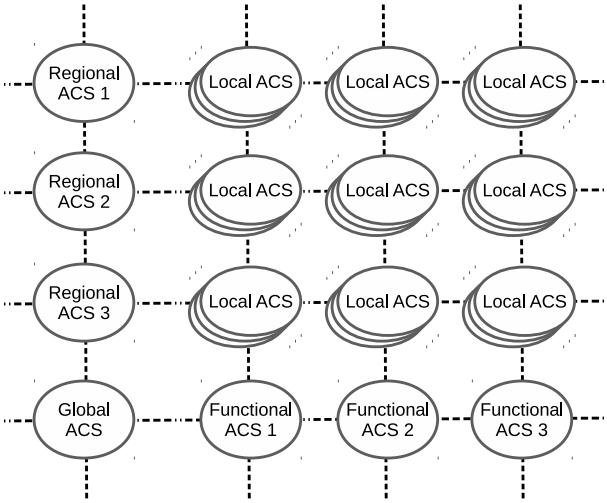


FIGURE 2. Acsess control services on a campus

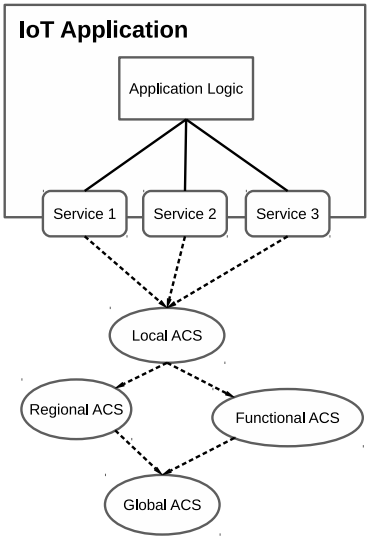
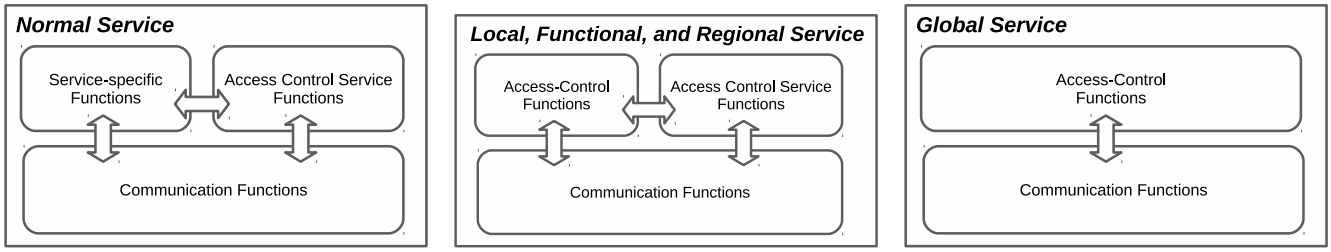


FIGURE 3. Relationships between an IoT application and the services



(a) (b) (c)

FIGURE 4. Modules for implementing different types of services

are the in-service access rules and the identities of the upstream ACS(s). The service-specific module implements the functions of the service, including the initialization of the communication module and the ACS module. Regional and functional ACSs are normal services that implements the access control rules in the service-specific module. For the global ACS, the ACS module is not needed because it is the terminal ACS.

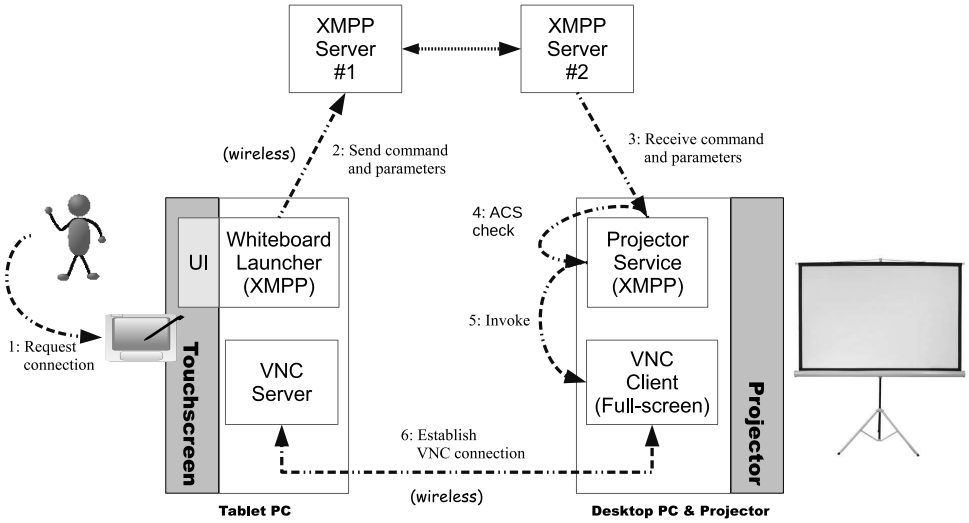


FIGURE 5. The scheme of the digital whiteboard

The ACS module processes the inquiry message and yields the result. The formats of the messages are defined as follows:

- `('acs-check', <appl-user-id>, <service-id>, <loc-id>, <time>)`
- `('acs-result', <grant or reject>)`

The `<appl-user-id>` is the identity of the user (the bare JID is sufficient in most of the cases), `<service-id>` indicates the service entity, `<loc-id>` represents the location of the service, and `<time>` is the wall-clock time when the inquiry is issued.

4. **Application Examples.** In this section, two examples of an application in a future classroom environment are illustrated. A low-cost interactive whiteboard is described in Sec. 4.1. Its multi-location variant application is described in Sec. 4.2.

4.1. **Interactive whiteboard.** The design described in this section is directly motivated by the construction of an interactive whiteboard application [21, 26, 27, 28, 29]. A whiteboard (or blackboard) is an important tool for a teacher to communicate teaching materials to the students in the classroom and prompt the students to pay attention to the location of the teacher while the teacher observes the reaction of the students. Due to the implicit two-way communication induced by a whiteboard, the replacement of a whiteboard with personal displays is not appropriate.

To implement an interactive whiteboard (IWB), an open-source based solution is proposed in [21][22]. The approach is based on devices that are commonly available in a multimedia classroom, such as the controlling computer, the projector, and the wireless network. Using an additional tablet computer, we can construct an effective IWB. The integration scheme is shown in Fig. 5.

In this scheme, the projector and its controlling computer are wrapped as the projector service. The teacher's tablet is the actual device that is used for display and interaction. To export the tablet display to the projector system, the open-source virtual network computing (VNC) solution is used. The application contains the following components:

- The whiteboard launcher, which instantiates the portable digital whiteboard application, locates the required projector service, and sends the commands to the projector service.
- The projector service, which receives the commands for connecting to or disconnecting from the tablet.

- The VNC server program in the tablet system, which can be started at boot time or on-demand, as long as it is available prior to the launch of the portable digital whiteboard application.
- The VNC client program in the projector's controlling computer, which is launched when the projector service is asked to connect to the tablet and terminated when the projector service is asked to detach from the tablet.
- The XMPP server(s), which are required by the tablet launcher and the projector service.

Because the VNC programs and the XMPP server program are developed by third party developers, the whiteboard launcher and the projector service are implemented in the prototype. The whiteboard launcher can be implemented as a simple command issuer program. The whiteboard launcher is a message sender that logs in the network, discovers the projector service, sends the command as a message, waits for the response message, and logs out. The projector service should support the following two commands:

- `('connect', <VNC server address>, <credentials>)`: The whiteboard launcher should supply the tablet's VNC address (the host address and the display number) and the credentials (typically, the pass phrase). When the projector service receives the request, it creates a thread to launch the VNC client with the given parameters. Some of the parameters are preset, such as the full-screen, scaling display, and view-only options. Depending on the result of the established VNC connection, the projector service will respond with one of the following messages:
  - `('ACK', <key>)`: When successfully connected, an identification string is created and returned. This string is used to terminate the VNC session.
  - `('NAK', <reason>)`: When failing to connect, a string that indicates the reason may be returned.
- `('disconnect', <key>)`: The whiteboard launcher should supply the corresponding key. When the projector service receives the command, it checks the key and terminates the VNC session. Similar to the *connect* command, either an *ACK* or *NAK* message is sent to the launcher.

A projector service should announce the necessary information for service discovery. A typical campus contains rooms with projectors. In a controlled environment, we assume that the location of a service is known and that the capabilities can be expressed as simple strings.

- The **service controller** (the owner or the administrator) is represented by the bare JID. This ID is represents an entity that manages a collection of services.
- The location, the capability, and the enumerator are represented by three feature strings. The first two feature strings are straightforward. The third feature string is designed as an extension to enumerate multiple devices of the same model.

Sometimes the service controller is correlated with the location or functionality. The method for selecting the proper granularity of the service controller is an administrative issue. A universal policy for making a "correct" decision has not been developed. Consider the case of 10 classrooms and a maximum of three projector devices. In this case, a single service controller is sufficient for managing all devices.

The scheme shown in Fig. 5 is based on ideal conditions. The scheme does not address the exclusiveness of the projector device and the faults caused by the VNC programs. An enhanced version is implemented to announce the availability of the projector and to handle the crash of a VNC client. The availability of the projector is announced by altering the projector service's presence between *available* and *do not disturb (dnd)*. From an implementation viewpoint, the availability is synchronized with the existence of a VNC

client thread. The service becomes *dnd* when a VNC session is established and becomes *available* when the VNC session is terminated.

To handle an unexpected crash of the VNC client, the projector service should monitor the created VNC client thread. When the thread is not intentionally terminated (with the disconnect command), the projector service should notify the whiteboard launcher and reset the projector's status to *available*. Upon receiving the notification message, the whiteboard launcher decides whether to restart the VNC session. To be able to receive the notification, the whiteboard launcher is implemented as a notification receiver. This implies that the notification service [24] is available and the event `vnc-crash` is registered.

An additional enhancement is to allow the projector service's controller to issue the *disconnect* command. In this case, the key is ignored and the VNC session is terminated. This command is a privileged command for emergency administration. A notification event is published as follows:

- `('publish', 'admterm', <key>)`: The key is supplied as the verification information. The message explicitly specifies that the termination is caused by an administrative reason.

4.2. **A multi-location scenario.** In the interactive whiteboard example, only the projector service is involved in the application. The configuration of the ACS is to verify whether a teacher's tablet is allowed to access the projector during the specified time period. In this section, an application scenario is presented to show the usage of the multi-layer ACS design.

Suppose we have all the access control rules for the regular classes defined in the local ACSs. An invited talk will be given in classroom A. The slides and the audio should be available to all the conference rooms B and C in the same building D at the same time. To support audio playback, an audio sink service is needed in each conference room. An audio sink service receives the control message that specifies the multicast audio source, and starts to playback the audio stream. On the teacher's tablet, the application discovers the projector services and audio sink services, launches the VNC server and the audio encoder, and initiates the controlling messages to the services. The audio encoder is implemented by invoking the open-source VideoLan Client (VLC) to capture the audio from the microphone and to produce the RTP multicast stream. The overall configuration of the application is shown in Fig 6. The invited talk is a temporary event and requires additional facilities outside of classroom A. As mentioned in Sec. 3.3, the reference ACS implementation is white-list based. The access control rule are added in the conference room ACS (the functional ACS) and the building's ACS (the regional ACS) to grant the access from the teacher's tablet to the conference rooms' projectors and audio sinks.

5. **Discussions.** During the experiments, both a local XMPP server (OpenFire) and the Google Talk server performed adequately, which indicates that the service registry could be any XEP-0030-enabled XMPP server. Different projector services based on different VNC clients (TightVNC, RealVNC, SSVNC, and UltraVNC) were also implemented. The four implementations are started concurrently. The VNC implementations could be selected from the launcher's user interface for testing. We discovered that allowing different implementations to coexist is convenient for prototyping the application.

One of the obstacles that restricts the integration of services is the presence of firewalls. A firewall is commonly employed to protect an enterprise Intranet from external attacks. However, a firewall also restricts the communication to a service that is hosted in the Intranet. The firewall issue can be solved at the service level or at the protocol level. The service-level solution may vary, depending on the supported mechanisms in different

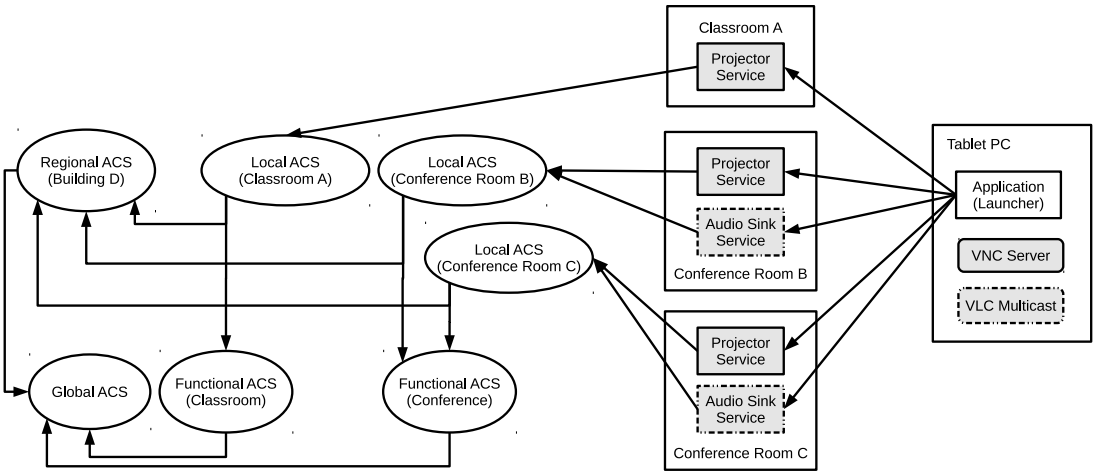


FIGURE 6. The overall configuration of the multi-location scenario

applications. For example, we may construct tunnels among services that use XMPP binary streams. The protocol-level solution could be transparent to the service if the XEP-0124 (BOSH) is enabled. Because BOSH requires additional setup, a BOSH-specific communication module should be developed and be used in the required services.

In the framework, the ACS serves an important role to control the service-level security. Each incoming message should be verified by the ACS chain before it is available to the service-specific module. The ACS traffic is likely to overload a higher-level ACS. To reduce the messages to an upstream ACS, a lower-level ACS can include a cache of the granting results to ensure that the ACS checks are local for the majority of the time. To support the cache mechanism, each response should include the expiration time for cache entry clean-ups.

6. **Conclusions.** In this paper, a service-oriented framework for the future classroom with IoT concept was proposed. According to the properties of the services in a future classroom, we design our framework based on the XMPP. The framework supports asynchronous communication, unicast/multicast/broadcast messaging, a service discovery mechanism, and basic security capabilities.

One of the features of our design is that the first processing step of an incoming message is to verify it against the ACS chain – the local ACS (classroom-based), the regional ACS (building-based), the functional ACS (purpose-based), and the global ACS (campus-based). Each ACS contains in-service access control rules. If it cannot render a decision, it forwards the problem to the upstream ACS(s). We may define general rules in a higher-level ACS and override the rules in a lower-level ACS. This service-level access control structure enhances the protection of the service network.

Two example applications are presented to describe how to integrate a physical device to the service network, and to illustrate how to use out-of-band communication to integrate other protocols. The results demonstrate the effectiveness and flexibility of the framework design.

Acknowledgment. This work was partially supported by the National Science Council, Taiwan, under Grants NSC 102-2221-E-032-066, and by the Ministry of Science and Technology, Taiwan, under Grants MOST 103-2221-E-032-052.

REFERENCES

[1] L. R. Winer and J. Cooperstock, The "intelligent classroom": changing teaching and learning with an evolving technological environment, *Computers & Education*, vol. 38, no. 1-3, pp. 253–266, 2002.

[2] E. Welbourne, *et al.*, Building the internet of things using RFID: The RFID ecosystem experience, *IEEE Internet Computing*, vol. 13, no. 3, pp. 48–55, 2009.

[3] G. Broll, *et al.*, Perci: Pervasive service interaction with the internet of things, *IEEE Internet Computing*, vol. 13, no. 6, pp. 74–81, 2009.

[4] M. Kranz, P. Holleis, and A. Schmidt, Embedded interaction: Interacting with the internet of things, *IEEE Internet Computing*, vol. 14, no. 2, pp. 46–53, 2010.

[5] A. Castellani, *et al.*, Architecture and protocols for the internet of things: A case study, *Proc. IEEE Int'l Conf. Pervasive Computing and Communications Workshops*, pp. 678–683, 2010.

[6] S. Hong, *et al.*, SNAIL: an IP-based wireless sensor network approach to the internet of things, *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010.

[7] Y. Liu and G. Zhou, Key technologies and applications of internet of things, *Proc. Int'l Conf. Intelligent Computation Technology and Automation*, pp. 197–200, 2012.

[8] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services, *IEEE Transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.

[9] L. Tan and N. Wang, Future internet: The internet of things, *Proc. Int'l Conf. Advanced Computer Theory and Engineering*, vol. 5, pp. V5–376—V5–380, 2010.

[10] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view, *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[11] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] J. Gubbi, *et al.*, Internet of things (IoT): A vision, architectural elements, and future directions, *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[13] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, Internet of things: Vision, applications and research challenges, *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[14] A. Sarma and J. Girão, Identities in the future internet of things, *Wireless Personal Communications*, vol. 49, no. 3, pp. 353–363, 2009.

[15] H. Ning and Z. Wang, Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Communications Letters*, vol. 15, no. 4, pp. 461–463, 2011.

[16] P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: The Definitive Guide*, O'Reilly, 2009.

[17] J. Hildebrand, P. Millard, R. Eatmon, and P. Saint-Andre, XEP-0030: Service discovery, *XMPP Standards Foundation, Standards Track*, 2008.

[18] P. Saint-Andre and P. Šimerda, XEP-0231: Bits of binary, *XMPP Standards Foundation, Standards Track*, 2008.

[19] J. Karneges and P. Saint-Andre, XEP-0047: In-band bytestreams, *XMPP Standards Foundation, Standards Track*, 2012.

[20] S. Ludwig, J. Beda, P. Saint-Andre, R. McQueen, S. Egan, and J. Hildebrand, XEP-0166: Jingle, *XMPP Standards Foundation, Standards Track*, 2009.

[21] F. C. Chang and D. K. Chen, An open-source enabled scheme for improved interactive whiteboard, *Proc. Int'l Conf. Ubi-media Computing*, pp. 213–216, 2011.

[22] F. C. Chang and D. K. Chen, The design of an XMPP-based service integration scheme, *Proc. IEEE Int'l Conf. Intelligent Information Hiding and Multimedia Signal Processing*, pp. 33–36, 2011.

[23] F. C. Chang and H. C. Huang, An XMPP based service framework with a telecare application, *Research Notes in Information Science*, vol. 13, pp. 58–62, 2013.

[24] F. C. Chang and H. C. Huang, A framework for prototyping telecare applications, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 1, pp. 61–71, 2014.

[25] D. K. Chen, F. C. Chang, and S. Y. Lin, A location-based context-aware service discovery approach for cycling experience, *Int. J. of Ad Hoc and Ubiquitous Computing*, vol. 16, no. 2, pp. 125–135, 2014.

[26] C. Greiffenhagen, Out of the office into the school: electronic whiteboards for education, *Programming Research Group, Tech. Rep.*, 2000.

[27] H. J. Smith, *et al.*, Interactive whiteboards: boon or bandwagon? a critical review of the literature, *Journal of Computer Assisted Learning*, vol. 21, no. 2, pp. 91–101, 2005.

[28] P. Levy, *Interactive whiteboards in learning and teaching in two sheffield schools — a developmental study*, Master's thesis, Department of Information Studies (DIS), University of Sheffield, 2002.
[29] M. Lee and M. Boyle, The educational effects and implications of the interactive whiteboard strategy of Richardson primary school: a brief review, *Richardson Primary School, Tech. Rep.*, 2003.