An Architecture of Mobile Web 2.0 Context-aware Applications in Ubiquitous Web

I-Ching Hsu
Department of Computer Science and Information Engineering
National Formosa University
64, Wenhua Rd., Huwei Township, Yunlin County 632, Taiwan
hsuic@nfu.edu.tw

Abstract—**The rapid development of the wireless communication technologies, including wireless sensors, intelligent mobile devices, and communication protocols, has led to diverse mobile devices of accessing various context-aware systems. Existing context-aware systems only focus on characterize the situation of an entity to exhibit the advantage of contextual information association. The contextual information can represent semantic implications to provide decidable reasoning services, but it has no mechanism to facilitating the interoperability and reusability among heterogeneous context-aware systems and various mobile devices. This study addresses these issues developing a Multi-layer Context Framework (MCF) that integrates Web 2.0 technologies into context-aware system for supporting ubiquitous mobile environment. The proposed MCF includes context sensor layer, context information layer, context service layer, context representation layer, mobile device layer, and context-aware mobile Web 2.0 application layer. To demonstrate the feasibility of the infrastructure, a Mobile Web 2.0-based Context-aware Attendance Monitoring System (MWCAMS) is implemented with ZigBee devices to provide continuous and context-aware monitoring of employee attendance status.**

Index Terms—**Context-Awareness, Mobile Web 2.0, Ubiquitous Web, Sensor Web**

I. Introduction

Context awareness was first discussed by Schilit and Theimer [1] to develop context-aware systems that allow users to access information and services according to their current context, including location, time, identity, activity, and preferences. In [2], authors have defined context in terms of the situation of an entity. An entity may be a person, device, place, or object that is considered relevant to the interaction between a user and an application. A system is context-aware if it adopts context to provide relevant information and services to the user, where relevancy depends on the user's task [3].

Context-aware systems require an exhaustive volume of communication to ensure the context-aware information exchange between information providers and consumers. As new standards, technologies, protocols, sensors and mobile devices continue to emerge, the main problem of existing context-aware systems is the lack of an uniformly access to cope with the heterogeneous effects, including various context presentation, context information, mobile device constraints, and context middleware [4]. To allow interoperability among the various context-aware systems and mobile devices, a common standard is needed to uniformly access context information provided via a fundamental infrastructure. The Web 2.0 technologies provide a catalytic solution to this problem.

Within the last two to three years, the Internet has greatly changed our way of sharing resources and information. As well known, Web 2.0 is recognized as the next generation of web applications proposed by T. O'Reilly [5]. The main feature of Web 2.0 applications is that they provide a medium for the sharing and exchange of resources [6-8]. These resources, such as Web feed and Web API, allow web developers to take advantage of these resources to enrich their own applications or produce new integrated solutions by integrating resources, which they could not have provided on their own. Many Internet companies have enabled easy access to the web resources that they provide. Anyone can create a new integrated application with these resources. This new way of building applications has also created a new method of thinking of the Internet as a resource. When an application combines resources from different websites to produce a new web application it is called a Web 2.0 Mashup [7].

Mobile Web 2.0 [9, 10] can be regarded as a kind of Web 2.0 Mashup, which integrates Web 2.0 and mobile wireless communication technologies as a basis for supporting pervasive context-aware systems. It enables the development of context-aware module easy to integrate the various Web 2.0 standards to facilitate the sharing and exchange of context-aware information in the mobile internet environment. This study proposes a Multi-layer Context Framework, called MCF, based on Mobile Web 2.0 technologies is presented to cope with the heterogeneous issues of context-aware systems. This implies at least two requirements for the development of the MCF. The first is heterogeneity. It can present a metamodel to integrate Web 2.0 technologies into context-aware applications independently from the context information, context presentation, context service, and context sensor hardware. The second is performance.

doi:10.4304/jsw.6.4.705-715

It can facilitate to develop a generalized context-aware architecture to promote adequate context information transcoding for various mobile clients.

This study makes three main contributions. First, Web 2.0 technologies are adopted as a backbone of the Multi-layer Context Framework (MCF) to facilitating the interoperability and reusability among heterogeneous context-aware systems and various mobile devices. Second, the MCF is adopted to develop the Mobile Web 2.0-based Context-aware Architecture (MWCA), which enables developers to combine context resources that are distributed over the Internet to create new context-aware applications that can satisfy customers' needs. Third, a context-aware application, Mobile Web 2.0-based Context-aware Attendance Monitoring System (MWCAMS), is implemented based on MWCA to provide continuous and context-aware monitoring of employee attendance status. The MWCAMS carries out to integrate three emerging research areas: Mobile Web 2.0, ZigBee wireless network, and Ubiquitous Context-awareness.

This paper is organized as follows. The next section briefly reviews some existing context-aware systems. Section 3 presents a Multi-layer Context Framework (MCF) for context-aware systems development. Section 4 proposes the Mobile Web 2.0-based context-aware architecture based on MCF. In Section 5, we implemented the MWCAMS to demonstrate the feasibility of Mobile Web 2.0-based context-aware architecture. Finally, summary and concluding remarks are included.

II. RELATED WORKS

Context-aware techniques have been widely used in different domains, but still are limited to small scale environments due to the lack of an uniformly retrieve for exchanging context information. Web 2.0 technologies can support and simplify the exchange of context information in large scale and heterogeneous environments, thus enable context-aware systems to utilize various types of context information to adapt their applications to dynamic changes.

The previous researches [11, 12] have focused on location information as an important aspect of context. A variety of context-aware systems have been developed for various application domains [13, 14], such as mobile applications [15-17] , e- learning [18, 19], Web services composition [20-22], and healthcare systems [23-25]. While they are not Web services-based, some of them are distributed systems, and they have introduced and implemented various context-aware techniques. In [10], authors present a context-aware mobile Web 2.0 service architecture that connects user context and community information with the web services.

The main problem of the mentioned above context-aware systems is that it only focuses on associating context information with user activity. However, they ignore the sharing and exchange of context information to facilitate the interoperability and reusability among heterogeneous context-aware systems and various mobile devices. This study addresses these issues developing a Multi-layer Context Framework (MCF), which integrates Mobile Web 2.0 technologies into ubiquitous context-aware environment. The proposed MCF includes context sensor layer, context information layer, context service layer, context representation layer, mobile device layer, and context-aware mobile Web 2.0 application layer.

III. CONTEXT-AWARE APPLICATIONS BASED ON MOBILE WEB 2.0 TECHNOLOGIES

The Web 2.0 mashup is an emerging Web development paradigm that combines functionality from different websites and enables the loose coupling of heterogeneous Web resources. This study proposes a Multi-layer Context Framework for supporting most of the tasks involved in dealing with pervasive context-aware systems development. The Web 2.0 technologies, including Web feed, Web API and Web presentation, can be mapped into context information, context service, and context presentation respectively.

A. Multi-layer Context Framework

This section presents a Multi-layer Context Framework, called MCF, to explain that Web 2.0 technologies are the backbone of context-aware applications in mobile Web environment. This architecture is depicted in Figure 1, which can be represented in six layers: Context Sensor Layer, Context Information Layer, Context Service Layer, Context Presentation Layer, Mobile Device layer, and Context-aware Mobile Web 2.0 Application Layer. The Web 2.0 technologies consist of Web Feed Web API and Web Presentation, and can be mapped into Context Information Layer, Context Service Layer, and Context Presentation Layer of MCF, respectively. The context-aware mobile web 2.0 applications are across the three layers. The MCF provides a flexible infrastructure that context-aware developer can dynamically add, replace, and remove components in each layer. Each layer contains multiple technologies, all of them providing a service suitable to the function of that layer.

All of the layers are involved when sending request from a context-aware client to a context-aware application. Therefore the upper layers have to rely on the lower layers to process the context-aware resources over the ubiquitous Web [26]. Web Feed layer is used to distribute users regarding changes of contents at some context-aware systems. Web API layer is used to facilitate data exchange between context-aware applications and allow the creation of new context-aware applications. The Web Presentation layer provides independence from differences in context information representation by translating the format from application format to a valid markup language for a specific context-aware client device. Everything is Web 2.0 application specific at the Web 2.0 Application layer. This layer provides Web 2.0 application services for end-users.

Figure 2 shows the semantic structure between context-aware application and Web 2.0 Mashups as a UML class diagram. The UML class diagram has as goal to give a graphical overview of the domain concepts and the relations among them. The components of Web 2.0 Mashups include Web Feed, Web API, and Web Presentation. Web Feed is a typical context information resource, while Web API is a typical context service resource. There is dependency relationship, annotated with a "use" stereotype, from context-aware application to Web 2.0 technology. Its semantic indicates that Web 2.0 technology is a core technology of context-aware applications.

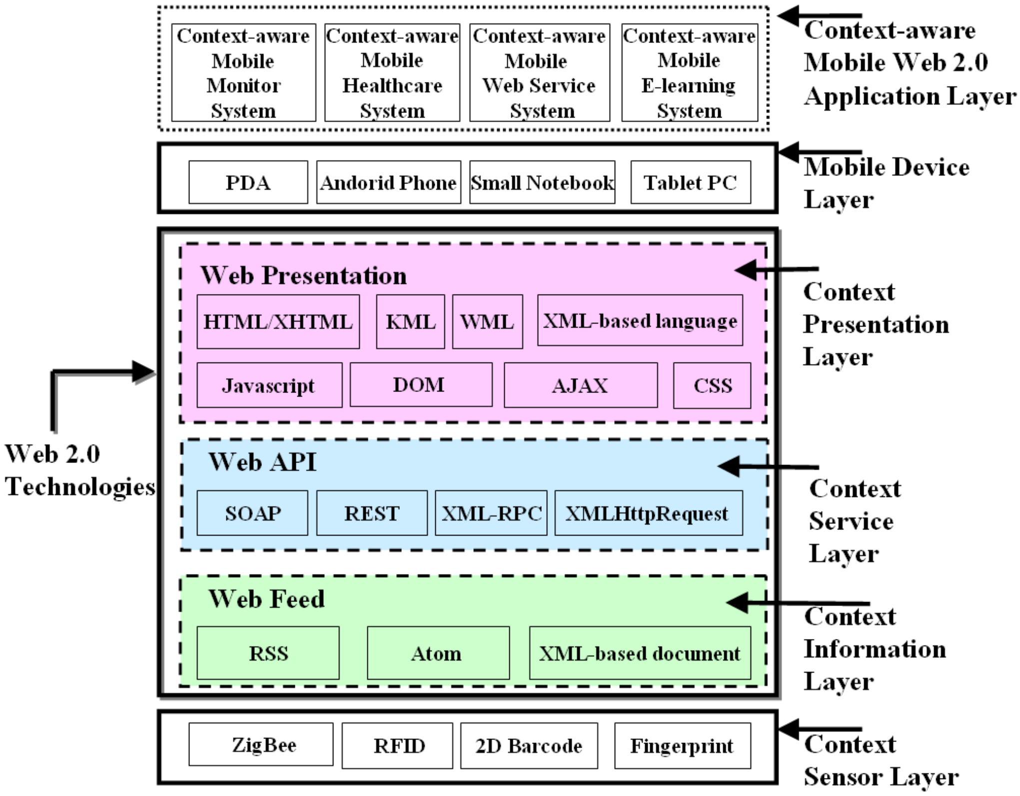

Figure 1. Multi-layer Web 2.0 technologies mapping to context-aware mobile Web

B. Context Information

Context information can be acquired from heterogeneous and distributed sources, including sensors, profile files, and applications. A Web feed contains a structured information source which is written in XML to facilitate the machine-readable. This means that Web feeds can be used to automatically transfer context information from one context-aware system to another, without any human intervention. One major feature of Web 2.0 is to adopt Web feeds to build a more maintainable and cooperative Web. Web feeds allow both context-aware systems to publish frequently updated context information such as Weblog, news headlines, location, time, identity, activity, and preferences. RSS [27] and Atom [28] are currently the two main formats of Web feed. RSS is a family of Web feed formats specified in XML standard. There are three different version of RSS, namely Rich Site Summary, RDF Site Summary and Really Simple Syndication. The Really Simple Syndication (RSS 2.0) is the most widely used. Unlike RSS, Atom is proposed RFC 4287 and is defined with XML schema.

Besides XML-based Web feeds, JSON (JavaScript Object Notation) is another option for data exchange in the various context-aware systems. JSON a lightweight computer data interchange format that a text-based and human-readable format for representing simple data structures. Its main application is in Ajax web application programming, where it serves as an alternative to the traditional XML-based data format. The XML-based data format is more complex than JSON, which is specifically designed as a data interchange format.

Using Web feeds to present the context information have a few advantages. (1) Users can be notified of new context information without needing to visit the context-aware systems. (2) Web feeds provide summaries of context information together with hyperlinks to the full versions of the Web content. (3) Web feeds allow users to pull the context information they are interested in rather than data being pushed to the users.

C. Context Service

Context services are developed to support a reusable solution for context acquisition to simplify the development of context-aware applications. This study argues that it is more appropriate to adopt Web API for context services to allow for the integration of Web services technologies via a rich and open Simple Object Access Protocol (SOAP)/XML-based API. Web API provides more interactive web technologies than was possible in context-aware applications.

Web 2.0 Mashups use Web API technologies to facilitate context information exchange between context-aware applications and allow the creation of new context-aware applications. Most of Web APIs are constructed based on the Web Services architecture. They hide the detailed Web Services protocols from the context-aware developers and make it easier for the developers to use. The main types of Web APIs are: XMLHttpRequest, XML-RPC, SOAP, and REST.

XMLHttpRequest is not a web service technology but a Web API that provides scripting languages to transfer XML or other text data between a client and a server. It is used to communicate asynchronously with a server-side component. XML-RPC [29] is a remote procedure calling employing HTTP as the transportation protocol. It provides a standard for heterogeneous programs to communicate with each other regardless of their implementation language and system platform. Simple Object Access Protocol (SOAP) [30] is an another communications protocol for Web services that emerged immediately after the XML-RPC. It is developed to address some of the limitations of XML-RPC, including only for RPC over HTTP, not easily extendable, and no support WSDL. Representational State Transfer (REST) [31]is a style of software architecture used to describe how Web resources, such as web service, web page, text, database, or website, are defined and addressed. The main advantages of REST-based Web services are lightweight, human readable, and easy to build.

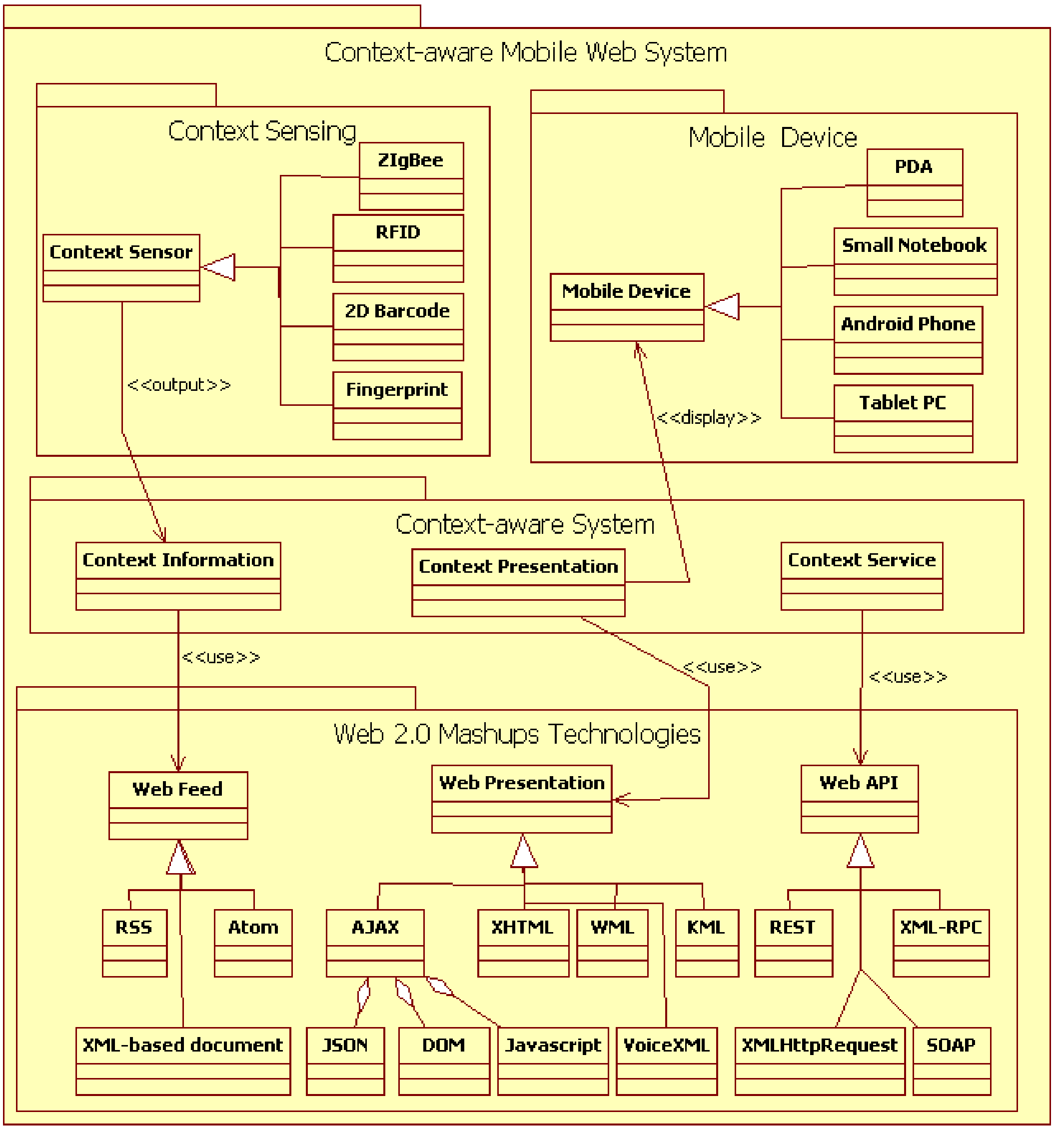

Figure 2. UML diagram for Mobile Web 2.0-based Context-aware architecture

D. Context Presentation

Web presentation technologies are mainly to provide a valid markup language for context presentation on a specific context device. The same context information needs to be rendered differently on different context devices. By considering the physical and performance constraints on a context device, such as screen size, memory size, and connection bandwidth, a significant challenge in context information rendering is automatically converting heterogeneous markup based documents (in formats such as HTML/XHTML, KML, WML and VoiceXML) into the desired XML-based format, which can be understood by a specific context device. The heterogeneous markup-based context information conversion is a type of transcoding [32], which is a technology used to adapt the visual output of context information so that they can be viewed or processed by the multiple devices of context-aware systems.

HTML is widely regarded as the standard publishing language of the Web environment. XHTML (eXtensible HyperText Markup Language) a reformulation of HTML 4.0 based on XML standard. It is intended to be used as a language for content that is both XML-conforming and operates in HTML 4.0 conforming user agents. Based on XML, XHTML is relatively easy to extend new elements or additional element attributes for general user agents. AJAX (Asynchronous JavaScript and XML) is not a new technology or language, but a new framework that combines various existing technologies, including XML, XHTML, XMLHttpRequest [33], and JavaScript. With the promotion of the XML and JavaScript, the AJAX framework becomes a basic ingredient in Web 2.0 applications. It supports a more interactive presentation of context-aware application has enhanced online collaboration and sharing context information among users.

IV. THE MOBILE WEB 2.0-BASED CONTEXT-AWARE ARCHITECTURE

The Mobile Web 2.0-based Context-aware Architecture (MWCA) is developed based on MCF to include the following core components: the Context-aware Platform, Mobile Transfer, Context-aware Services, Web 2.0 API Mashups, and Context Client. The dataflow-oriented Mobile Web 2.0-based context-aware architecture is depicted in Figure 3.

▪ **Context Client**

Context Client is a kind of mobile device, including PDA, smart phone, tablet PC, and small notebook. AJAX is widely used in Web-based devices to enable interactivity to HTML/XHTML pages. It allows a Web-based device to update parts of a HTML/XHTML page asynchronously by communicating with a Context-aware Server using Javascript XMLHttpRequest object.

▪ **Context-aware Platform**

Context-aware Platform serves as a context information broker for accessing of distributed context-aware systems. It uses Web 2.0 mashups technologies to facilitate context information reusability. It contains the following seven main modules:

Manager Interface listens to the client’s request and interacts with other components of the Context-aware Platform.

Transcoding Engine offers a generalized solution to support the transcoding of heterogeneous context information for various mobile devices.

Mobile Device Profile corresponds to user's preferences or device configurations.

Context Information Composer is an intelligent agent that queries the Context Database to finding relevant context information. As the Context Information Composer cannot find available context information, it will be enabled to create a new hybrid context configuration. The composer plays the role as a coordinator, which finds, contracts, and composes appropriate hybrid context information to fulfill the context-aware process.

Context Collector is a tool designed to help Context-aware Services filter and maintain context information in an easy and efficient manner.

Context Database is a XML-based information repository that stores real time context information acquiring form distributed Context-aware Services.

GSM Modem serves as a GSM (Global System for Mobile communications) cell phone to transmit SMS (Short Message Service) over a GSM network just like an ordinary cell phone. GSM modem requires a SIM card to work and can be connected to computer to be used as modem for the SMS sending/receiving. SMSC (SMS Center) is responsible for handling the SMS operations. When the GSM modem generates a SMS message, it first reaches the SMSC. SMSC then routes the message to its destination.

▪ **Web 2.0 API Mashup**

Web 2.0 API Mashup is a website that provides appropriate Web APIs for a particular application purpose. To facilitate Web API retrieval, Web API providers often expose their services through Web protocols such as SOAP, XML-RPC, XMLHttpRequest or REST. Major Internet companies, such as Google, Microsoft, Yahoo, Amazon, and eBay, have published APIs based on web standards that allow you to access their services and context information.

▪ **Context-aware Services**

Context-aware Services provide Web feeds to distributing fresh context information for context-aware applications. The study argues that the classic context-aware system can survive in Web 2.0 trend by publishing Web feeds. Therefore, the classic context-aware system can serve as a Web feed provider.

▪ **Mobile Transfer**

Mobile Transfer serves as a context-aware information broker for accessing of the wireless sensor network. It consists of information collector and information transformer. The information collector receives the context-aware information from sensor network through wireless communication. The information transformer listens to the consumer's request to acquire the context-aware information form information collector, and then converts the context-aware information into a XML-based document to response to consumer.

The MWCA platform offers two different modes, push-based approach and pull-based approach, to send context information to mobile clients. The push-based approach provides timely propagation of up-to-date context information to mobile clients with SMS message. While the pull-based approach allow mobile clients at any time and any place to check for context information updates at MWCA platform. Push-based approach emphasizes real time, and pull-based approach focuses on convenience.

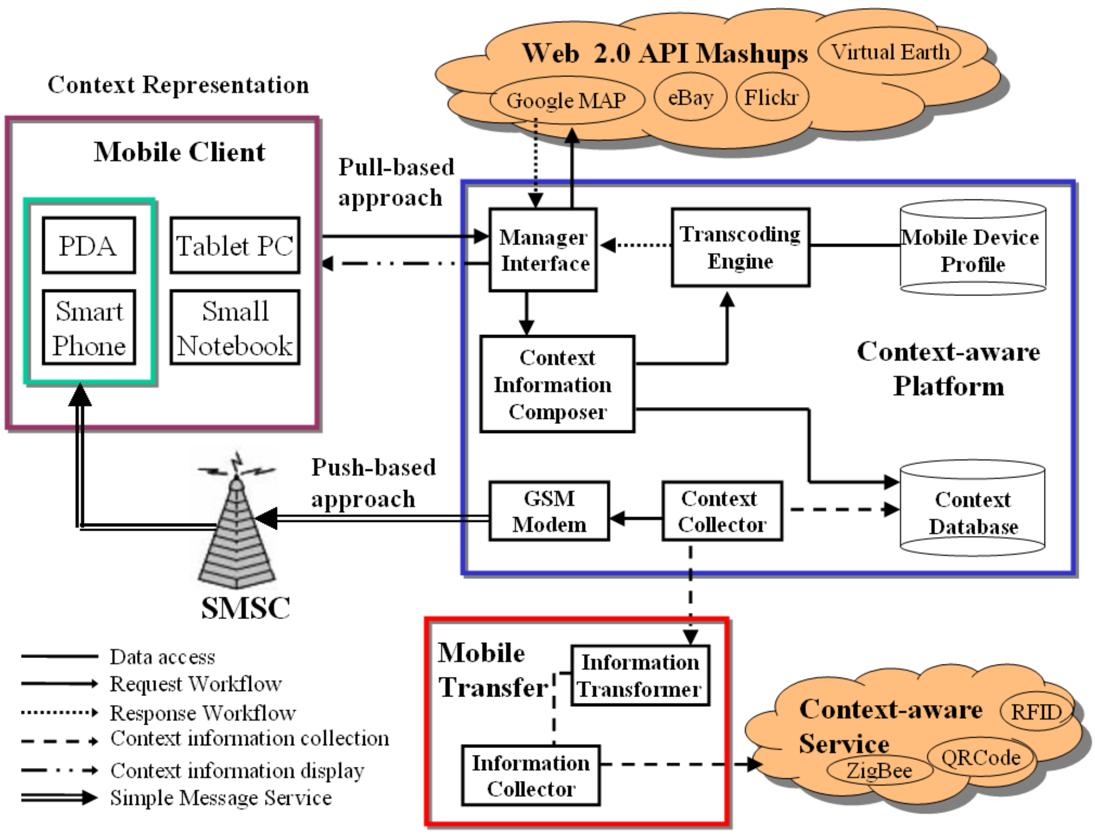

Figure 3. The context-aware platform based on Mobile Web 2.0

V. Implementation and Evaluation

The section developed a Mobile Web 2.0-based Context-aware Attendance Monitor System (MWCAMS) based on the proposed Mobile Web 2.0-based Context-aware Architecture (MWCA). The example is explained in detail, and help the reader better understand the MWCA and how it can be adopted.

This study developed a Mobile Web 2.0-based Context-aware Attendance Monitor System (MWCAMS) based on the proposed Mobile Web 2.0-based Context-aware Architecture (MWCA) to provide continuous and context-aware monitoring of employee attendance status. The dataflow-oriented architecture of MWCAMS is depicted in Figure 4. The devices relevant to MWCAMS are described in Table 1. In the MWCAMS, employee attendance records are collected through the IP-Link 3200, transferred and stored into the ASUS P320 and then provided to the ASUS TS300E5 through Internet. A sample client was built using Java Server Page (JSP) and AJAX to query the ASUS TS300E5, and display the employee attendance information in real-time.

TABLE I. Devices in the MWCAMS

Device	Description
IP-Link 3200 (ZigBee dongle)	It is a ZigBee dongle that supports both the new ZigBee protocol and custom mesh networking solutions. The IP-Link 3200 can use Bluetooth to wireless communicate with Mobile Transfer.
IP-Link Tag (ZigBee Tag)	It is a wireless location tracking tag based on ZigBee network. Each IP-Link Tag has a unique ID ties to an employee for identification.
ASUS P320 (Mobile Transfer)	It is a PDA that serves as a Mobile Transfer for retrieving real time attendance information from ZigBee Sensor Network.
ASUS TS300E5 (Context-aware Platform)	It is a PC server that serves as a Middleware Web Server to provide the function of dynamic employee attendance information generation and transcoding.
WaveCom Q2303A (GSM Modem)	It serves as a GSM cell phone to transmit SMS messages over a GSM network just like an ordinary cell phone. GSM modem requires a SIM card to work and can be connected to computer to be used as modem for the SMS sending/receiving.
HP iPAQ 212 (Mobile Client)	It is a PDA that displays employee attendance information based on XHTML pages.
MSI U100Plus (Mobile Client)	It is a small notebook that displays employee attendance information using RSS reader.

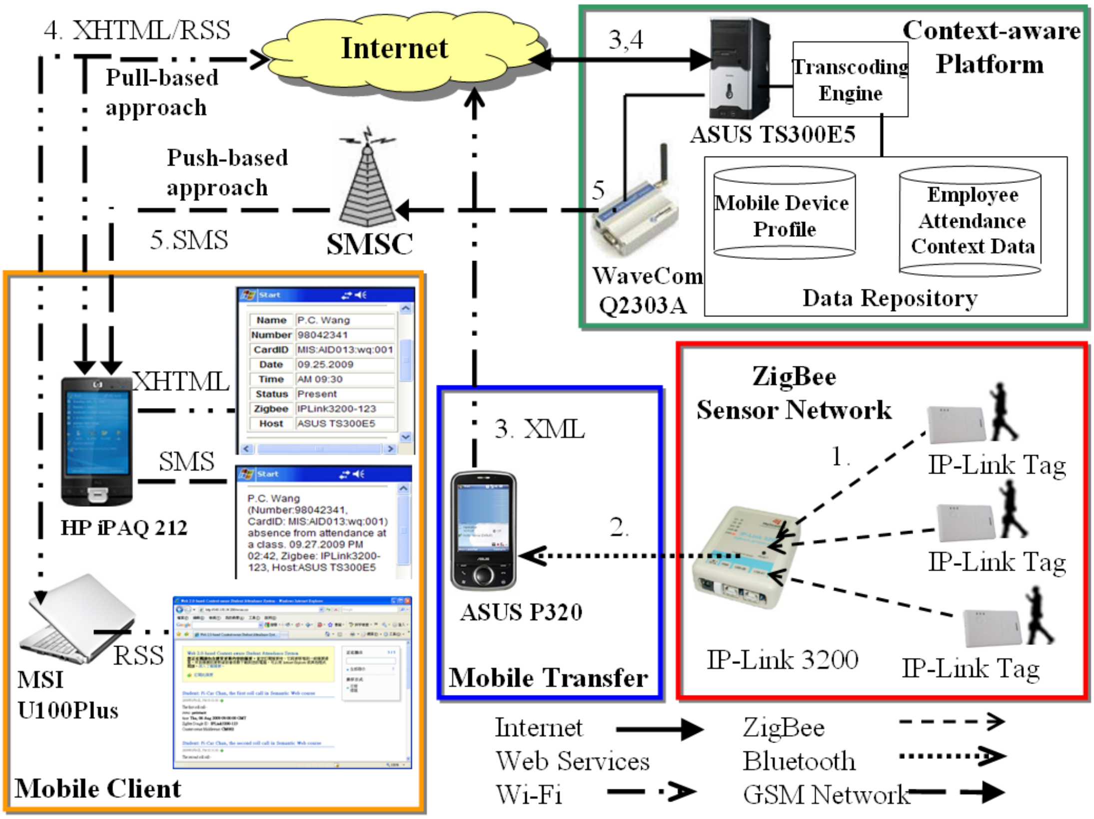

Figure 4. The architecture of MWCAMS

The following steps explain the message flow illustrated in Figure 4:

1. Each employee has an IP-Link Tag associated with a unique ID as the employee identification. IP-Link Tag sends employee attendance information to the IP-Link 3200 through ZigBee network.
2. IP-Link 3200 receives the employee attendance information and then sends them to ASUS P320 by Bluetooth wireless communication.
3. The ASUS P320 accomplishes the following tasks:
    3.1 It parses the employee attendance information to filter the available information.
    3.2 It encodes the employee attendance information to an XML-based document, as shown in Figure 5.
    3.3 It sends the XML-based document to ASUS TS300E.
4. The step is a kind of pull-based approach that users actively read the context information form MWCAMS. The various devices, such as mobile PDA or notebook, can send a request to ASUS TS300E5 with the employee ID to browse the employee attendance records. The XML-based attendance document can be converted to various XML-based documents, such as XHTML (shown as Fig 6.), to display in mobile PDA (HP iPAQ 212).
5. The step is a kind of push-based approach that MWCAMS actively sent context information with SMS message to users.

```
<?xml version="1.0" encoding="utf-8" ?>
<attendance id="20101006">
<employee id="AID02123">
<name>Ha-Yau Wang</name>
</employee>
<department id="MIS002">
<title>Computer Center</title>
<manager>Dar-Wan Hsu</manager>
<room>CMI-100-2W</room>
</course>
<record id="MIS:AID02123:wq:001">
<rollcall>first</rollcall>
<status>presence</status>
<time>Thu, 06 Aug 2010 09:00:00 GMT<time>
<sensor>IPLink3200-123</sensor>
<middleware>ASUS TS300E5</middleware>
</record>
…….
</attendance>
```

Figure 5. Partial code of XML attendance information

```
<?xml version="1.0" encoding="utf-8" ?>
<rss version="2.0" ><channel>
<title> Attendance Monitoring System</title>
<link>http://140.130.35.139/?source=rss</link>
<category>Attendance Monitoring </category>
<generator>MWCAMS</generator>
<item>
 <title>Employee: Ha-Yau Wang, the first roll call
</title>
<link>http://140.130.34.206/2010Cti/201008060001.jsp
</link>
 <pubDate>Thu, 06 Aug 2010 09:01:00 GMT</pubDate>
  <guid>MIS:AID013:wq:001</guid>
  <subject>employee attendance record</subject>
  <description>
  <![CDATA[The first roll call -
   <br>status: <b>presence</b>
    <br>time: <b>Thu, 06 Aug 2010 09:00:00 GMT</b>
    <br>ZigBee Dongle ID : <b> IPLink3200-123</b>
    <br>Context-aware Middleware: <b> ASUS
      TS300E5</b> ]]></description>
</item>….
</channel></rss>
```

Figure 6. Partial code of RSS attendance information

VI. Experimental Results

This section evaluates the Mobile Web 2.0-based Context-aware Architecture (MWCA) for Web 2.0-based context-aware applications against our requirements. The requirements include heterogeneity and performance, which have been mentioned in section 1.

A. Heterogeneous Evaluation

The heterogeneous assessment contains context information independence, context presentation independence, and context service independence. The study adopts MWCA as a generalized architecture of mobile Web 2.0 context-aware applications. Table 2 shows significant comparisons between the Web 2.0 technologies and MCF based on the heterogeneity. The independence of the platform and the hardware allows for a lightweight and simplified evolution of more complex mobile context-aware applications. Furthermore, the constructs in the proposed MCF and MWCA are not specifically designed to match one particular context-aware application. Therefore, they can support the heterogeneity to develop various mobile Web 2.0-based context-aware applications.

TABLE 2. Related Web 2.0 technologies adopted to address the heterogeneous issues

Web 2.0 technology	Service Type	MCF	Description
RSS	Web feed	Context information	The attendance information is described in RSS format to support to mobile phone or notebook RSS reader.
AJAX	Web presentation	Context presentation	A sample client was built using Java Server Page (JSP) and AJAX to query the ASUS TS300E5, and display the employee attendance information in real-time.
SMS Web service	Web API	Context service	The SMS Agent is developed as a Web service that sends an attendance SMS message to mobile phone.
XML	Web feed	Context information	The attendance information is coded to an XML-based document
XHTML	Web presentation	Context presentation	The attendance information is rendered in XHTML for mobile PDA.

B. Performance Evaluation

This section presents a preliminary experiment for evaluating the performance of the context information transcoding and the response time of context presentation. The Transcoding Engine of MWCAMS is a kind of template-based transcoder [34], which adopts some transformation language, such as XSL, XSLT, and CSS, to describe rules for transforming XML-based document. Currently, the Transcoding Engine of MWCAMS has offered three XSLT style sheets, XML2XHTML, XML2RSS, and XML2SMS, to convert XML-based context information to XHTML, RSS, and SMS, respectively. The test scenario contains three context presentations, namely XHHTM, RSS, and SMS, displayed in Figure 4.

The experience conditions are described as follows:

1. Experiments were performed on a 2.66GHz Intel Xeon (Quad-Core) PC with 8G of RAM running Windows 2008.
2. Context information is encoded in XML.
3. The test dataset contains 300 XML-based context information documents.

Table 3 shows the execution times of different context information transcodings.

The 3500 XML-based context information documents undertaken transcoding tests by XML2XHTML, XML2RSS and XML2SMS, for which the average transformation times were 22.1, 20.2 and 19.6 ms, respectively. The experimental results indicated that the XML2XHTML had Experimental results indicate that the XML2XHTML had the slowest transformation time for each context information document followed by

XML2RSS and XML2SMS. These experimental results indicate that the major factor in transformation time is the number of template modules in an XSLT style sheet.

TABLE 3. RESULTS OF THE TRANSCODING EXPERIMENTS

Context Presentation	XSLT Style Sheet	Template Modules in the XSLT Style Sheet	Average Transcoding Time
XHTML	XML2XHTML	26	22.1 ms/per document
RSS	XML2RSS	15	20.2 ms/per document
SMS	XML2SMS	7	19.6 ms/per document

TABLE 4. RESULTS OF THE PULL-BASED EXPERIMENTS

Request Number	XHTML (ms)	RSS (ms)
10	297	293
100	302	297
200	305	303
300	310	307
400	314	312
500	1461	1357

TABLE 5. RESULTS OF THE PUSH-BASED EXPERIMENTS

Mobile Device Number	SMS Web Service Execution Time(ms)	Receive SMS Time (ms)	GSM Communication (ms) = Receive SMS Time- SMS Web Service Execution Time
1	32	3132	3100
5	36	3160	3124
10	40	3196	3156
20	51	3235	3184

The response time of MWCAMS can be evaluated along two fronts. Firstly, the response time of pull-based approach, which underlies the context information transcoding and Internet transmission, evaluates how the response time of the MWCAMS increases when the number of the request increases. Secondly, the execution time of push-based approach, which underlies SMS Web Service and GSM communication, evaluates the increasing trend of the execution time of the MWCAMS when the number of mobile device increases.

The pull-based response time contains the context information transcoding time and Internet transmission time. This experiment evaluated the MWCAMS as a context-aware broker that processed from 100 to 500 requests. Table 4 shows the average values obtained for pull-based response time. Notably, the threshold value for request number was about 400. When request number was lower than the threshold value, average response time was about 0.3 seconds. Conversely, when the number of concurrent requests increased beyond this threshold number, response time increased very rapidly because both I/O consumption and contextual information transcoding performance increased substantially.

The push-based execution time contains the SMS Web service execution time and GSM communication time. Due to the limited number of mobile devices and the cost of sending SMS, the actual test of the experiment can only send SMS to 20 different mobile devices. The experience results of the push-based average execution time are shown in Table 5. We observed that the average execution time of SMS Web service and the number of mobile device increase in equal proportion. The average time of GSM communication is about 3.1 second. Additionally, this tests show that the execution time of SMS Web Service takes a very limited percentage of the receive SMS time (about 1%). It is worth noting that a significant variation on push-based execution time result form variable GSM communication.

VII. Conclusion and Future Work

This study proposes a Multi-layer Context Framework (MCF) that consists of context sensor layer, context information layer, context service layer, context representation layer, mobile device layer, and context-aware mobile Web 2.0 application layer. We argue to adopt Mobile Web 2.0 technologies as the backbone of context-aware systems to facilitate the sharing and exchange of context-aware resources including context-aware information and context-aware service. From the Mobile Web 2.0-based context-aware attendance monitoring system carried out in the paper, we demonstrated that Mobile Web 2.0 technologies serve as a core technology for Web 2.0 Mashups to enhance added value of traditional context-aware applications. Additionally, this study also realizes how Mobile Web 2.0 technologies can be integrated into context-aware applications.

Context-awareness is a very important feature for ubiquitous computing to enhance current context-aware applications by finding right context information and right context services in the right place at the right time.
The main limitation of Mobile Web 2.0 is that it lacks the computer-interpretability to support knowledge representation for context-aware applications. One future work is to investigate how to integrate Semantic Web technologies [35] into Web 2.0-based context information to facilitate the implementation of intelligent Mobile Web 2.0-based context-aware applications. In recent years, cloud Computing is an emerging computing paradigm to facilitate the rapid building new generation of information systems via the Internet. Another topic for future work would be to extend MCF to support the Mobile Web 2.0-based context-aware applications for cloud computing environment.

Acknowledgment

The author would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. 98-2221-E-150-072- and NSC 99-2221-E-150-069-.

References

[1] Schilit BN and Theimer MM, *Disseminating active map information to mobile hosts.* IEEE Network 1994; 8(5):22-32.
[2] Dey AK and Abowd GD, *Toward a better understanding of context and context-awareness.* 1999.
[3] Dey AK, Abowd GD, and Salber D, *A Context-Based Infrastructure for Smart Environments*, in *Proceedings of the First International Workshop on Managing Interaction in Smart Environments*. 1999: Dublin, Ireland. p. 114-128.
[4] Wagner M, *Context as a service*, in *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing*. 2010, ACM. p. 489-492
[5] O'Reilly T. *What Is Web 2.0*. 2005 [cited 12 July 2010]; Available from: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.
[6] Yakovlev IV, *Web 2.0: Is It Evolutionary or Revolutionary?* IEEE IT Professional 2007; 9(6):43-45.
[7] Murugesan S, *Understanding Web 2.0.* IEEE IT Professional, 2007; 9(4):34-41.
[8] KNIGHTS M, *Web 2.0.* Communications Engineer 2007; 5(1):30-35.
[9] Jonghong Jeon SL, *Technical Trends of Mobile Web 2.0: What Next?*, in *World. Wide Web (WWW) Conference*. 2008: Beijing, China.
[10] Timo Koskela NK, Otso Kassinen, Juuso Ohtonen, Mika Ylianttila, *Towards Context-Aware Mobile Web 2.0 Service Architecture*, in *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. 2007: Papeete, French Polynesia.
[11] Martín S, et al., *A context-aware application based on ubiquitous location*, in *The 2nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. 2008. p. 83-88.
[12] Vijayalakshmi M and Kannan A, *Proactive location-based context aware services using agents.* International Journal of Mobile Communications 2009; 7(2):232-252.
[13] Boukerche A, et al., *Wireless sensor and actor networks context interpretation for the emergency preparedness class of applications.* Computer Communications 2007; 30(3):2593-2602.
[14] Malik NA, Javed MY, and Mahmud U, *Estimating User Preferences by Managing Contextual History in Context Aware Systems.* JOURNAL OF SOFTWARE 2009; 4(6):571-576.
[15] Driver C and Clarke S, *An application framework for mobile, context-aware trails.* Pervasive and Mobile Computing 2008; 4(5):719-736.
[16] Falchuk B, Loeb S, and Panagos T, *Seamless mobile context: Towards high speed context-aware event-based middleware*, in *5th IEEE Consumer Communications and Networking Conference*. 2008. p. 89-90.
[17] J. M-RF, Vanderdonckt J, and Arteaga JM, *Context-aware generation of user interface containers for mobile devices*, in *Proceedings of the Mexican International Conference on Computer Science*. 2008. p. 63-72.
[18] Zheng Y and Yano Y, *A framework of context-awareness support for peer recommendation in the e-learning context.* British Journal of Educational Technology 2007; 38(2):197-210.
[19] Yu Z, et al., *Ontology-based semantic recommendation for context-aware e-learning*, in *Ubiquitous Intelligence and Computing - 4th International Conference*. 2007, Lecture Notes in Computer Science p. 898-907.

[20] Niu W, et al., *A DDL-based model for web service composition in context-aware environment*, in *Proceedings of the IEEE International Conference on Web Services*. 2008. p. 787-788.
[21] Han W, Shi X, and Chen R, *Process-context aware matchmaking for web service composition.* Journal of Network and Computer Applications 2008; 31(4):559-576.
[22] Lopes AL and Botelho LM, *Context-aware semantic web service discovery*, in *9th International Conference on Enterprise Information Systems*. 2008. p. 231-234.
[23] Koufi V, Malamateniou F, and Vassilacopoulos G, *A mediation framework for the implementation of context-aware access control in pervasive grid-based healthcare systems*, in *Advances in Grid and Pervasive Computing - 4th International Conference*. 2009, Lecture Notes in Computer Science p. 281-292.
[24] Song J-G, Kim S, and Park G-C, *Towards a mobile and context-aware technology based healthcare system*, in *Proceedings of the Frontiers in the Convergence of Bioscience and Information Technologies*. 2007. p. 591-594.
[25] Jung JY and Lee JW, *ZigBee device design and implementation for context-aware U-healthcare system*, in *Second International Conference on Systems and Networks Communications*. 2007.
[26] Sanchez-Loro X, et al., *Ubiquitous web access: Collaborative optimization and dynamic content negotiation*, in *2008 International Conference on Multimedia and Ubiquitous Engineering*. 2008.
[27] Winer D. *RSS 2.0 Specification*. 2003 [cited 16 July 2010; Available from: http://cyber.law.harvard.edu/rss/rss.html.
[28] Nottingham M. *The Atom Syndication Format*. 2005 [cited 11 July 2008]; Available from: http://www.atomenabled.org/developers/syndication/atom-format-spec.php.
[29] Winer D. *XML-RPC*. 2003 [cited 16 July 2010]; Available from: http://www.xmlrpc.com/spec.
[30] Mitra N and Lafon Y. *SOAP Version 1.2 Part 0: Primer (Second Edition)*. 2007 [cited 16 July 2010]; Available from: http://www.w3.org/TR/2007/REC-soap12-part0-20070427/.
[31] Fielding RT. *Architectural Styles and the Design of Network-based Software Architectures*. 2002 [cited 16 July 2010]; Available from: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.
[32] Hsu I-C, Chi L-P, and Bor S-S, *A platform for transcoding heterogeneous markup documents using ontology-based metadata.* Journal of Network and Computer Applications 2009; 32(3):616–629.
[33] Kesteren Av. *The XMLHttpRequest Object*. 2008 [cited 12 July 2010]; Available from: http://www.w3.org/TR/XMLHttpRequest/.
[34] Hsu I-C, Chi L-P, and Bor S-S, *A platform for transcoding heterogeneous markup documents using ontology-based metadata.* Journal of Network and Computer Applications 2009; 32(3):616-629.
[35] Roa-Valverde AJ and Aldana-Montes JF, *Towards a Scalable Infrastructure for Semantic Web Services Execution.* JOURNAL OF SOFTWARE, 2009; 4(8):821-832.

I-Ching Hsu received a Ph.D. degree in the computer science department at the National Chung-Hsing University of Taiwan in 2007. Dr. Hsu has been working at Chung-Shan Institute of Science & Technology in the area of information engineering technologies since 1991. He is currently an assistant professor of Computer Science and Information Engineering at National Formosa University, Taiwan. He has participated and directed projects in the area of XML, Semantic Web and Internet application development. His current research aims at the creation and study of Cloud Computing, Semantic Web, Web 2.0 and Software Engineering.