Journal of Mobile Multimedia, Vol. 1, No.1 (2005) 062-090

DYNAMIC MULTIMEDIA TEMPLATES FOR USERS OF WIRELESS PERVASIVE COMPUTING DEVICES*

EDOUARD MA POON

National Chiao Tung University
edouard@csie.nctu.edu.tw

DENG-JYI CHEN

National Chiao Tung University
djchen@csie.nctu.edu.tw

Received March 11, 2005

The advent of wireless computing has created a myriad of opportunity for developing mobile applications, but it has also created several challenges for application software developers because of the limited computing resources offered by these devices. Porting multimedia application software from traditional desktop computers to pervasive computing devices, in particular, has become a big challenge. Both academic and industrial sectors are investing great attention and effort to overcome these issues. In an effort to address this problem, we present the use of dynamic web-based multimedia templates. In a mobile environment where time and resources are scarce and user productivity is particularly demanding, templates minimize the complexity of creating presentations by offering assistance and flexibility for mobile e-learners to create and customize them directly from their devices in an efficient and effective way. Application examples based on the proposed dynamic web-based multimedia templates are constructed and used to demonstrate the applicability and feasibility on pervasive computing devices. Also, an assessment of the proposed approach is given to address some application challenges that exist in a pervasive computing environment.

Key words: Multimedia templates, Pervasive computing, e-learning, XML

1 Introduction

Pervasive computing extends the traditional computing infrastructure to a new class of devices, mobilizing itself beyond the desktop PC, enabling delivering of on-demand information wherever, whenever and on any device a user prefers. It is about accessing information, applications, and services — solutions that enable anywhere, any time, access to people, information, and processes over any network [2].

Pervasive computing is about three things [3]. First, it is concerned about how people view mobile computing devices and use them within their environments to perform tasks. Second, it is concerned about the way applications are created and deployed to enable such tasks to be performed. And third, it concerns the environment and how it is enhanced by the emergence and ubiquity of new information and functionality.

Recently, pervasive computing has been maturing from its origins as an academic research area to a technical and commercial reality [3, 9]. Recent hardware developments and advances in location sensors, wireless communications infrastructures and global networking [9] have made this possible.

* An earlier version of this paper has been presented at the IEEE Sixth International Symposium on Multimedia Software Engineering in Miami, FL. December 13 – 15, 2004 [1]

To complement this, we have seen an increase in the number of mobile users [2] as access becomes pervasive. Their ease of use concerns are driving new forms of interaction, the infrastructure is extending to new devices, new devices are demanding access to information-rich applications over high-speed networks and applications are becoming increasingly connected. The diversity of these technologies coming together has contributed towards the vision of pervasive computing – anywhere, anytime data access on any device [5, 10]. An overview of the vision of pervasive computing is depicted in Figure 1 [5].

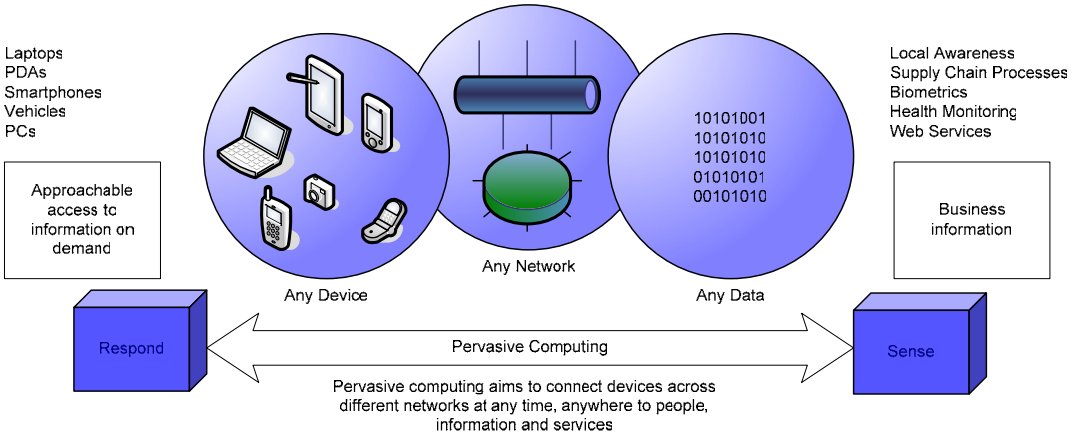

Figure 1. Overview of pervasive computing [5]

An increasing amount of electronic information takes place in the form of multimedia – a mixture of animation, images, video, audio and text. As users become more reliant on pervasive computing devices, there will be a growing need to bring multimedia information to these devices [5, 10, 11]. These advances have opened several new challenges, opportunities and perspectives for developing and interacting with pervasive computing applications that allow these devices to effectively access, store and process multimedia information [2, 3, 9].

It is also becoming obvious that the nature of interactions between users and computers must evolve [13]. The way mobile computing devices and applications are developed, deployed, managed and used today does not meet the expectations of the user community and falls short of the potential for pervasive computing [4]. In the realm of pervasive computing, every aspect of development creates a challenge that is intermittently linked with other challenges that realize the potential and vision of pervasive computing.

This paper focuses primarily on presenting a dynamic web-based multimedia templates approach using XML technologies for customization and how it might be effective in adapting in a pervasive computing environment. Through this, we attempt to overcome constrained computing resources offered by pervasive computing devices when porting multimedia application software from desktop counterparts. We believe that templates will minimize the complexity of creating presentations by offering assistance and flexibility for mobile users to create and customize them directly from their devices in an efficient and effective way. In addition, we identify specific application challenges amongst others in a computing environment that contributes towards the vision of pervasive computing and attempt to resolve them.

2 Related Work

2.1 Challenges for Pervasive Computing and Mobile Users

The proliferation of heterogeneous devices over recent years has spurred the need for increasingly complex applications that demand access to information-rich resources. While such diversity has led to different vendors pursuing different standards, the challenges for pervasive computing remain, essentially the same. These challenges are divided into several categories [2, 9, 14], namely:

- Mobility challenges;
- Device challenges;
- Usability challenges;
- Business challenges; and
- Application challenges.

Mobility Challenges

Mobility challenges associated with pervasive computing encompass a range of issues that are associated with the nomadic user of pervasive computing devices. These issues are largely hardware related and depend much on the state of the wireless environment, for example, the maintenance of connections as devices move between areas of differing network connectivity, and the handling of network disconnections [13]. While protocols for wireless networking do handle some of the problems associated with mobility, such as routing and handovers, there are some problems that cannot be solved at the network level as they require knowledge of application semantics. Some important mobility challenges include [7]: connection state, authentication, voice and data access, device management, scalability and services.

Device Challenges

The heterogeneity in computing systems will not disappear in the future, but instead increase as the range of computing devices widens. Heterogeneous devices will be required to interact seamlessly, despite the varying differences in hardware and software capabilities. This will require an infrastructure that maintains knowledge of device characteristics and capabilities and manages the integration of devices into a coherent system that enables arbitrary interactions [7, 13]. Some of the more specific challenges include: unique devices, varying programming models and a wide range of target environments.

Usability Challenges

Pervasive computing presents an increasingly difficult challenge when developing and deploying pervasive computing applications. Many analysts argue that usability is the number one barrier to the successful deployment of pervasive computing applications [21]. Without a simple, intuitive and efficient user experience, organizations may end up with huge investments in applications that mobile workers refuse to use. The heterogeneity and the particular characteristics of mobile users make the interaction challenges extremely complex. Demand for ubiquitous access will create a requirement for universal interfaces [12] such as MoDAL [22].

Business Challenges

Taking advantage of the pervasive computing opportunity means leveraging non-traditional, embedded computing technologies – both wired and wireless – to enable, integrate and extend e-business

opportunities and new applications [24]. Improving productivity, developing and applying better cost-management strategies, finding new markets, adding value to existing customer relationships and maintaining competitive advantage in an ever-changing marketplace are all part of the business challenges. While the business challenges aren't new, circumstances have changed dramatically. One misconception about migrating to pervasive devices as simply screen scraping or content transformation [14], the challenge lies in reconfiguring the business logic necessary to process the workflow of the application. There are several ways in which pervasive computing can meet the business needs in a mobile environment [13, 24]. There exist several opportunities both in terms of customer and enterprise perspectives.

Application Challenges

In the area of service platforms in a pervasive computing environment, there are currently studies on adaptation, context awareness, smart environments, scalability, integration and interoperability, invisibility and so forth [9, 10, 12, 24, 25], however, on the specific challenges for pervasive computing applications, the following issues are relevant [5, 10, 21, 25]:

- **Automation**. The idea of *doing more by doing less* is achieved through three things [25]: new technologies should be brought into our lives, not vice versa; new technologies must increase human productivity; and integrating technologies so that they become ubiquitous and require minimal intervention. A thin client approach supports this vision – by minimizing the client's functionality and providing support within the network at servers [16, 26].
- **Content aggregation**. In a pervasive computing environment that enables on-demand information, content aggregation offers aggregating of disparate data dynamically from different sources or services and converting it to a single presentation for the users' pervasive computing device.
- **Customization and personalization**. Customization and personalization solve individual user preferences adds value to the overall user experience.
- **Multi-device capability**. An impedance mismatch is apparent between devices in terms of capability and support. Supporting existing and future devices is a challenge for any pervasive computing application. Applications should be able to handle multi-device profiles.
- **Web content and custom applications**. This provides rich context information that is gathered from a wide range or sources, interpreted, and disseminated in a scalable fashion to end users [12].
- **Integration of services**. Integration across heterogeneous protocols and services should be seamless and unnoticeable to the user.

2.2 Mobilizing a Successful Application

While pervasive computing devices will almost never achieve the power of their desktop counterparts, they cannot be compared directly because their environment of use, purpose of use, and approach is altogether different. We approach our problem by exploiting these limitations and taking advantage of alternatives to achieve the reasonable results from a pervasive computing perspective.

There have already been several successful commercially deployed pervasive computing applications, such as at National Express [27], The Waterbury Police Department [28], Lotte Sam Kang [29], Kudos Restaurant [30] and Enning GmbH [31].

We identified three specific criteria necessary for the development of successful pervasive computing applications within a particular computing environment; a successful pervasive computing application is considered as one that:

- Addresses specific application challenges found relevant to the purpose of the application, supports the vision of pervasive computing and enhances the user experience
- Identifies the characteristics of users that will use such an application and take these into account in the development of the application; and
- Considers device limitations by working around them to achieve desired outcomes

2.3 Research Scope

The pervasive computing environment extends beyond the traditional networked environment to utilize wireless technologies such as IEEE 802.11, CDMA, GPRS, GSM and so forth. Such an environment addresses the goal of providing "anytime, anywhere" information access by decoupling users from devices and viewing applications as entities that perform tasks on behalf of users. There are a number of ongoing projects in this area that focus on holistic pervasive computing environments, including PIMA [3], Carnegie-Mellon University's Aura project [32], The University of Washington's Portolano project [33], HP's CoolTown project [34], MIT's Project Oxygen [35], The University of California at Berkeley's Endeavour project [36] and IBM's Websphere Anyplace project [37].

Most of these projects focus their efforts on creating a platform or environment in which pervasive computing applications might adapt and excel. Other efforts include infrastructure issues, dynamic service discovery, device independent models and so forth.

Due to the extensive and complicated process of setting up a pervasive computing environment, the proposed system does not focus on the network or hardware infrastructure, but rather, from an application-centric perspective and assume such an environment.

Specific Application Challenges

Earlier, we identified the application challenges that we found necessary to the development and deployment of successful pervasive computing applications. We focus our efforts on these challenges:

1. Automation;
2. Content aggregation;
3. Customization and personalization;
4. Multi-device capability;
5. Web content and custom applications; and
6. Integration of services.

The proposed system will address each of these application challenges in an attempt to support the vision of pervasive computing from an application-centric perspective. Each of these challenges will be discussed and critically evaluated later to identify particular contributions and limitations of the proposed system.

Mobile Users' Characteristics

In a pervasive computing environment, great importance is placed on the user, how users perform tasks [4] and how results are achieved. We define our target users have the following characteristics:

- They are constantly on the move;
- They have little time and resources at their disposal to perform tasks; and
- They need to perform tasks with the least amount of time, effort and skill.

The proposed system focuses on how the users perform tasks and enhance the user experience rather than on other aspects associated with usability by taking into account the user characteristics.

Pervasive Computing Device Constraints and Limitations

We also consider the device constraints and limitations to maximize user efficiency while maintaining a reasonable computing performance. We address the following device limitations:

- Screen size, resolution and colour depth
- Processing power and battery power
- Other unique device capabilities

In summary, this study will attempt to address each of the specific application challenges, the context of use and the constraints and limitations posed by the devices. By taking these issues into account, we establish a pervasive computing environment in which we believe dynamic multimedia web-based templates will necessarily be effective for mobile users.

3 The Design and Implementation of Dynamic Multimedia Web-Based Templates

3.1 Dynamic Multimedia Web-Based Templates

We define web-based multimedia presentation templates as having a specific set of criteria that describe the structure of the template, the semantic relationships between elements, rules for customization and device profile information. The purpose of these templates is to provide flexibility and opportunity for end users to create customized multimedia presentations from templates. Dynamically assembled templates have several advantages over static-based templates, specifically in terms of managing, integrating, scaling and communicating information to constituents in real-time which is particularly beneficial in a pervasive computing environment that enables on-demand information. By implementing dynamic templates we can address different device specifications and user requests at run-time without having to create several presentations of a single template to support different devices or possible user requests.

Approaching Templates

We propose using templates as a method of assisting users and addressing device limitations. Templates are particularly useful because they provide a starting point and create options for the user to create powerful multimedia presentations without needing much effort, knowledge or skills about multimedia presentations. Templates have the following properties:

- They are dynamic;
- They are web-based; and
- They support multimedia

Templates are dynamic because it was found that dynamically assembled templates are better suited for content aggregation, scalability, and better device support. They are web-based so they are

able to reach a larger device audience through standard formats such as HTML. And finally, they support multimedia to enrich and enhance the user experience.

Static vs. Dynamic Templates and Device Independence

Templates exist either statically or dynamically. Static templates are configured manually whereas dynamic templates are configured at run-time. Static templates are particularly useful if the presentation does not depend on on-demand information at run-time. A comparison of static and dynamic templates is summarized in Table 1.

Table 1. A comparison of static and dynamic templates

Criteria	Static Templates	Dynamic Templates
Supports real-time, on-demand content delivery	No	Yes, adaptable to real-time, on-demand content delivery
Supports heterogeneous pervasive computing devices	No	Yes, device profiles can be stored and loaded dynamically depending on the device
Supports customization and personalization	No	Yes, use of style sheets offers customization opportunities
Supports content aggregation	No	Yes
Applicability towards proposed system	Applicable, but difficult to manage	Applicable and scalable to support new devices
Rendering issues	Fast, no rendering involved	Slower than static templates

Device heterogeneity poses presentation formatting problems. A presentation created for a large-screen device, should, theoretically be displayed and interacted with a small-screen device because they have the same mark-up [38]. However, this is not always the case because of the small form factor issues. Currently there are several attempts [38, 39, 40, 41] to handle device profiling. Device profiling is the process whereby certain specifications of the device are stored in a repository and called when a device connects so that the content is formatted correctly for that specific device. Ideally, content authors would need to create only one version of their Web content for delivery on multiple devices. During the delivery and rendering process, the transformation would create a presentation to match the delivery device's capabilities. Device profiling is still in its infancy, and like ISAPI [39] and MMIT [42], are platform dependent. Cocoon and HP's CC/PP [41], are open standard attempts and both make extensive use of XML and XSL for device customization, but are complex to implement.

When developing for pervasive computing devices, one must ask some key questions [38]:

- How can we express a Web application independent of the delivery device?
- How can we adapt device-independent applications to suit delivery device capabilities?
- How can authors retain some control over the final presentation of their content?

Device profiling and device independence are important, because creating separate versions for each kind of device is both economically and administratively impractical. However, there are currently no standards supporting this, despite efforts to create one. We chose XML and XSL because they are two open standard technologies that can be configured dynamically and particularly suited to handling the translation and transformation process respectively. XML is simple and lightweight for passing data. It separates content from business logic and presentation and supports content

aggregation. XSL is ideal for transforming from multiple data outputs to a single source for different devices. In addition, for web-based multimedia presentation templates, we found XSL extremely robust and an ideal technology for holding information on the criteria of the structure of the template, the semantic relationships between elements and rules for customization.

3.2 Dynamic Multimedia Web-Based Template Creating and Its Translation and Transformation Process

We used the BestWise Multimedia Visual Authoring Tool [6] to create dynamic multimedia templates. There are several benefits associated with this, as users will be able to create customized templates without prior knowledge or understanding of how the templates are coded and created. This supports our vision of minimizing and hiding the complexity of the system to the user. In this section, we discuss the translation and transformation process – two processes that deal with the construction and delivery of the templates. These processes are depicted in Figure 2.

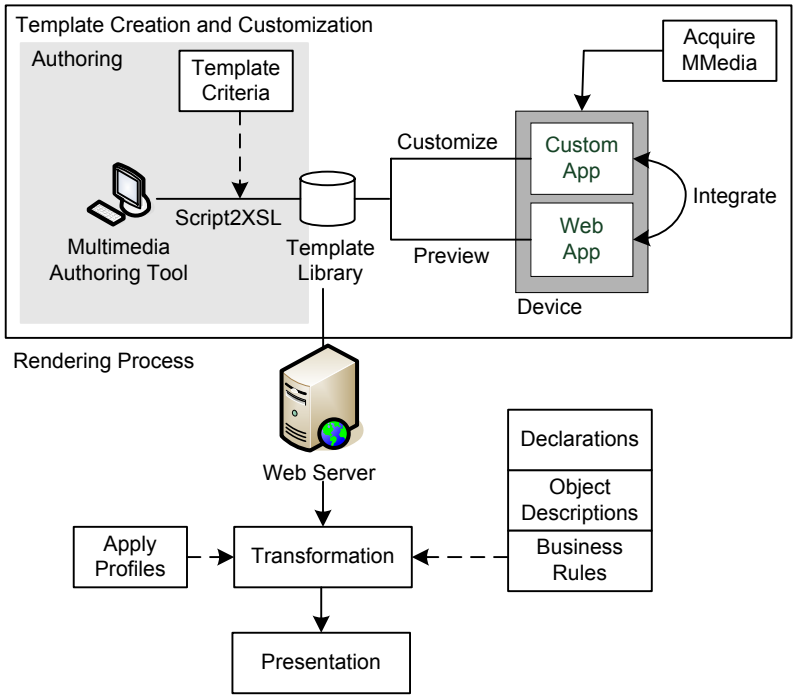

Figure 2. The translation and transformation process

To integrate the proposed system with the Multimedia Visual Authoring Tool, we modified the translator to support the conversion from the Authoring Tool's multimedia script file to XSL.

Figure 3 shows how the translation process interacts with the Template Management Service (discussed in Chapter 4). The output script file (depicted in Figure 4) contains general information about the template such as typical scene information, title, background image, animation effects, background music, background colour, dimensions and so forth. Multimedia elements are listed as actors, actors have a specific type (animation, images, video, audio and text) as well as several attributes related to the type, for example, text actors contain font size, font style, font encoding, font

effect attributes and so on. This script file is parsed and translated to support XML and XSL tags. They are then re-defined, by replacing actual multimedia elements created in the Authoring Tool with anchors. Anchors are defined as placeholders for customization or replacement. In addition, the process embeds the structure of the template, the semantic relationships between multimedia elements and rules for customization.

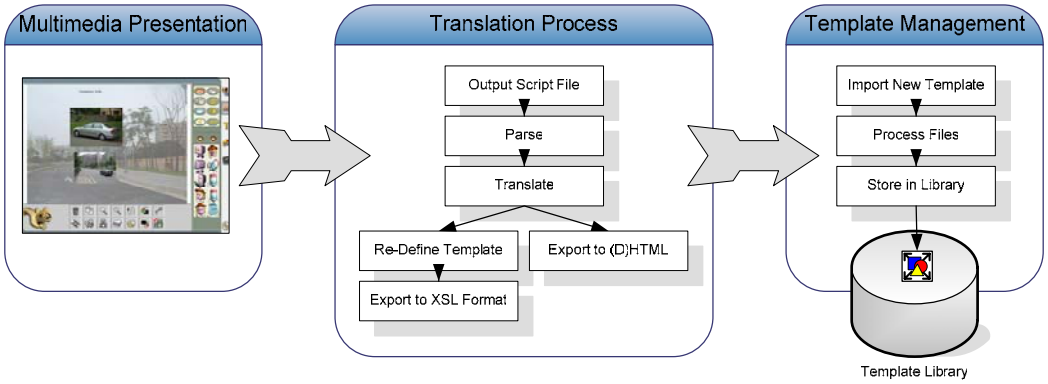

Figure 3. The translation process (Script2XSL)

```
[SystemInfo]
  ScriptVersion = "2.0"
  Application = "Multimedia Authoring Tool"
[End_SystemInfo]

[SCENEINFO]
  Title = "MobileCampus"
  BackgroundImage = "@Sc550.jpg"
  BackgroundMusic = ""
  PreludeEffect = 0
  PlayMode = 2
  Width = 640
  Height = 460
  TextureImage = ""
  BackgroundColor = 0
[END_SCENEINFO]

[CASTDEFINE]
[CAST] MCText
Begin
  {...}
End

[CAST] McMovie
Begin
  {...}
End

[CAST] MCAnim
Begin
  {...}
End
[END_CASTDEFINE]
```

Figure 4. Sample multimedia script file

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:template match="usertemplate/template">
              {…}
   <!-- image anchor definition -->
   <xsl:value-of select="imagelocation"/>
              {…}
   <!-- unique id -->
   <xsl:value-of select="@id"/
              {…}
   <!-- text anchors -->
   <xsl:value-of select="registration"/>
   <xsl:value-of select="date"/>
   <xsl:value-of select="time"/>
   <xsl:value-of select="location"/>
   <xsl:value-of select="comments"/>
   <xsl:value-of select="officername"/>
   <xsl:value-of select="officerphone"/>
   <xsl:value-of select="officeremail"/>
              {…}
   <!-- video anchors (AVI, WMV) -->
   <xsl:value-of select="videolocation1"/>
   <xsl:value-of select="videolocation2"/>
              {…}
   <!-- sound anchor (WAV) -->
   <xsl:value-of select="voicelocation"/>
              {…}
  </xsl:template>
</xsl:stylesheet>
```

Figure 5. Sample XSL blueprint

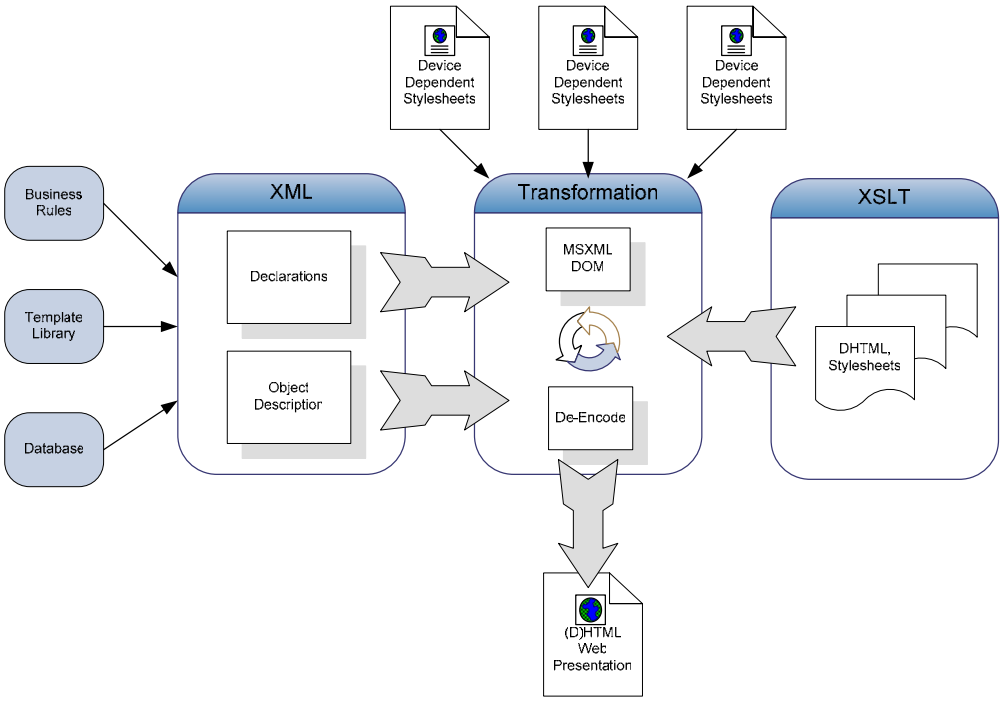

Figure 6. The transformation or content aggregation process

The result of the translation process is an XSL template that is specifically designed to support the proposed system's template format. Figure 5 depicts a blueprint of the resultant template.

The Transformation Process

The transformation process or content aggregation process describes how the data, content, layout and style are dynamically aggregated at run-time from disparate data sources and rendered into a single web presentation.

As depicted in Figure 6, the XML contains information such as the business rules, information about the templates and media. These are transformed using the XSL template and CSS (using the Mobile CSS 1.0 standard) containing device profile information and presented on the Web as HTML.

4 Application Example: MobileCampus

MobileCampus is a traffic incident management system that demonstrates the use of templates and how they might assist mobile users and enhance the user experience while considering device limitations. It is integrated with the BestWise Multimedia Visual Authoring Tool [6] mentioned earlier, an authoring tool for creating multimedia templates.

Business Situation

In a university, parking is often an ongoing day-to-day problem because the number of vehicles often exceeds the number of parking places available. In the case of conferences or exhibitions, the problem usually gets worse. Often, vehicles are found to be in the wrong place at the wrong time and officers are not equipped to deal with the situation immediately. One problem is that owners are unable to be contacted at that time causing disruptions in traffic and instability to the traffic situation on campus. Another problem is, even if an officer can produce a ticket and leave it on the car, the owner may only be notified once he or she returns to his or her car late in the evening. In addition, in the case of campus speeding, officers are often in poor positions to deal with such situations. MobileCampus utilizes the university wireless environment by equipping campus officers with mobile devices that interact in this environment and capture necessary information about the traffic violation, for example, parking and speeding on campus, and uploading this information to a database and making the violation available to violators online. By having a central repository of vehicle registration numbers and owner's contact details, campus officers can, at anytime and anywhere access such information and inform owners of their violations in real time. In addition, we demonstrate how templates can be used to assist the officer perform tasks efficiently and effectively.

MobileCampus is divided into six services that operate dependently and independently of one another:

- Multimedia Capturing Service
- Template Management Service
- File Transfer Service
- Template Creation Service
- Content Management Service
- Internet Service (Web and E-mail)

The Multimedia Capturing Service resides on the end users' client device and incorporates the capturing of multimedia. This service communicates with the File Transfer Service to transfer files and

templates between the server and the client device. The File Transfer Service connects to the Template Management Service (and the template library), which handles the template selections, importing of templates and storing of media files from the client device. The Content Management Service is activated when the end user requests to view a presentation. It dynamically assembles the content, data, layout and style to a single presentation using the Internet Service which in turn renders the presentation for the client device. Figure 7 depicts a system overview of MobileCampus and how each of the system's services relates to each other.

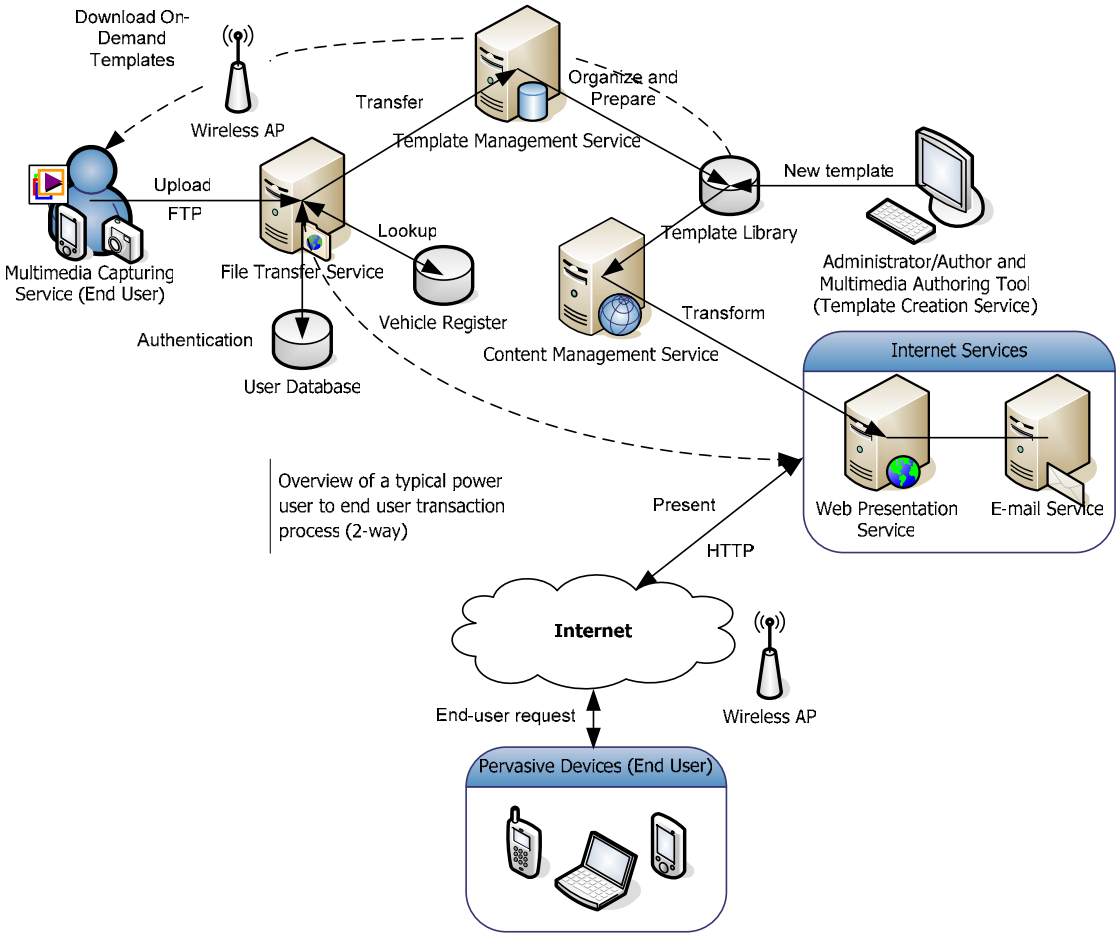

Figure 7. MobileCampus system overview

Figure 8 presents the system architecture of the proposed system mapped from the system overview.

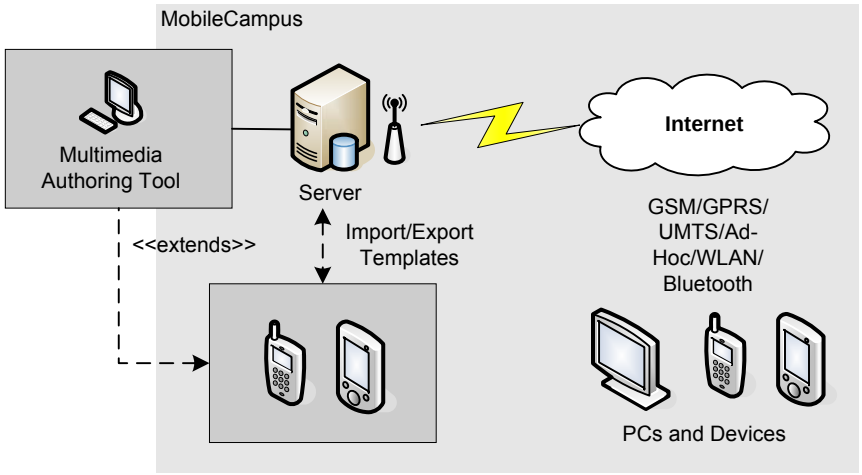

Figure 8. MobileCampus system architecture

4.1. The Multimedia Capturing Service

The Multimedia Capturing Service is client-oriented and works on the pervasive computing device. This service facilitates the capturing of multimedia such as the acquiring of audio, images, text and video on the PDA and compresses them for optimal bandwidth transfer. Figure 9 depicts the process of acquiring multimedia. For this service, the device needs a camera attached in order to acquire images and video. The built-in microphone facilitates the voice recordings and stylus input for text.

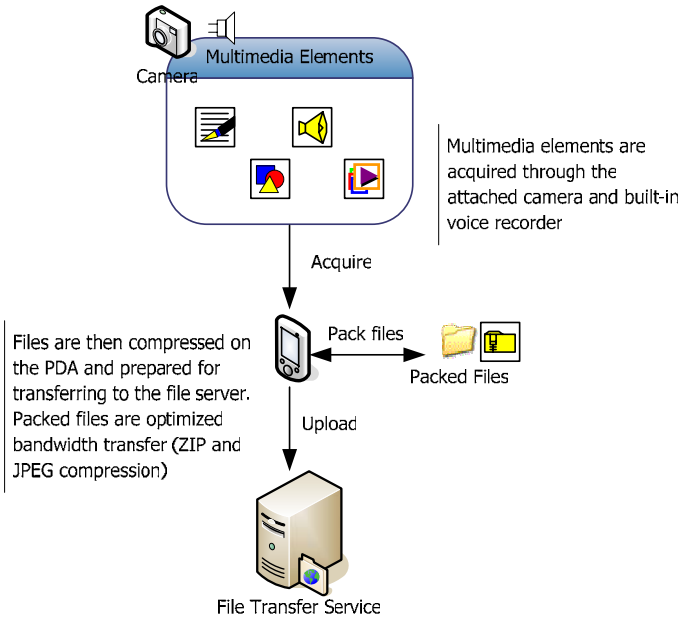

Figure 9. The Multimedia Capturing Service

Figure 10 depicts the screenshots of the Multimedia Capturing Service. Its simplistic interface makes it very convenient for users to capture multimedia.

Figure 10. Multimedia Capturing Service screenshots

Figure 11 depicts the hardware components of the device used by the end user. The entire set includes the handheld device [8], the expansion pack and the Veo PhotoTraveler [7], used for capturing multimedia. The built-in voice recorder in the handheld device facilitates the acquiring of audio.

Figure 11. Hardware device components

4.2. The Template Management Service

The Template Management Service is called by the File Transfer Service when the files are transferred from the client to server. The purpose of the Template Management Service is to organize and store the media in the template and media database as depicted in Figure 13. It separates the content, data, layout and style. This information is stored in XML (Figure 12) that describes the business logic and disparate data information for each transaction.

```
<!DOCTYPE usertemplate [
        <!ELEMENT usertemplate (template)*>
        <!ELEMENT template (registration, time, date, location,
         comments, officername, officeremail, officerphone, imagelocation,
        videolocation1, videolocation2, voicelocation, templatechoice)>
                <!ATTLIST template id CDATA #REQUIRED>
                <!ELEMENT registration (#PCDATA)>
                <!ELEMENT time (#PCDATA)>
                <!ELEMENT date (#PCDATA)>
                <!ELEMENT location (#PCDATA)>
                <!ELEMENT comments (#PCDATA)>
                <!ELEMENT officername (#PCDATA)>
                <!ELEMENT officeremail (#PCDATA)>
                <!ELEMENT officerphone (#PCDATA)>
                <!ELEMENT imagelocation (#PCDATA)>
                <!ELEMENT videolocation1 (#PCDATA)>
                <!ELEMENT videolocation2 (#PCDATA)>
                <!ELEMENT voicelocation (#PCDATA)>
                <!ELEMENT templatechoice (#PCDATA)>
]>
```

Figure 12. The XML Data Type Declaration

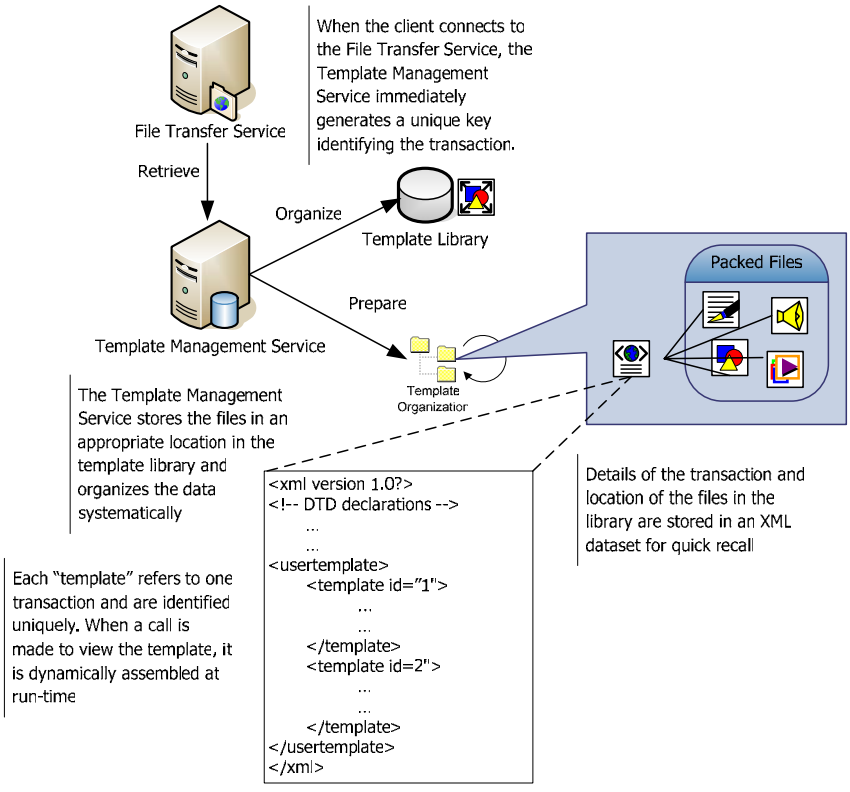

Figure 13. The Template Management Service

In addition, the Template Management Service encodes video into popular formats such as MPEG-4 [46] and WMV for optimal playback on devices.

The Template Management Service also handles the importing of new templates created by the Multimedia Visual Authoring Tool [6] into the template and media library.

4.3. The File Transfer Service

The File Transfer Service operates on both the client and server sides over the FTP protocol. The File Transfer Service can handle multiple clients and deals primarily with the user authentication and the transferring of files to and from the server. It also connects to the e-mail server when necessary, for example, if the owner of a vehicle is registered in the database, the owner is e-mailed immediately and notified of the violation. At the same time, the File Transfer Service connects to the Template Management Service to store necessary information about the transaction and organize the transferred files on the server.

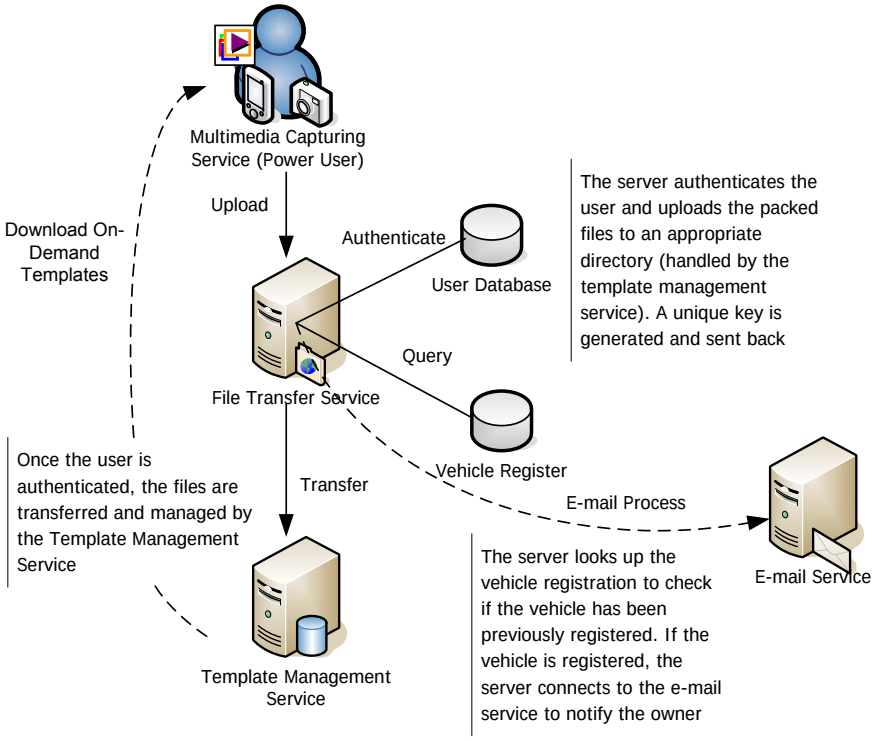

Figure 14. The File Transfer Service

In particular, the File Transfer Service is called in the following two situations (Figure 15):

- When the user wishes to transfer acquired multimedia to the server
- When the user wishes to access on-demand templates and preview them

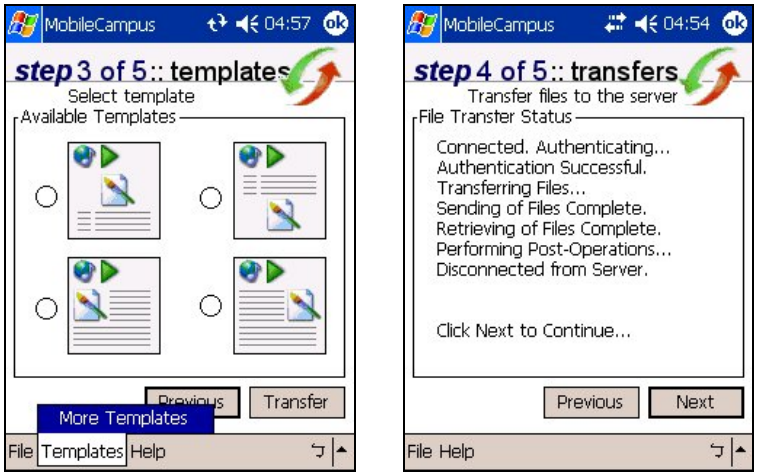

Figure 15. File Transfer Service screenshots

Figure 16 depicts a screenshot of the transactions on the server side.

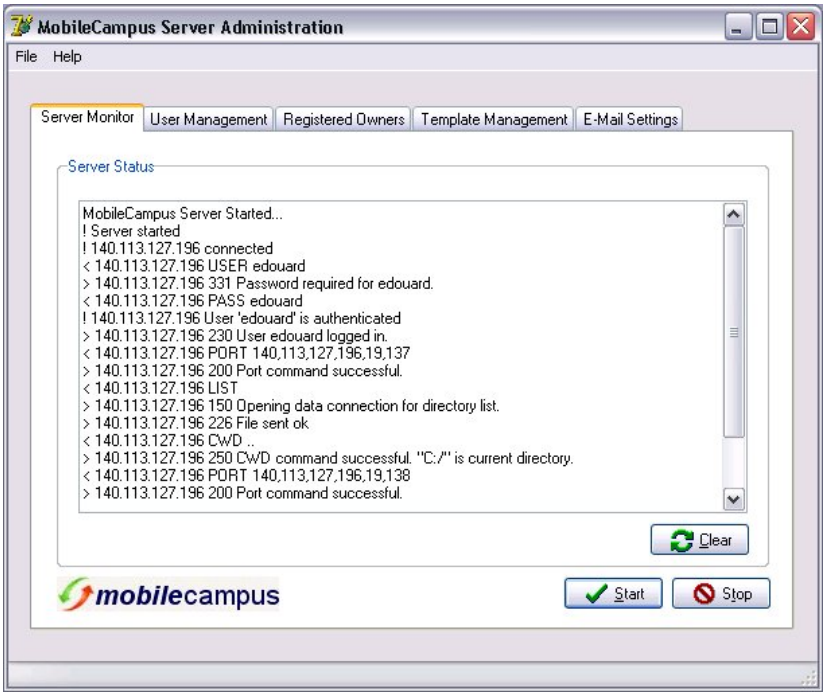

Figure 16. File Transfer Service Server screenshots

4.4. The Template Creation Service

The Template Creation Service is directly configured under the Multimedia Visual Authoring Tool. It handles the creation of templates and the translation process from the template specification to XML and XSL for support on pervasive computing devices. While the Multimedia Visual Authoring Tool has powerful capabilities to create entire multimedia presentations, it is also effective for creating basic

templates that can be exported into various formats such as SCORM, XMT-Ω, Java, XMT, and, in this case XSL.

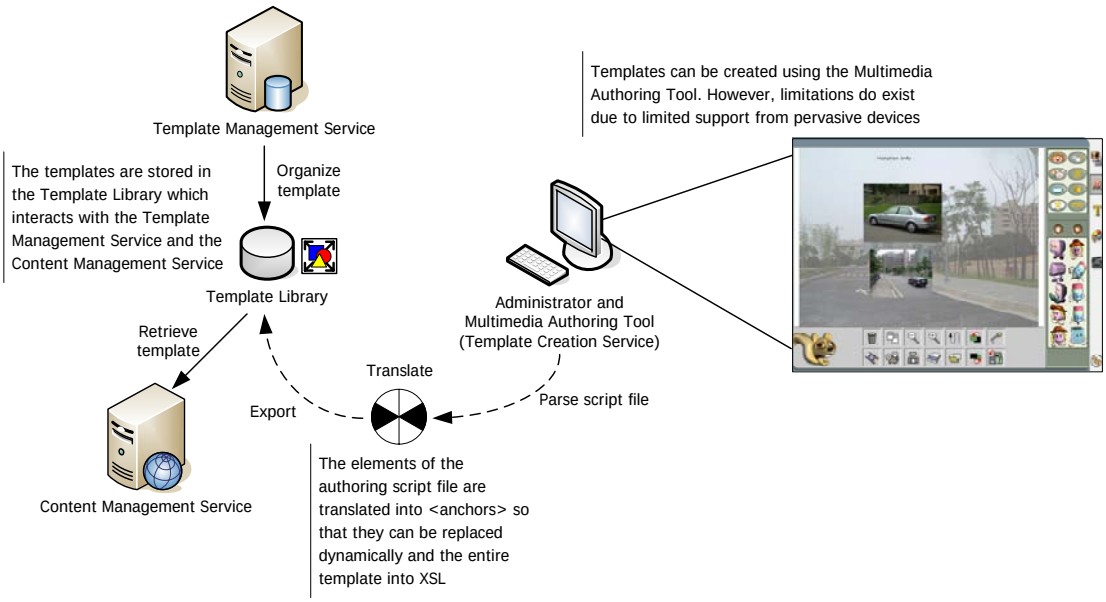

Figure 17. The Template Creation Service

Figure 17 illustrates how the Multimedia Visual Authoring Tool exports a script file that is parsed and translated into XML and XSL format which is in turn imported into the template library for later use.

4.5. The Content Management Service

The Content Management Service is called when a user requests to view the presentation. It dynamically assembles the content, layout, style and data and aggregates the content to present a coherent web presentation to the end user.

Figure 18 illustrates the process of gathering information from disparate data sources and dynamically assembling them at run-time and aggregating the content and rendering it for presentation to the end user.

4.6. The Internet Service

The Internet Service consists of both the web presentation service and e-mail service. It works at the presentation layer and renders the final presentation format for end users. The web server acts as a portal and calls the content management service when processing a user request. The file transfer service utilizes the e-mail server when needed to communicate with end users.

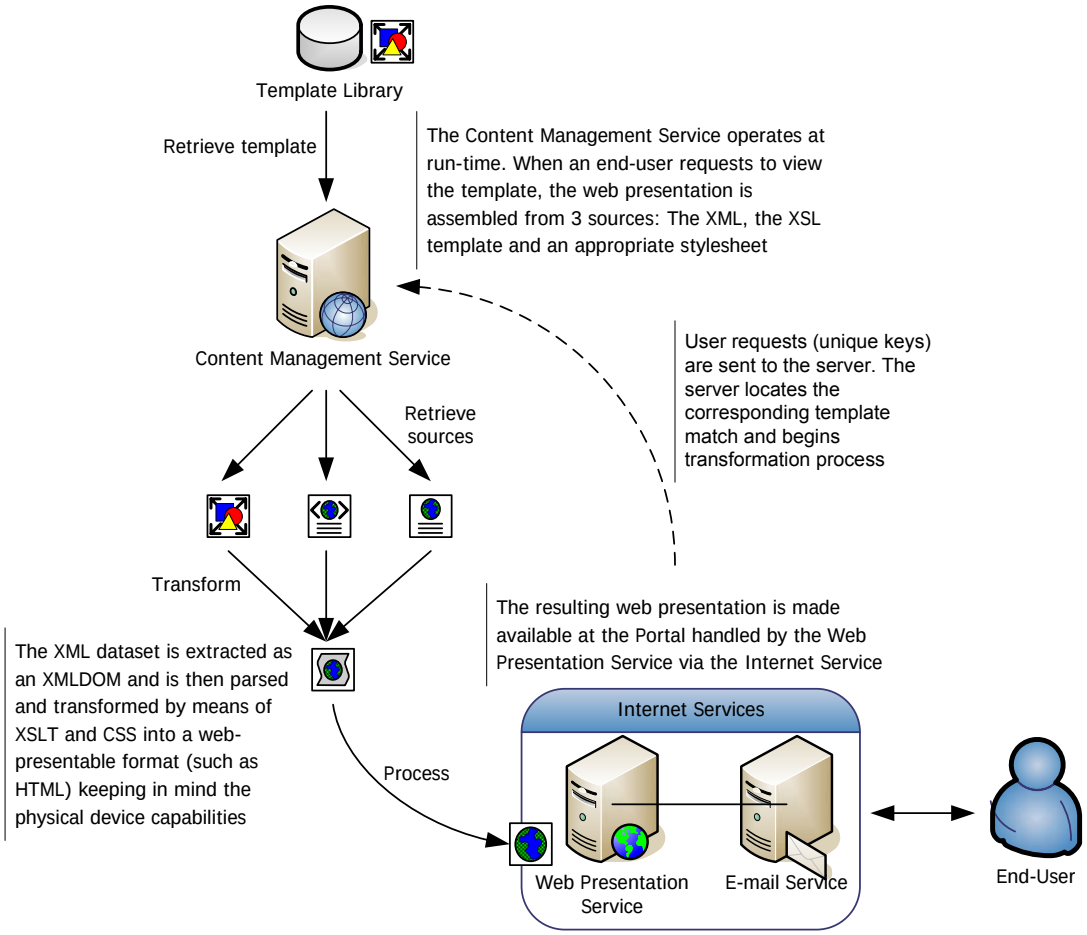

Figure 18. The Content Management Service

4.7. Other System Functionality

The proposed system contains other general functionality used to manage the system. These include:

- **User Management**. User management contains campus officer information as well as authentication information needed to access templates and transfer files.
- **Registered Owner Management**. This is the database containing registered vehicle owner information. If the vehicle's owner is registered in the database, an immediate e-mail notification is sent to the owner to inform him or her that their vehicle has received a violation notice.
- **Template Management**. This functionality deals with importing templates from the Multimedia Visual Authoring Tool into the template and media database. It is made available to end users to download and preview.
- **E-Mail Settings**. This functionality contains the relevant settings for outgoing e-mail.

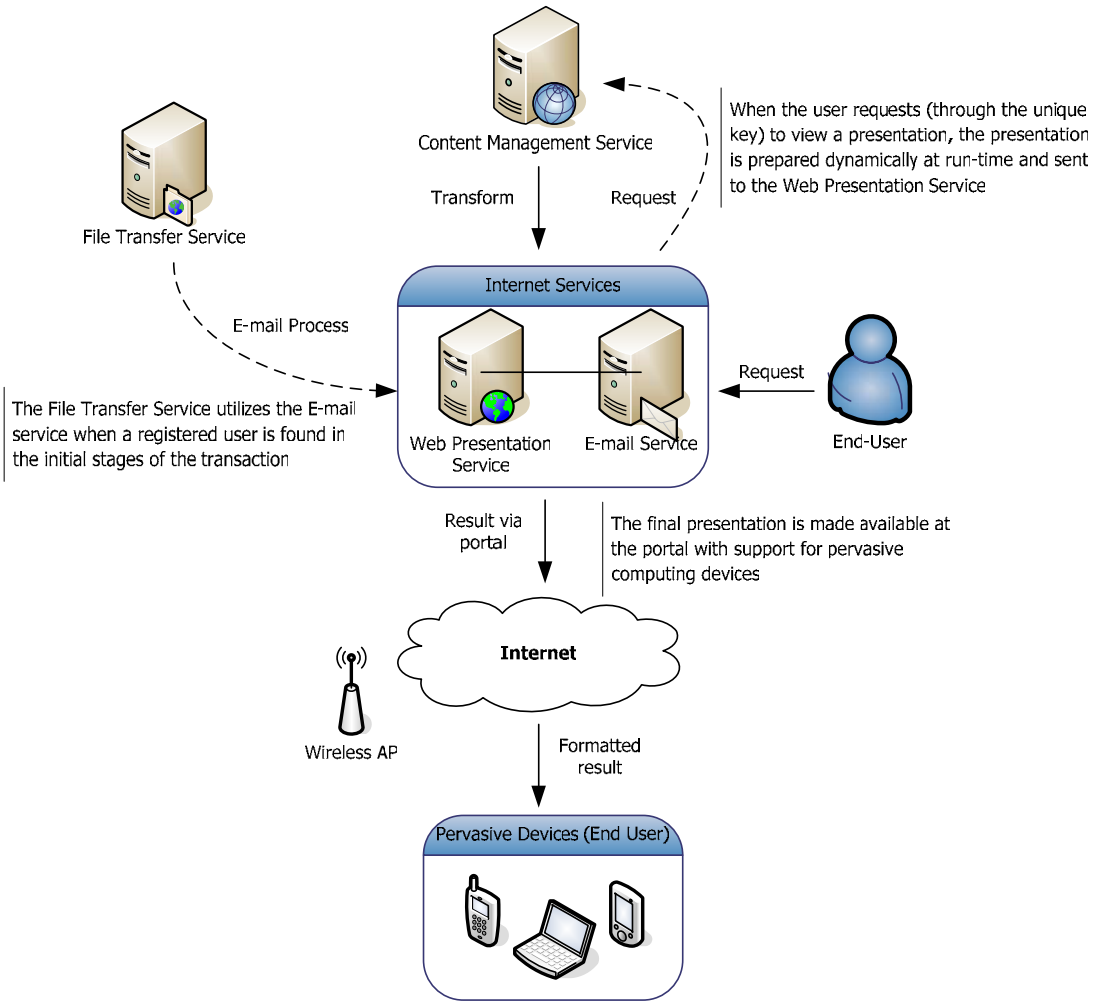

Figure 19. The Internet Service

4.8. Scenario Demonstration

In this section we present a scenario to demonstrate the practicability of the proposed system. We describe the process of each step from a high and low level view, and depict the process with screenshots where applicable. We envision the following situation:

A car parked in a "no-parking" zone outside the library is spotted by a campus officer. He takes out his device, attaches his camera and records a video, takes a snapshot and makes a voice recording. In addition, he writes down some comments about the violation. Satisfied, he selects a template created with the Multimedia Visual Authoring Tool, connects to the server, and transfers the files. He receives confirmation once his files have been uploaded. Because the user is a registered vehicle owner, the campus officer receives his contact details and violation number and issues the ticket, at the same time, the server automatically e-mails the vehicle's owner to inform him or her that their vehicle is in violation. The owner then views his or her violation on the MobileCampus portal

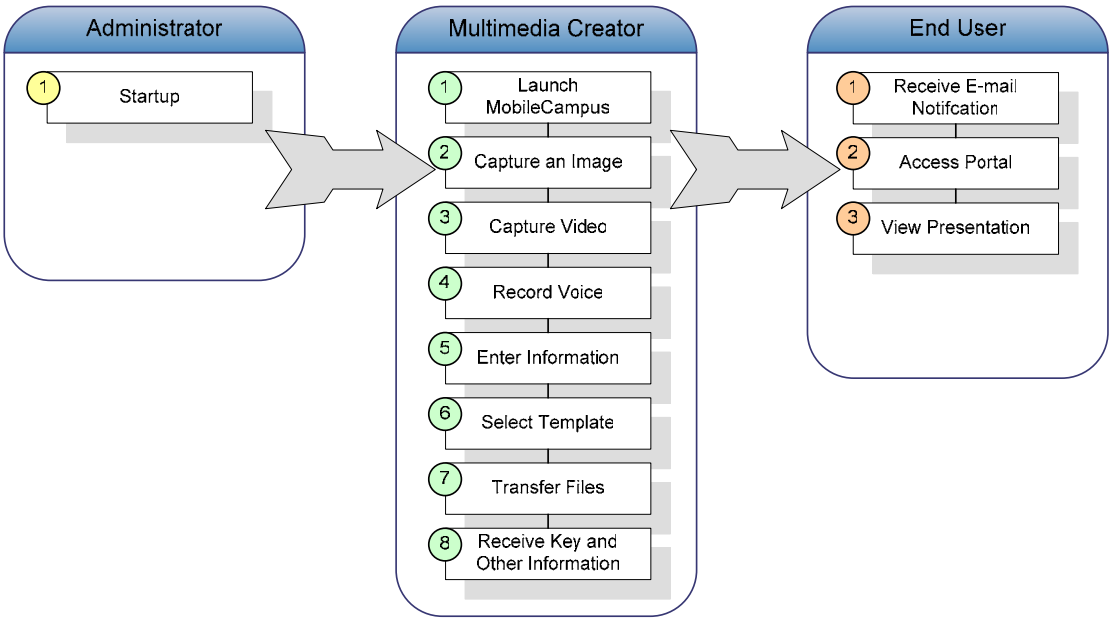

Figure 20. Scenario

An outline of this scenario is depicted above in Figure 20. A detailed description of this transaction is elaborated as follows:

Table 2. The first four steps of the transaction

Steps	**High-Level View**	**Low-Level View**
Launch MobileCampus	The campus officer launches MobileCampus and accesses the Multimedia Capturing Service	None
Capture an Image	The “still image” radio button is selected and the campus officer clicks the “capture” button	The image is stored locally as a JPEG image
Capture Video	The “movie” radio button is selected and the campus officer clicks the “record” button. When the officer has completed recording the video, he clicks the “stop” button. Any adjustments to lighting, brightness and size are made here	Because the current camera API does not support built-in video compression, images are saved as JPEGS at a rate of five images per second. Images are compressed and packed into ZIP format in the background
Record Voice	The officer uses the voice recorder to record voice	The voice recording is stored locally as a wave file and compressed and packed into ZIP format in the background

Table 2 and Table 3 describe the first five steps of the transaction (The Multimedia Capturing Service). Figure 21 and Figure 22 illustrate these steps from a high level viewpoint.

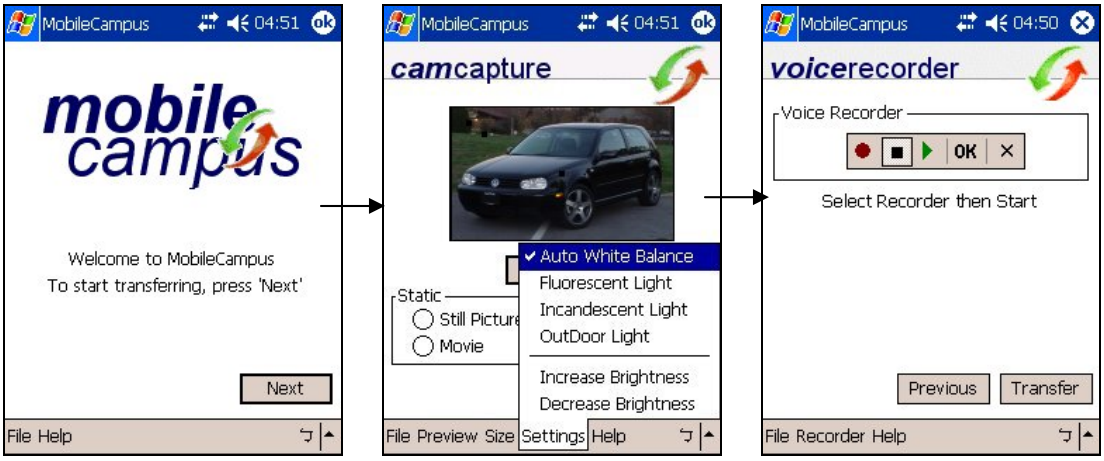

Figure 21. The first four steps of the transaction (screenshots)

Table 3. The fifth step of the transaction

Steps	High-Level View	Low-Level View
Enter Information	The officer enters the necessary information such as the time, location, nature of violation, comments and so forth	This information is stored locally as text

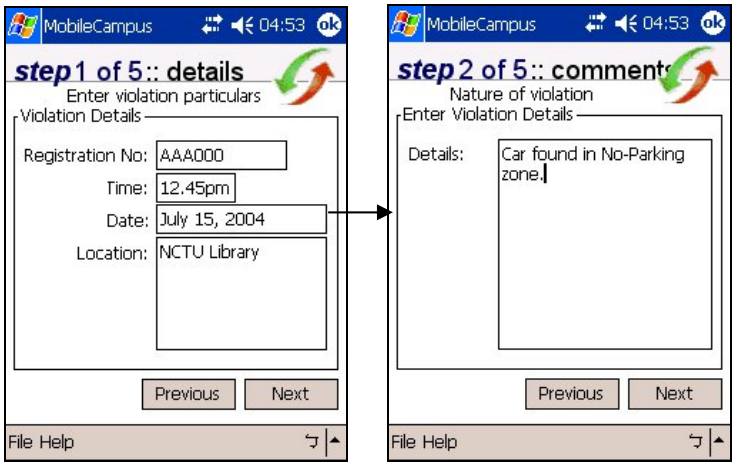

Figure 22. The fifth step of the transaction (screenshots)

Table 4 and Figure 23 describe and illustrate steps six to eight respectively.

Table 4. Steps six to eight of the transaction

Steps	High-Level View	Low-Level View
Select Pre-Defined Template	The officer selects a pre-defined template	Information about the template selection is stored locally
Transfer Files	Once the officer has completed acquiring the multimedia and is satisfied with his template selection, he connects to the server. The file transferring process is automatic	The client connects to the server via 802.11b/g using the File Transfer Protocol (FTP). The user is first authenticated and verified. If the verification is successful, a unique key is generated for the transaction. All acquired compressed multimedia information is then transferred to the server. During this time, information about the transaction is stored in XML. Acquired multimedia files are unpacked and stored on the server in the media library (via the Template Management Service). The server also connects to the vehicle register to check if the owner exists in the database. If the owner exists, an e-mail is generated and sent automatically to the owner
Receive Key and Other Information	When the transaction is complete, the officer will receive confirmation. All information about the owner, details about the transaction and the unique key are sent back to the officer for reference	If queries to the vehicle register are successful, this information is sent back to the officer

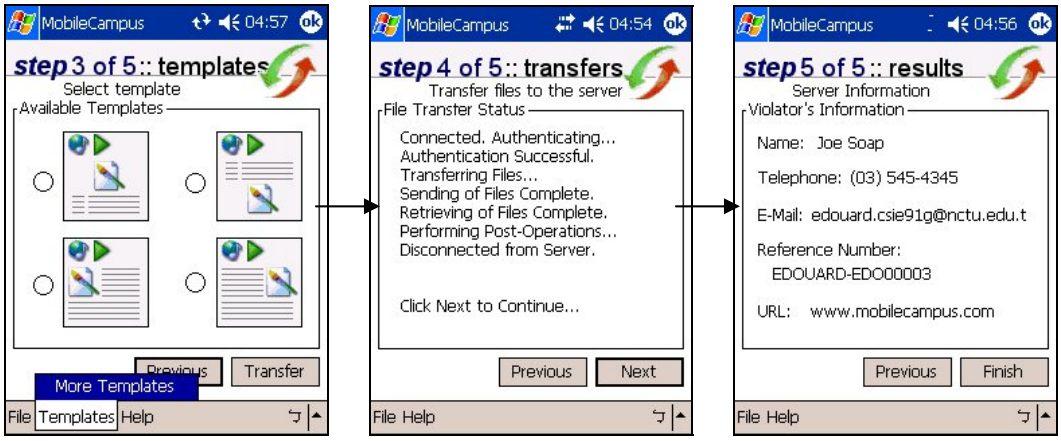

Figure 23. Steps six to eight of the transaction (screenshots)

The end user (vehicle owner) steps are described in Table 5 and depicted in Figure 24 respectively.

Table 5. The end user steps of the transaction

Steps	High-Level View	Low-Level View
Receive E-mail notification	The end user receives an e-mail on his or her pervasive computing device (or desktop PC)	The E-mail Service, when called by the File Transfer Service sends the e-mail to the recipient
Access Portal	The end user accesses the MobileCampus portal	The Web Presentation Service loads the portal. Depending on the device accessing the portal, the Web Presentation Service formats the portal accordingly using CSS and device profiling
View Presentation	The end user enters the key sent to him or her via e-mail, and views the presentation	Once the server receives the key, the transformation process and device customization process takes place for presentation to the end user

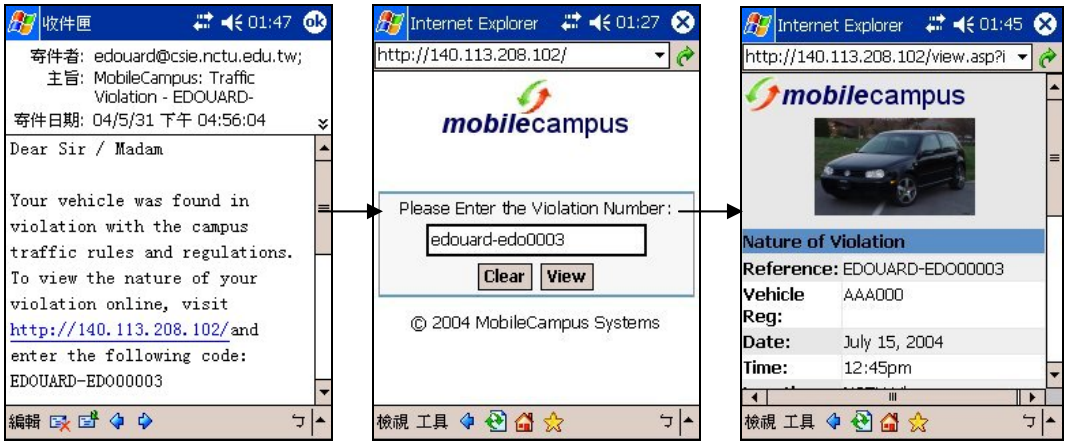

Figure 24. The end user steps of the transaction (screenshots)

5 Assessment

In this paper, we proposed the use of dynamic web-based multimedia templates using XML technologies and how they could be effectively ported for pervasive computing devices taking into account the context of use (environment). We then presented an application example, MobileCampus, to demonstrate the applicability and feasibility of the proposed template mechanism and its related systems.

Earlier, we laid the foundation, scope and criteria from which we intended to evaluate the proposed system. In two general aspects, we qualitatively evaluate the proposed system in terms of:

- How it addresses specific application challenges and supports the vision of pervasive computing

- How it enhances the user experience by taking into account the characteristics of users that will use such an application and the device limitations by working around them to achieve desired outcomes

We believe the proposed system achieved or partially achieved the vision of pervasive computing and enhanced the user experience, not only in terms of meeting the specific challenges and taking into account user characteristics and device limitations, but also in terms of allowing users to access information anytime, anywhere and on-demand within the computing environment. The proposed system particularly contributes to pervasiveness in terms of:

- **Productivity**. Pervasiveness is able to turn idle time into productive time by allowing mobile users to access timely resources while reducing paperwork.
- **Quality**. Access to resources reduces the inconvenience to users.
- **Portability**. The proposed system allows users to connect to resources from anywhere at anytime and on-demand.
- **Centralized Data**. By centralizing data on back-end servers, mobile users are reduced of the burden of unsynchronized data, and offered accurate, up-to-the-minute analysis of information.

Table 6 summarizes the system evaluation in terms of the application challenges.

Table 6. Summary of evaluation against the application challenges

Challenge	System Evaluation
Automation	We emphasize a thin-client approach by transferring intensive processing duties to backend services (the File Transfer Service, Template management Service, Template Creation Service, Content Management Service and Internet Service) to assist with automation responsibilities to relieve end users of several tasks and saving processing power on devices (addressing device limitations). It also assists users with little or no experience working with multimedia presentations and hides system complexity
Content Aggregation	We support content aggregation using XML/XSL. Content aggregation works well in a pervasive computing environment that demands information at run-time and supports heterogeneous devices.
Customization and personalization	We allow users to customize their experience. The integration of the Multimedia Visual Authoring Tool allows users to customize content and import them into the template library. In a pervasive computing environment, tasks should be completed minimally with little effort and time required.
Multi-device capability	The dynamic nature of templates reduced the problem of device heterogeneity by allowing the storing of device profiles and catering for future devices.
Web content and custom applications	We distributed our applications and converged web content with custom applications taking advantage of both browser facilities and custom applications on the pervasive device to offer access to a rich application and content delivery environment.
Integration of Services	MobileCampus is a system that is divided into several services that communicate together to achieve a common system goal across different platforms and protocols.

We found that the implementation of templates addresses the mobile users' characteristics by assisting users to create powerful multimedia presentations from their devices, relieving them of intensive task duties and limitations posed by these devices. In particular, we found that templates enhance the user experience in the following ways:

- They empower users by allowing them to perform tasks efficiently and effectively because they act as starting points for building powerful multimedia presentations minimizing the time needed to perform tasks. Minimum navigation and interaction with the device and real-time access to information creates fewer steps and fewer problems while minimizing errors and increasing data accuracy.
- They help reduce errors by transferring intensive tasks to back-end servers
- They help reduce paperwork and increase accuracy
- They address device limitations such as screen size, screen resolution and colour depth by storing device profiles
- They assist preserving power and power consumption on devices by transferring demanding tasks to back-end servers
- The choice of an open standards approach allows easy deployment, support of multiple devices, networks and back-end systems.

In addition, MobileCampus enables a simplified, independent client representation layer that optimizes the user experience with minimal keystrokes and providing step-by-step navigation. Back-end integration hides the complexity of the process simplifying the user task.

After an extensive qualitative evaluation, it was established that the proposed system generally achieves its goals well, in terms of supporting the vision of pervasive computing and enhancing the user experience by addressing the application challenges, mobile user characteristics and device limitations. However, certain limitations were identified such as the inability to support entire multimedia presentations but only a single scene and the formatting results of presentations on different devices.

6. Conclusion

This study addressed the problem of assisting users perform tasks (such as create powerful multimedia presentations) efficiently and effectively by introducing dynamic web-based multimedia templates. In order for a pervasive computing application to be successfully developed and deployed, a number of challenges need to be addressed. From an application-centric viewpoint, we are particularly interested in specific application challenges, characteristics of mobile users and the device limitations. In addition, we are concerned about the contribution towards the vision of pervasive computing as well as enhancing the user experience in such an environment. The proposed system attempts to consider these factors as important to the successful development and deployment of the system. These factors are also used as metrics from which we qualitatively evaluated the system. The major contributions of this study follow.

We investigated and identified certain challenges for developing and deploying successful applications in a pervasive computing environment and established some specific criteria that should be addressed from an application-centric viewpoint that supports the vision of pervasive computing

and enhances the user experience. We studied the characteristics of the mobile user, and analyzed the attributes of successful pervasive computing applications. In addition, we investigated some approaches to templates. We then designed and implemented a system that demonstrated the use of dynamic web-based multimedia templates for assisting users of pervasive computing devices. We evaluated the system qualitatively in terms of its performance against the criteria mentioned earlier and how it supports the vision of pervasive computing and enhances the user experience. Major contributions of this project include:

- Assisting mobile users with little time and resources at their disposal to create powerful presentations or reports in a quick and efficient manner directly from their devices with minimal effort and skill required.
- Offering users with little experience or programming ability the opportunity to create powerful multimedia presentations from their devices.
- Contributing towards pervasiveness and enhancing the user experience by addressing specific application challenges associated, mobile user characteristics and device limitations with the vision of pervasive computing and dynamically presenting them to users to support the heterogeneity of devices and future devices

Acknowledgements

The authors wish to acknowledge and express appreciation for the collaborated funding support from the National Science Council (Taiwan), Bestwise International Computing Co., and CAISER of National Chiao Tung University.

References

1. E. Ma Poon, D. J. Chen and D. T. K. Chen. “Extending Dynamic Web-Based Multimedia Templates to Reach Users of Pervasive Computing Devices in an e-Learning Environment”. *In Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering*. Miami, Florida. December 13 – 15, 2004. pp. 110 – 113.
2. G. Cohen, “New Opportunities”. International Business Machines. 2003. Available: http://www-1.ibm.com/partnerworld/pwhome.nsf/weblook/news_busopps_adkins_qanda.html
3. G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman and D. Zukowski, “Challenges: An Application Model for Pervasive Computing”. *In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom2000)*. Boston, Massachusetts. 2000. pp 266 – 274
4. G. Couloris, J. Dollimore and T. Kindberg, “Distributed Systems Concepts and Design”, 3rd Edition. Addison-Wesley, 2001.
5. *Introduction to Pervasive Computing*. International Business Machines. 2003. Available: http://www-306.ibm.com/software/pervasive/business_partners/
6. BestWise International Computing Company. Available: http://www.caidiy.com.tw
7. Veo PhotoTraveler. Available: http://www.veo.com.tw
8. Hewlett-Packard iPAQ H5555 PocketPC. Available: http://www.hp.com
9. D. Saha and A. Mukherjee, “Pervasive Computing: A Paradigm for the 21st Century”. *IEEE Computer Society*. March 2003.
10. S. Acharya, “Application and Infrastructure Challenges in Pervasive Computing”. Bell Laboratories, Lucent Technologies, Inc. 2002

11. J. R. Smith, R. Mohan, and C. S. Li, "Scalable Multimedia Delivery for Pervasive Computing". *In Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*. Orlando, Florida. 1999. pp 131 – 140
12. K. Henricksen, J. Indulska and A. Rakotonirainy, "Infrastructure for Pervasive Computing: Challenges". Distributed Systems Technology Centre, The Unviersity of Queensland, 2001.
13. D. Bevis, "Extending Enterprise Applications to Mobile Users". Websphere Everyplace Access, July 2002.
14. *Mobilizing the Enterprise: Solutions for Automating the Everyday Tasks of Mobile Teams*. A Business Whitepaper. Orsus Solutions Limited. 2002.
15. Webster's Online Dictionary. Available: http://www.websters.com
16. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges". *IEEE Personal Communications*. Vol. 8, No. 4. pp 10 – 17. August 2001.
17. D. Saha, A. Mukherjee and S. Bandopadhyay, *Networking Infrastructure for Pervasive Computing: Enabling Technologies and Systems*. Kluwer Academic, 2002.
18. E. Brewer *et al*. "A Network Architecture for Heterogeneous Mobile Computing". *IEEE Personal Communications*. Vol 5, No.5. October, 1998
19. D. Arnold *et al*. "Discourse with Disposable Computers: How and Why you will talk to your tomatoes". *Usenix Workshop on Embedded Systems*, March 1999
20. D. Gribble *et al*. "The Ninja Architecture for Robust Internet-Scale Systems and Services". *Computer Networks, Special Issue on Pervasive Computing*, June 2000.
21. B. Egan. "Making the Case for Wireless Mobility Investment". Mobile Competency, 2003.
22. K. Eustice. "A Universal Information Applicance". *IBM Systems Journal*, Vol 38, No. 4, 1999.
23. D. Norman. "The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution". MIT Press, 1998.
24. *Imagination. Innovation. Results.* IBM's Enterprise Vision for Pervasive Computing. March 2002.
25. M. Dertouzos. "The Future of Computing". Scientific American. August, 1999.
26. S. Midkiff. "Mobile and Wireless Networks and Applications". *IEEE Pervasive Computing*. October – December, 2002.
27. National Express. Available: http://www.gobycoach.com
28. The Waterbury Police Department. Available: http://www.wtbypd.org
29. Lotte Sam Kang. Available: http://www.lotte.co.kr
30. Kudos. Available: http://www.microsoft.com/australia/business/casestudy/studies/kudos.aspx
31. Enning GmbH. Available: http://www.enning.de
32. Carnegie-Mellon University's Aura Project. Available: http://www-2.cs.cmu.edu/~aura
33. University of Washington's Portolano Project. Available: http://portolano.cs.washington.edu
34. Hewlett-Packard's CoolTown Project. Available: http://www.cooltown.com
35. University of California at Berkeley's Endeavour Project. Available: http://endeavour.cs.berkeley.edu
36. Massachusetts Institute of Technology's Project Oxygen. Available: http://oxygen.lcs.mit.edu
37. IBM Websphere Anyplace Project. Available: http://websphereadvisor.com/doc/13254
38. M. Butler, F. Giannetti, R. Gimson and T. Wiley. "Device Independence and the Web". *IEEE Internet Computing*. September – October 2002. pp 81 – 86.
39. J-B. Minchelli. "XSL Device Formatting Using the Microsoft XSL ISAPI Filter". March 2002
40. The Apache Cocoon Project. Available: http://cocoon.apache.org
41. Composite Capabilities/Preferences Profile. Available: http://www.w3.org/Mobile/CCPP
42. Microsoft Mobile Internet Toolkit. Available: http://www.microsoft.com
43. W. Huang, M. O'Dea, and A. Mille, "ConKMeL: A Contextual Knowledge Management Framework to Support Intelligent Multimedia e-Learning". *In Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering (ISMSE'03)*. 2003. pp 223 – 231.
44. L. Bouthilher, "Synchronized multimedia on the Web". *Web Techniques Magazine*, 3(9), September 1998

45. Unwired Planet. Handheld device markup language specification. Technical Report Version 2.0, Unwired Planet, Inc., April 1997.
46. O. Avaro, P. Chou, A. Eleftheriadis, C. Herpel, and C. Reader, "The MPEG-4 systems and description languages: A way ahead in audio visual information representation". *Signal Processing: Image Communication, Special Issue on MPEG-4*, 4(9):385-431, May 1997
47. J. Honeycutt and R. Lutsch. "Wireless Network for Mobility and Collaboration". *Microsoft Corporation*. December 2002.
48. V. Stanford. "Pervasive Computing Puts Food on the Table". *IEEE Pervasive Computing*. January – March, 2003.
49. P. Lipton. "Making your Mobile e-Business a Success using XSL and Web Services". *Computer Associates*. October 2002.