Personalised mobile learning content delivery: a learner centric approach

Ty Mey Eap
School of Interactive Arts and Technology, Simon Fraser University, 250-13450 102 Avenue, Surrey, BC, Canada V3T 5X3
E-mail: teap@sfu.ca

Dragan Gašević, Kinshuk, Fuhua (Oscar) Lin
School of Computing and Information Systems, Athabasca University, 1 University Drive, Athabasca, AB, Canada T9S 3A3
E-mail: dragang@athabascau.ca
E-mail: kinshuk@athabascau.ca
E-mail: oscarl@athabascau.ca

Abstract: This research proposes a context-aware framework that automatically adapts learning content designed for personal computers for mobile learners. The framework is a web application plug-in that captures access patterns and uses the patterns with trained configurations, learner preferences and device profiles to tailor learning content to the learners' needs. Learners can switch devices without losing learning context, and the framework would use Semantic technology to search and make content learners may eventually need available to them. The framework is inline with the surveys recently conducted on mobile learning and uses only proven technologies to guarantee a successful implementation.

Keywords: content adaptation; content organisation; knowledge management; mobile learning; Semantic web application; software architecture; software engineering; software framework; web technology.


Biographical notes: Ty Mey Eap is a PhD candidate at Simon Fraser University, Canada. He received his BSc and MSc from Simon Fraser University. His primary interests lie in the areas of internet security, web service security, service-oriented architecture, e-learning management, artificial intelligence and Semantic web. His current research looks at the internet security that gives end-users full control over the management of their identities and privacy and improves the security of web services. He has authored/co-authored more than 20 research papers and has been involved in divert projects funded by NSERC, SSHRC, Canarie, A.W. Mellon Foundation (USA).
Dragan Gašević is an Assistant Professor in the School of Computing and Information Systems at Athabasca University, Canada. Gašević received his Dipl.Ing., MSc and PhD in Computer Science from the Department of Computer Science, University of Belgrade, Serbia, in 2000, 2002 and 2004, respectively. His primary interests lie in the areas of service-oriented architectures, Semantic web, model-driven engineering, technology enhanced learning, and petri nets. Gašević has authored/co-authored more than 160 research papers, books and book chapters. He had organised a number of international conferences and had been editorial board member and guest editor of many international journals.

Kinshuk is a Professor and the Director in the School of Computing and Information Systems at Athabasca University, Canada. Also, he was an Associate Professor and the Director of Advanced Learning Technology Research Centre at Massey University, New Zealand. He holds a docent position with the University of Joensuu in Finland and is the Chair of IEEE Technical Committee on Learning Technology and Editor of the SSCI indexed Journal of Educational Technology and Society. He has been involved in large-scale research projects for exploration based adaptive educational environments and has published over 200 research papers in international refereed journals, conferences and book chapters.

Fuhua Lin is a Professor in the School of Computing and Information Systems at Athabasca University, Canada. Before joining Athabasca University, he was a Research Officer of Nation Research Council (NRC) of Canada. He has published more than 70 journal papers, book chapters and conference papers. He is an Editor of a book entitled ‘Designing distributed learning environments with intelligent software agents’. His research interests include knowledge representation and reasoning, intelligent agents, multi-agent systems, software architecture for e-learning and virtual reality. Lin obtained his PhD from the Hong Kong University of Science and Technology, Hong Kong, China, in 1998.

1 Introduction

The computing power of mobile devices has increased significantly over the recent years. Mobile browsers have taken advantage of this capability to offer better rendering of web content designed for personal computers. Opera had announced recently that Opera 9 (http://www.opera.com) can provide an overview with intelligent zooming and is capable of supporting advanced functionalities of Web 2.0 such as Ajax. Other leading mobile browsers such as Pocket Internet Explorer, NetFront, Minimo, Series 60, Konqueror and OpenWave will undoubtedly follow suit if they have not offered similar capabilities. Yet, according to Hines’s (2006) article in eWeek, many e-businesses still prefer using client/server type solutions to deliver business applications to mobile users. Mobile applications have better control over its environment and can compensate for the disadvantages of mobile devices such as intermittent connections and the limitations of bandwidth, storage capacity and input/output capabilities to provide richer user experiences.
Having mobile browsers capable of adapting web content for mobile devices allows users to access all currently available web content. However, this approach requires all users to have mobile devices that have the same capability of a personal computer and assumes that the users are satisfied and have adapted to navigate on the small screens of mobile devices. In reality, regardless of the devices they use, users always want to access their desired content in fewer steps possible. To compensate for the disadvantage of mobile devices, researchers have looked into the mobile content adaptation that offers better personalisation for mobile content delivery (Anderson, Domingos and Weld, 2001; Lee, Ko and Fox, 2003; Lum and Lau, 2003; Armstrong et al., 2006; Artail and Raydan, 2006; Blekas, Garofalakis and Stefanis, 2006; Harper, Yesilada and Goble, 2006; Lehtonen et al., 2006; Maekawa, Hara and Nishio, 2006; Mohomed, Cai and De Lara, 2006). This group of research focuses on the use of proxies and dynamic content re-authoring to adapt content for mobile devices. Content re-authoring consists of striping or modifying graphics, scripts, and complex HTML into a format that mobile devices can display (e.g. converting nested tables into single-column tables, resizing images, etc.). The research produces interesting results in reducing the bandwidth, but content re-authoring does not always produce high quality mobile content. Furthermore, the layouts, graphics, and formats represent the cognitive landscape of a webpage. Users may no longer have the same understanding of the content when all layouts and graphics are removed.

Other researchers decide to tackle the grassroots of the web design and adopt device independence web application framework (DIWAF) proposed by Giannetti (2002). DIWAF takes into account the device profiles at the design stage to design content for a wide range of end-user devices (Glover and Davies, 2005). In this design framework, authors provide metadata for the content adaption process. However, they only have to provide metadata for the most used devices. DIWAF would use metadata of closely related devices to adapt content for a device that has no metadata. This principle is difficult to achieve in practice and is not scalable. Designing a metadata that works for many devices is similar to designing an HTML page that works on different browsers. Authors must find a layout of the metadata that is common to all the target devices. The task is complex and difficult to achieve. At most, they can design a metadata that works for two or three devices. In addition, with the rapid growing of technology, the authors have to redesign their metadata continually.

Blekas, Garofalakis and Stefanis (2006) proposed using RSS feeds to adapt web content for mobile devices. This could be an interesting approach if all websites enable RSS feeds. The layout of RSS documents is simple and easy to display on any mobile device. It consists of a few elements (e.g. channel, item, title, link, description, creator and date), which fit very nicely onto a mobile screen. However, the purpose of RSS is for websites to publish frequently updated content such as blog entries, news headlines, or podcasts. Most websites do not have RSS feeds enabled. Blekas, Garofalakis and Stefanis (2006) have to provide a content re-authoring module for their proposal to adapt content when RSS feed is not available. To enable RSS feeds, web publishers must design their RSS; they have to decide on the content they want to publish in the RSS feeds. Some will discover that RSS cannot store all the information they want to publish.

So far, content adaptation research has considered the use of mobile devices as a substitution or a replacement of a personal computer. However, no one in an office or at home would use mobile devices to do their work or navigate the web. The mobile devices’ screens are not adequate for regular work. Furthermore, most users, students in
particular, cannot afford a high-speed internet connection for their mobile devices. Most students would use personal computers for their everyday study. In some occasions, however, students might use their mobile devices to flip through specific learning materials while sitting down in a park or commuting from homes to schools or visa versa. Therefore, it is reasonable to assume that learners would use mobile devices as an assisted tool to retrieve specific information when they are away from schools or from homes or when their laptop is out of wireless range. Students may use mobile devices to check their class schedules, have a quick chat with another student or a professor, or review some learning materials before the start of their classes. This assumption is confirmed by the survey conducted by Chen and Kinshuk (2005). Students participated in the survey mentioned that mobile devices are useful for accessing information such as registration, enrolment, library records, exam results, lecture notes, assignments, university news and some useful campus advertisements. Another survey conducted by Brown, Ryu and Parsons (2006) also produced similar results. These surveys clearly show that the learners would use mobile devices to access specific information due to the ‘urgency’ of learning needs, which is also an observation of Goh and Kinshuk (2004).

Based on the analysis above, we propose a personalised and context-aware framework (PCAF) to adapt and deliver learning content designed for personal computers to mobile learners. PCAF is a web application plug-in component that uses participatory design approach to gather required information on mobile devices, access patterns and learner preferences to adapt mobile content tailored to the learners’ needs. Learners decide what they want to see on their mobile devices by configuring the mobile content using provided mobile content-delivery editor (MCDE). However, PCAF would pre-configure the content as close as possible to the learners’ needs using learner and device profiles and other available information such as learners’ access patterns and previous configurations. As a result, learners only have to modify a few configurations. The configurations are stored as templates, which become a feedback to PCAF to pre-configure other content for the learners. Moreover, a template is structured information designed for interoperability and sharing. PCAF can extract patterns from the templates and use them to adapt content for other learners. To improve usability, authors can pre-configure a set of learners to provide PCAF with some trained configurations so that average learners do not have to do any configuration. Implemented as a plug-in to a web application, PCAF is easy to integrate with content management systems and is readily to service mobile learners. PCAF is a low cost solution for enabling mobile portal without having to redesign web content for every type of mobile devices.

2 Motivation

Mobile technology can enhance education in many ways: collaborative learning, outdoor activities, educational games, etc. As cornered by Vavoula and Sharples (2002) and Uden (2007), mobile technology provides learners with learning tools and educational content and aids in knowledge acquisition at any place and time. Enabling mobile learning is the main motivation of this research project. However, the solution produced by this project should be able to open mobile portal for any web application developed for personal computers. However, our ultimate goal is to develop an intelligent and context-aware solution that can deliver personalised mobile content to learners from existing and new web-based learning content management systems (LCMS). Context plays an important
role in learning (Uden, 2007). It influences the embodiment of knowledge. This goal leads us to propose a context-rich framework that automatically captures Learner Information Retrieval Profiles (LIRProf), User Agent Profiles (UAProf) and learner inputs to deliver and suggest content to learners that enhances their individual learning experience. This project simplifies the previous mobile agent approach taken by Kinshuk and Lin (2004). PCAF would filter unwanted data to reduce data transfer cost and internet traffic and personalise the mobile content with respect to the learning context of each learner. Learners can rapidly access learning content they need without having to zoom in and out or navigating through a maze of mobile content constructed from existing web content. To improve the processing speed, PCAF would cache and fetch learner data based on the learner information retrieval patterns but would not have the ability of an agent, as proposed by Kinshuk et al. to process information in the background on behalf of a learner. Agent technology has many advantages but is difficult to maintain, and it does not integrate well with web applications.

3 Background information

This section provides an overview of research in the area of the content adaptation and the design of the mobile content. The literature review emphasises the limitations of mobile devices and the scalability of the mobile content design.

3.1 User agent profiles

Mobile devices have a wide range of capabilities. The law of competition forces mobile vendors to diversify and offer new types of mobile devices with better computing power and innovated characteristics. Concerned that the device heterogeneity limits user abilities to convey their content presentation preferences to the servers, open mobile alliance (OMA) proposed UAProf (User Agent Profile, 2006) specification to enable a mobile client application to convey the device profiles and preferences to web servers. The specification is concerned about capturing classes of device capabilities and preference information. The classes include (but are not restricted to) the hardware and the software characteristics of a device as well as information about the network to which the device is connected. In accordance with Composite Capability/Preference Profiles (CC/PP) of W3C recommendations (CC/PP Structure and Vocabularies, 2004), OMA defines the UAProf using RDF schema. UAProf, also known as capability and preference information, may include following definitions:

- hardware characteristics (e.g. screen size, colour and image capabilities, manufacturer, etc.)
- software characteristics (e.g. operating system, list of audio and video encoders, etc.)
- application preferences (e.g. browser, supported markup languages, supported scripting languages, etc.)
- WAP characteristics (e.g. WML script libraries, WAP version, WML deck size, etc.)
- network characteristics (bearer characteristics such as latency and reliability, etc.).
Mobile device-related applications could use information in the UAProf to optimise algorithms and tailor content to the capabilities of the mobile devices (see Figure 1). However, as Glover and Davies (2005) pointed out, in practice, manufacturers do not always supply accurate UAProf information. New devices have new capabilities and need new vocabularies that have not been defined in the UAProf specification. Consequently, many device profiles are not up-to-date or unavailable during runtime.

### 3.2 Content adaptation

Content adaptation research focuses on finding technologies to adapt existing web content suitable for mobile browsers. This literature review focuses on two approaches: automatic content re-authoring and device independence design. Considered as a user-centric approach, the content re-authoring approach has two options: integrating content adaptation re-authoring software on a proxy server or on a mobile browser. A proxy server sits between the mobile devices and the web portals. Its role is to intercept and forward end-user requests to target web servers and then capture the outputs from the servers and transform the outputs to a suitable format for the mobile devices. Similarly, content re-authoring software can be integrated with mobile browsers to transform the outputs from web servers. Either way, the content re-authoring software should have access to the device profiles and the user preferences to optimise the rendering quality.

Device independent design, also known as a device-centric approach, focuses on mobile devices rather than mobile users. Research in this area has more or less adopted the DIWAF. The design framework allows authors to design content based on the DIWAF specification and provide metadata to the DIWAF engine to render content for different device classes.
Content re-authoring technique consists of restructuring web pages and reducing image sizes to fit the screens of mobile devices. Some mobile browsers already use this technique for their web content rendering strategies. This section reviews and analyses the approaches proposed by some of the content re-authoring projects and uses the analysis to construct the arguments to solidify PCAF idea.

A popular approach used by some mobile browsers, also proposed by Chen, Ma and Zhang (2003), is presenting an original webpage as a top-level thumbnails page that has each thumbnail links to a portion of the original page. The technique used by Chen et al. is a two-step-process content adaptation. The first step is analysing and splitting a webpage into a two-level hierarchy structure. The top hierarchy is a table of content of sub-pages presented in colour-coded thumbnails. A page analysis algorithm undergoes through several iterations to find the best way to break down the content into smaller partitions that can fit into mobile screens. However, not all the web pages are suitable for splitting. When this occurs, auto-positioning method is used as an alternative to provide similar user experiences. This content re-authoring method produces crude rendering of Opera 9 browser. The method preserves the layout of the original content, which is represented as a layout of thumbnails. Applying this method at the client side may be more efficient than at the server side since a server needs to cache all the partitions and spoon-feed the partitions to mobile devices. This can create huge internet traffic. On the other hand, supporting this feature at the client side requires mobile devices to have large memory capacity.

An equally popular approach is the restructuring of complex HTML structures into a simple structure and splitting web pages into sub-pages. To reduce the bandwidth further, some proxies strip all the comments and scripts that cannot be processed by the target mobile browsers. This approach may also resize images proportionally to the display area. Different favours of this approach can be found in the literature (Artail and Raydan, 2005; Blekas, Garofalakis and Stefanis, 2006; Mohomed, Cai and De Lara, 2006 and Zhang, 2007). Some mobile browsers such as Pocket Internet Explorer and NetFront had adopted this approach to render the display of the web content. Along with this approach, Artail and Raydan (2005) had proposed a solution that could be of interest. They use JavaScript to control the display of the content by hiding parts of the page and allowing users to navigate through the page interactively. By downloading the entire webpage to mobile devices before displaying, they can reduce waiting time, power consumption, and internet traffic. The technique involves re-authoring the web pages such as converting tables to single-column rows of data, resizing images to fit the screens of a target mobile device, and adding JavaScript to control the display of each partition. The trade-off of this approach is the requirements of mobile devices with fast computing power and large memory size. Another alternative to this approach is the use of RSS feeds to service mobile devices. As mentioned in Section I, RSS is an efficient way to deliver information to mobile devices, but RSS is not suitable for every web publisher.

With the versatility of mobile devices and their usage, content re-authoring software cannot always guarantee the quality of service (QoS). Lum and Lau (2003) proposed a user-centric decision-making engine that uses user preference profiles and device profiles as a quantitative method to measure user satisfaction. Since QoS is a ‘collective effect’ that determines the degree of user satisfaction, an algorithm that fulfils most of user
preference profiles and device profiles should have higher QoS. Hence, the decision engine can look for the best tradeoffs among various parameters to get the optimal QoS.

Re-authoring software should integrate user-centric decision-making engine to ensure its rendering QoS. However, the use of user preference profiles needs infrastructure support. The concept of user preference profiles is difficult to capture in design because user preferences are not static values. In addition, to process user preference profiles, software applications must understand the vocabularies. W3C recommends using the CC/PP specification to describe user preferences. However, CC/PP itself does not define the actual components and attributes to be used (Glover and Davies, 2005); it is only a framework to define those components and attributes. To ensure interoperability, developers must agree upon a vocabulary and a set of attributes that they use to define the user preferences. Furthermore, users usually use trial and error to select their preferences. Users do not know what their preferences are until they experience those preferences on a setting. Therefore, asking users to enter their preferences without visual aid is not an ideal solution.

Ideally, user preference profiles should be captured automatically from user activities without users having to enter manually. Based on this idea, Anderson, Domingos and Weld (2001) proposed a framework that infers user preference profiles automatically from users’ web access activities. PCAF is adopting similar approach. It will deduce learner preference profiles from learning activities.

3.4 Device independence design

As Blekas, Garofalakis and Stefanis (2006) pointed out, device-specific content authoring would produce high quality content, but most websites cannot absorb the cost of this type of development. Giannetti et al. (2002) proposed a DIWAF that enables the delivery of content to many kinds of device classes. It is a ‘single authoring’ principle of designing for the most capable device and automatically adapting content for different device classes. The framework uses author-provided metadata to adapt content for mobile devices. If DIWAF engine cannot find the metadata for a target device, it will use a metadata of a closely related mobile device. Therefore, in theory, authors only have to provide metadata for the most used devices. However, this design approach is facing several issues. A metadata of a closely related device does not guarantee QoS, and the design framework imposes a steep learning curve on authors. To provide metadata for the adaptation engine, authors must know how the engine works and need to adjust content for each specific device. The content developments will be constrained by the authors’ abilities to produce the metadata. Figure 2 shows the author creates a layout of the mobile content in an XML document and sets the presentation priority to provide alternatives for the display. In the example, the author creates $2 \times 4$ grids and assigns a display area for each cell. In each area, the author lists a number of paragraphs that the adaptation engine can use. Clearly, this is not very intuitive. In Figure 3, Glover and Davies (2005) proposed a simpler framework based on the use of templates. In this design framework, authors design a template for each device. The adaptation engine selects an appropriate template for a target device and populates the template with raw data. Glover and Davies framework is no less than a device specific design. It requires authors to provide a template for each type of mobile devices. The number of templates grows as the product of mobile devices’ types. Consequently, both frameworks face a scalability issue and high development cost.
Figure 2 An overview of device independence web application framework metadata (see online version for colours)

```xml
<root>
  <viewport name="display">
    <layout name="main">
      <grid row="3" column="4">
        <row name="first">
          <cell ref="/areas/area[name='01']"/>
          <cell ref="/areas/area[name='02']"/>
        </row>
        <row name="second">
          ...
        </row>
      </grid>
    </layout>
  <areas>
    <area name="01" priority="10" navindex="001" width="50%" height="50%">
      <paragraph priority="100"/>
      <content ref="Binding.xml#pointer(//alt...name='alt_bind_0'))"></content>
      <paragraph>
        <paragraph priority="10"/>
        ...
      </paragraph>
    </area>
    ...
  </areas>
</viewport>
</root>
```

Figure 3 Glover and Davies’s simplified Version of device independence web application framework (see online version for colours)
4 Personalised and context-aware framework

The findings in the literature and surveys conducted by Chen and Kinshuk (2005) and Brown, Ryu and Parsons (2006) led us to propose our PCAF. Our framework targets mobile learners such as students who use mobile devices as an assisted tool to retrieve specific information when they are away from their schools or homes. In our framework, we attempt to fulfil the following design objectives:

- provide a low cost infrastructure to enable mobile learning portals from existing web portals designed for personal computers
- transferring LIRProf from personal computers to mobile devices
- allow mobile learners to customise the delivery of their mobile content.

The design is based on the following constraints:

- device profiles are not always available during runtime
- mobile learners do not always know what their preferences are
- learners may change their preferences constantly
- authors cannot deal with too many devices or parameters when designing learning content.

Figure 4 shows an overview of PCAF architecture. Although this article shows the framework’s proof-of-concept on Java platform and Tomcat web server, the proof-of-concept should be valid for most platforms and web servers. Filter technology exists in
Personalised mobile learning content delivery: a learner centric approach

all web servers. Usually, filter is used to determine whether to accept user requests or not. Request filter is the main requirement for PCAF architecture. PCAF encapsulates filter in the mobile request filter (MRF) to intercept and redirect HTTP requests to PCAF core component, the mobile proxy portal (MPP). Equipped with device identifier module, MRF is able to determine from HTTP request headers whether a request is from a mobile device or from a personal computer. If the request is from a mobile device, it redirects the request to MPP; otherwise, it releases the request to LCMS. Meanwhile, it captures and analyses learning activities to construct information retrieval patterns.

4.1 Mobile proxy portal

MPP is the core component of the PCAF architecture. It has four interface modules (see Figure 5): mobile request filter, HTML validation wrapper, mobile content editor, and content adaptation.

- **MRF** contains a device identifier sub-module that is able to identify devices and construct a basic knowledge of different UAProfs based on the HTTP request headers. This research implements MRF as an extension of Tomcat Servlet filter. MRF has access to HTTP request headers as well as the web application context; it should be able to determine the URI of the target learning content. We will provide in Section 4.2 a detailed implementation of the MRF module.

- **HTML validation wrapper** is responsible for tidying up HTML documents to valid XML documents. Although web industry has adopted XHTML specification to encourage authors to develop XHTML-compliant web content, a vast majority of web documents are valid XML documents. As a result, software cannot manipulate or process web documents without passing them through HTML validation software. MPP also has HTML validation incorporated and use it to validate and tidy up invalid HTML documents. MPP HTML validation is a wrapper of HTML Cleaner (http://htmlcleaner.sourceforge.net/). Its main function is to transform outputs from LCMS to XML streams and pass the streams to MPP.

- **MCDE** allows mobile learners as well as authors to personalise mobile content. The MCDE role is to get additional information on UAProfs and learner preferences with the assistance of learners and authors. MCDE appears to learners as a tool that enables them to control the flow of the information to their mobile devices. Meanwhile, MCDE captures learner preferences, access patterns, and device profiles. Each learner has a master template created automatically by the MPP based on the learner’s LIRProf and UAProf. As learners go from web pages to web pages to personalise their mobile content, MCDE uses their master templates to pre-process content before presenting the content to them. The master templates are updated as learners enter or change their preferences. This allows learners to control the delivery of their mobile content. Section 4.4 will describe in detail the MCDE mechanisms and functionalities.

- **Content adaptation module (CAM)** contains content rendering sub-module that is capable of rendering personalised mobile content based on templates. CAM serves as a coordinator for the storage management and provides various utilities to the MCDE and MRF. We will describe CAM in more detail in Section 4.5.
4.2 Mobile request filter

Filter is a feature that exists in all web servers. Its function is to intercept HTTP requests. Web servers intercept coming requests to determine if the requests have sufficient privilege to access web resources. MRF performs in similar way. It intercepts and redirects mobile requests to MPP, which uses CAM to render learning content for mobile devices. Although UAProf may be not available at runtime and may not provide all the required information for CAM, MRF should be able to determine if a request is from a mobile device or a personal computer. This responsibility is only part of the MRF’s tasks. MRF has two other major functions: building a UAProf repository (Section 4.2.1) and LIRProf knowledgebase (Section 4.2.2).

4.2.1 Building User Agent Profiles repository

As discussed in the literature review, the UAProf mechanism does not guarantee that all the required attributes are available for the content adaptation. To address this problem, we propose a two-step process for capturing UAProfs. The first part is in this MRF module. Through the UAProf mechanism, the MRF extracts as much information as possible on UAProfs and stores the information in a repository. The second part is in the MCDE. Learners provide information needed to make the personalised content delivery possible. However, a UAProf is shared information. If a learner has already entered the required information, other learners using the same version of browsers or devices do not need to re-enter the same information.

4.2.2 Building Learner Information Retrieval Profiles knowledgebase

To enable personalised mobile content, MPP must gather information about learners including learner preferences and their learning activities. Preferences can refer to different things including colour, language, and other characteristics such as graphical or
text-based user interface, and so on. Preferences also vary in time and depend on the device capabilities. On a desktop computer, some learners may welcome a rich graphical user interface while on a mobile device, due to the latency of the content delivery, some learners may prefer text-based user interface. When capturing LIRProfs, MPP may be able to deduce some of the learner preferences. For example, MPP can deduce that learners who spend more time on text-based presentations and skim through rich graphic content prefer or do not mind text-based presentations. Learners could confirm this deductive reasoning when they configure their mobile content. MRF, a sub-module of MPP, must initiate the process for building the LIRProf knowledgebase and learner preferences. While intercepting requests from learners, MRF collects information on the learners. MCDE uses this information to pre-process web content and lets learners or authors to do the final configuration.

4.3 HTML validation wrapper

W3C had launched the XHTML specification in the year 2000 to encourage the development of well-formed HTML and to improve the accessibility of HTML content. Several developers have engineered tools to tidy HTML documents. However, unless integrated with the authoring tools, most of the authors would not have time to use HTML-tidy tools to validate their HTML documents. In addition, when applying HTML-tidy tools on the content produced by an authoring tool, the content may no longer display correctly on commercial web browsers. As a result, a majority of existing HTML pages are not well-formed XML documents. Therefore, web pages must undergo validation process before any adaptation can take place. XHTML is ideal for content adaptation software because the documents follow strict XHTML guidelines, but this project primarily needs a well-formed XML documents. HTML Cleaner (http://htmlcleaner.sourceforge.net/), a lightweight conversion tool built on Java Swing HTML parser, is a good tool for converting a sloppy HTML into a valid XML document. The advantage of this tool is it uses Java native library, therefore, is easy to modify and adapt it to the need of this project.

4.4 Mobile content-delivery editor

As mentioned earlier, MRF builds a knowledgebase from access activities and captures UAProfs from the HTTP request header. The information allows MPP to generate a master template for each learner. Before loading content, MCDE passes the content through the HTML validation wrapper. The wrapper returns well-formed XML documents. MCDE analyses the documents, locates HTML tags that have content, and places markers on those elements. As shown in Figure 6, the markers are placed as check boxes for the learners to select. Using learners’ master templates, MCDE pre-selects the content for learners. Learners only have to modify a few selections. Learners can also verify their UAProfs and make necessary corrections. Configurations are saved as templates, and the master templates are updated. As learners continue to configure their mobile content, patterns would emerge from the configurations. These patterns provide good indication for learner preferences. In this project, we demonstrate only a proof of concept. MCDE is primitive and basic. Pending enhancements of MCDE are carried on in the future research and development. For example, a better algorithm could place markers without affecting learner visual perceptions. The mobile viewer could have a
T.M. Eap et al.

WYSIWYG feature and allow the learners and authors to personalise further the mobile content by rearranging the display of the content. Finally, a default set of configurations could be included in the framework so that average learners do not have to do any configuration. These enhancements will enrich furthermore learner experiences and make MCDE more user-friendly.

Figure 6  Personalised and context-aware framework mobile content-delivery editor (see online version for colours)
4.5 Content adaptation module

Once templates have been created, adapting content is a straightforward process. It is a matter of copying selected items from web content and adding them to the new document built from the templates (see Figure 7).

5 A summative evaluation

At this stage of the project, a summative evaluation should provide a valid proof of concept for PCAF. In this evaluation, we seek to analyse how PCAF would enhance learning environment. We use criteria selected by LearnCanada (http://www.learncanada.ca/eval.php), which is an organisation sponsored by CANARIE Inc (Canada’s advanced internet development organisation) to build an interactive virtual learning community for K-12 educators across Canada. LearnCanada had used these criteria for its summative evaluations in different projects. We add a few criteria to make the evaluation more relevant to this project. Using LearnCanada criteria is convenient and seems appropriate. LearnCanada used teachers from different provinces to evaluate its projects, but due to limited resources of our project, we have to lean on the literature and on our expertise in the domain for the evaluation. The result of the evaluation is shown in Table 1.
### Table 1  A personalised and context-aware framework summative evaluation

<table>
<thead>
<tr>
<th>Evaluation criteria</th>
<th>Analysis and description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accessibility</strong></td>
<td>PCAF would undoubtedly improve the accessibility of learning content</td>
</tr>
<tr>
<td></td>
<td>It delivers specific learning content to learners</td>
</tr>
<tr>
<td></td>
<td>Learners participate in the design of mobile content</td>
</tr>
<tr>
<td></td>
<td>Learners can use PCAF as learning aid</td>
</tr>
<tr>
<td></td>
<td>Learners can access learning content through mobile devices</td>
</tr>
<tr>
<td><strong>Response time</strong></td>
<td>Since learners get to their target content in fewer steps, PCAF response time is much faster than other approaches</td>
</tr>
<tr>
<td></td>
<td>PCAF response time should be faster than response time to deliver the entire web content to mobile devices. PCAF uses XSLT for its content adaptation. The time complexity of XSLT transformation is negligible. A web server may already load XSLT engine into its environment. The transformation time would take a fraction of a second. As shown on the equations below, the left hand side needs to convert the content, but at the same time, but PCAF only has to transmit 30% of original content. Since the transformation rate is normally less than the transmission rate, PCAF response time is faster than delivering the entire web content</td>
</tr>
<tr>
<td></td>
<td>[(lr + tr)<em>cs + icr</em>(cs \ast 30%) &lt; (lr + icr)*c].</td>
</tr>
<tr>
<td></td>
<td>Since (tr \ll icr), (hence) (PCAF) response time &lt; Web content response time</td>
</tr>
<tr>
<td></td>
<td>where (cs) is the content size in KB; (lr) is the load rate per KB; (tr) is the transformation rate per KB and (icr) is the internet connection rate per KB</td>
</tr>
<tr>
<td><strong>Frequency and ease of use</strong></td>
<td>Learners can use PCAF as learning aid. PCAF allows learners to access learning content at any time and place. They can get learning content when they need to complete their train of thought</td>
</tr>
<tr>
<td><strong>Learner expectation</strong></td>
<td>Surveys conducted by Chen and Kinshuk (2005) and Brown, Ryu and Parsons (2006) strongly suggest learners want to access specific learning resources. Hence, PCAF should meet learners’ overall expectation by delivering specific learning content to learners to fulfil the urgency of their learning needs</td>
</tr>
<tr>
<td><strong>Effectiveness of the tool to support learning</strong></td>
<td>An early 20th century psychologist, Boggs (1907), supports the observation of Goh and Kinshuk (2004) regarding the urgency of learning need. The urgency of learning needs is the urge to fill gaps in the train of thought. According to Boggs (1907), learners who have the most gaps can take most in and can learn most. Our goal is to help learners fill in most gaps as quickly as possible</td>
</tr>
<tr>
<td><strong>Expanding virtual educational community</strong></td>
<td>Making learning content available on mobile devices and promoting mobile learning will extend the virtual learning community. Therefore, the project is inline with the general electronic learning (e-learning) objectives</td>
</tr>
</tbody>
</table>

However, to ensure the success of the project, we need to conduct other evaluations. From the technical perspective, a simulation testing must be conducted on samples of different learning resources to ensure that PCAF can adapt any learning content and the speed is acceptable for learners. Second, a usability test or heuristic evaluation must be
conducted on the interface to ensure learner acceptance. Effort must be put into the
development of the MCDE to ensure that configuring mobile content is not a burden. The
success of the PCAF project is relative to this effort. Ideally, the tool should be part of the
web content delivery. While learners work on their learning materials, they can choose a
specific content they want to see on their mobile devices. As a result, learners see PCAF
as a learning aid.

6 Conclusions

This article introduced a novel and effective concept for enabling learning mobile portal
from existing and new web LCMS. The proposed solution is inline with the surveys
conducted on the use of mobile devices in online learning. Our proposed framework
responds to the needs of mobile learners and can deal with the lack of software
intelligence by involving learners and authors in the mobile content design. It is a web
application plug-in component designed for easy integration with LCMS. However,
regardless of the usability-testing outcome, the technology developed in this project is
valuable for the development of mobile learning content. Authors can use the
framework’s tools to design mobile content for each type of mobile devices without
much effort. The framework is low cost and provides an excellent tool for learning-
content developers to open mobile portals without re-authoring learning content for each
specific device manually.

This article has shown a vision and a proof-of-concept of the proposed framework.
We tested the technology to ensure the feasibility of the framework. The next stage is to
develop a working solution and deploy the solution on a real setting. This will allow us to
conduct the final heuristic evaluation and a true usability testing.

References

WWW ’01: Paper presented in the Proceedings of the 10th International Conference on World
4th International Conference on Mobile Systems, Applications and Services, pp.56–68.
Cross-Disciplinary Workshop on Web Accessibility, pp.79–85.
and Scientific Methods, Vol. 4, pp.477–481.
mobile learning environment’, OZCHI ’06: Paper presented in the Proceedings of the 20th
Conference of the Computer–Human Interaction Special Interest Group (CHISIG) of
Australia on Computer–Human Interaction: Design: Activities, Artefacts and Environments,
pp.297–300.
at: http://www.w3.org/TR/CCPP-struct-vocab/.


