Journal of Security Engineering
Vol. 2, No. 3, November, 2005

다양한 딜리버리 컨텍스트에 따른 모바일 컨테트 변환에 관한 연구*

박길철 [1], 김양석 [2], 강병호 [2]
[1] 한남대학교 정보통신 멀티미디어 공학부
e-mail : gcpark@mail.hannam.ac.kr
[2] 타즈마니아대학교 컴퓨터 공학과
e-mail : {yangsokk, bhkang}@utas.edu.au

Dynamic Mobile Content Adaptation According to Various Delivery Contexts

Gil-cheol Park[1], Yang-sok Kim[2], and Byeong-ho Kang[2]
[1] School of Information & Multimedia, Hannam University
[2] School of Computing, University of Tasmania

Abstract

Nowadays users can use various devices to access the internet, such as personal computers, mobile phones, and PDAs. Thus each client uses the internet in a different physical device context as well as a unique environmental and personal context. A device-independent internet service, which provides differently formatted internet contents according to the device context, is necessary to minimize content publishing costs and provide seamless service. In this research, we propose a dynamic content adaptation system that changes HTML based wired Web contents into various wireless contents, which are formatted by WML (Wireless Markup Language) or mHTML (Mobile HTML). We employ a single authoring strategy, in which an intermediate adaptation system dynamically generates appropriate wireless contents according to the 'delivery context'. Content extraction rules are created using DOM tree representation and rule editor. These rules are applied to extract relevant contents and make WSDL files. The system then generates appropriate wireless messages according to the delivery context. Our system enables content publishers to easily reuse existing abundant wired Web contents and provide consistent information between wired and wireless internet services.

1. Introduction

The internet is the largest information source in the world, and content publishers, whether they are professional writers or not, can easily publish their contents without time and location limitations. For this reason the amount of internet information has increased explosively since user-friendly graphic user interfaces were introduced in the early 1990s. HTML (Hypertext Mark-up Language) and HTTP (Hypertext Transfer Protocol) are used to draw Web graphic user interfaces and communicate between the server and the client application. Though this language and this protocol have contributed significantly to the success of internet, they also reveal several limitations. Some of these limitations stem from the especially rapid growth of various mobile devices such as PDAs and mobile phones.

A number of problems arise from the fact that the protocol and language of the Web were designed not for mobile applications and mobile devices, but rather for wired applications and devices. Firstly, HTTP protocol does not efficiently support mobile-based internet communication because it is designed for large bandwidth and low delay. Secondly, HTML is not appropriate for small and low performance display because it is designed for "high" performance computers; colour high-resolution, display, mouse, and hard disk. Furthermore, Web pages that are implemented with HTML are optimized for design, not for communication. Device independence is the original goal of HTML, but ad hoc presentation of specific elements of HTML, which have been introduced by browser developers, have blurred the distinction between semantics and presentation. Device independence therefore serves as an attempt to regain some of Web publications' original intent [1].

* This work was supported by a grant No. R12-2003-004-00010-0 from Ministry of Commerce, Industry and Energy, Korea

To remedy the second problem with regard to small and low performance display, two authoring strategies are proposed: multiple authoring strategy and single authoring strategy. Multiple authoring from the server-side for different devices solves the problem if we can effectively separate logic from presentation. Extensible Stylesheet Language (XSL) transformation, for example, is used successfully in this strategy. Though this approach may have the highest performance efficiency and give maximum control over content delivery, it requires high re-authoring costs and cannot provide seamless service between wired and wireless internet services [2]. For this reason, single authoring with appropriate content adaptation is regarded as the more promising solution [3].

The single authoring strategy aims to provide an independent internet service for the device. This goal is accomplished by matching the rendering of the application content and interaction to the device's capabilities. Content adaptation in the single authoring strategy can be implemented with an intermediate approach or client-side approach. Though client-side adaptation can be used to enhance server-side and intermediate content adaptation, it may cause problems related to processing time, network connection, and memory usage [2]. For this reason, many researchers focus on the intermediate adaptation rather than the client-side adaptation [4-8].

We too have employed intermediate adaptation strategy for our research. Our system mediates between mobile client and typical wired Web server. When clients request specific Web services via their own devices, our system detects 'delivery context' by analysing request header information. In this research, we have only focused on physical delivery context, which varies according to the client's device type and their affiliated telecommunication carrier. Our system extracts necessary contents from the Web server's HTTP response message by using heuristic content extraction rules and transforms extracted message into appropriate wireless markup languages according to delivery context.

This paper consists of following: Section Two summarises relevant research results, including context aware computing, device independent and HTML to wireless markup language conversion. Section Three describes our system architecture that mediates Web server and mobile devices. Section Four illustrates how our system works using an example case. Section Five provides conclusions of this paper and further research work.

2. Literature Review

2.1 Context-Aware Mobile Computing

The term "context" is very frequently used but its meaning in computing tends to be vague due to the fact that everything in the world happens in a certain context [9, 10]. In this paper we focus on the context used by applications in mobile computing. There are four categories of "contexts": computing context (e.g. network connectivity, communication cost and bandwidth), user context (e.g., user's profile, location, and current social situation), physical context (e.g. lighting, noise level, and display size), and time [10, 11]. Of the four, we focus on physical context of the internet accessing devices, because each user uses different internet accessing devices with different characteristics of display size, resolution and the like. Different internet accessing devices should render the same content in different ways to optimise their display capability. There are significant differences between desktop or notebook computers and mobile devices like phones and PDAs.

Despite their mobility, mobile devices have many drawbacks compared to personal computers. Siau et al. [12] summarises these limitations as follows: they have "(1) small screens and small multifunction key pads; (2) less computational power, limited memory and disk capacity; (3) short battery life; (4) complicated text input mechanisms; (5) higher risk of data storage and transaction errors; (6) lower display resolution; (7) less surfability; (8) unfriendly user-interface; and (9) graphical limitations". The degree of limitation also varies between mobile devices because manufacturers produce different kinds of devices.

Another diversification factor is the communication carrier's support for mobile markup languages. Unlike HTML in desktop Web publishing, there is no unanimous or dominant markup language in wireless publishing [12]. Therefore, each communication carrier employs a different markup language for their wireless internet services. For example, NTT DoCoMo's iMode uses condensed or compact HTML (cHTML), LG Telecom and SK Telecom use WML (Wireless Markup Language), and KTF uses mobile HTML (mHTML). In this paper, we use the term "delivery context" to specify an internet accessing device's physical context, which includes devices types (mobile devices or desktop devices), mobile devices' attributes (display resolution, and keypad types), and markup language (WML, mHTML, etc.).

2.2 Device-Independent Publications

The original aim of HTML was to provide a device-independent markup language based on document semantics. HTML identified document elements such as headings, paragraphs, and lists without specifying presentation. Therefore, a presentation for a large-screen device can be displayed and interacted with on a small-screen device using markup. In practice, however, it is difficult to use desktop Web presentation to provide mobile internet service because current HTML is blurred by ad hoc presentation-specific elements and attributes. Furthermore, the desktop Web author did not create the presentation with the characteristics of a mobile device in mind. Therefore, the aim of device independence is based on the "single authoring" principle and automatically adapting the content for different devices [1, 3]. The significant benefit of device independence is that the content authors can provide internet services seamlessly and ubiquitously. Authors can publish internet service consistently without significant costs and users can access using any device. Another benefit is the reuse of existing desktop Web resources to provide wireless internet services [13].

Device independence is accomplished by detecting "delivery context" and adapting contents accordingly. Users want to view internet content and use internet applications via various different devices, including PCs, PDAs, phones, interactive televisions, voice browsers, printers and embedded devices such as digital cameras. Before adaptation is performed, the "delivery context" should be known to the adaptation entity. Currently the request header field of the HTTP is used to identify a particular device, because user agents (web clients) identify themselves when they send requests to Web servers. This is done primarily for statistical purposes and the tracing of protocol violations, but does support the automated recognition of user agents.

In addition, there are four alternative proposed capability specification schemes: the W3C composite capability / preferences profile (CC/PP) (http://www.w3.org/2001/di/), the WAP User Agent Profile (UAPROF) standard (http://www.openmobilealliance.org/tech/index.html), the SyncML Device Information standard (DevInf) and the Universal Plug and Play Standard (UPnP) (http://www.upnp.org/). The generic CC/PP framework provides a mechanism through which a mobile user agent can transmit information about the mobile device. The UAPROF is based on the CC/PP framework and includes device hardware and software characteristics, information about the network characteristics, information about the network the device is connected to, and other attributes [2, 3].

Content adaptation can be take place on the client-side, server side or on the intermediate application. The main differences between the three approaches lie in who controls the adaptation process. In server-side content adaptation, servers provide directly adapted documents, whereas intermediate and client-side adaptations use input content provided by servers or proxies [1, 2]. The client-side adaptation is inefficient because delivering a complete page is not effective use of delivery bandwidth. Server-side adaptation offers maximum author control over the delivered content, but this requires many resources for publications and can cause inconsistency between contents. Therefore, to avoid client-side and server-side adaptation problems, the intermediate approach has been popularly researched.

A detailed review of prior research about the intermediate approach will be given in the following section. However, it needs to be stressed that the intermediate approach is only successful when it is based on knowledge of delivery context, including target device capabilities and adaptation hint (e.g., wireless markup language) [1]. In our research, we have used the request header field of the HTTP to recognize delivery context. By using the header information, our system detects whether users are connecting to the Website using mobile devices or desktop devices, what carriers they use and therefore which wireless markup language should be used for content publishing.

2.3 Intermediate Content Adaptations

There are two different approaches for the intermediate content adaptations: content reformatting and content extracting adaptation. The former approach aims to reproduce the entire Webpage in a more convenient form, whereas the latter aims to extract useful and relevant content from the Webpage.

Content Extracting Adaptation

Gupta et al. [8, 14] propose a content extraction framework based on the Document Object Model (DOM). A DOM tree is formatted by parsing input HTML document. The system, called "Cunch1 and 2", navigates such a tree recursively, using customisable filters to remove and modify specific nodes, leaving only content behind. Rahman et al. [15] propose a technique that uses structural analysis, contextual analysis, and summarisation. The structure of an HTML document is first analysed and then deconstructed into a smaller subsections. The content of the individual sections can be extracted and summarized. Contextual analysis is performed with proximity and HTML structure analysis. Bar-Yossef and Rajagopalan [16] define content extraction as a frequent template detection problem, and suggest a solution for it based on counting frequency item sets. Lin and Ho [17] propose a content extracting system, called InfoDiscover. It partitions a page into several content blocks according to the HTML tag <TABLE> in a Webpage and separates the informative block from the redundant one based on the entropy value of each feature (terms). Yi and Liu [18, 19] propose a feature weighting technique, which builds a compressed structured tree to capture the common structure and uses an information based measure to evaluate the importance of each node in the compressed structure tree. Researchers of the content extracting adaptation approach usually focus not on the transformation from HTML to wireless markup language, but on eliminating redundant information from the HTML document. However, their approaches can be used for more efficient content adaptation by eliminating redundant or noisy information from the desktop Web document.

Content Reformatting Adaptation

The content reformatting adaptation system modifies both the content and the structure as well as transcoding them to the appropriate markup language [20]. Many approaches have been suggested for reformatting Web pages to fit the small screens of mobile phones and PDAs. Metter and Colomb [21] have described the problems discovered with the conversion of complex HTML documents into WML documents. The major problems are displaying tabulated data, hyperlinks, navigation aids, and user input. Similar issues are also discussed in [13]. Kaasine et al. [13] propose a method that converts HTML to WML according to the delivery context, resulting in simpler implementation and less maintenance. Lum and Lau [5] propose an intermediate content adaptation system in which a decision engine automatically negotiates the appropriate content adaptation decisions. The system uses those decisions to generate an optimal content version. Hwang et al. [4] propose structure-aware transcoding heuristics to preserve the original Webpage's underlying layout as much as possible. They have introduced two heuristics; the generalized outlining transform and the selective elision transformation. Their heuristics extract the relative importance of Web components from an intelligent syntactic analysis. Schilit et al. [20] have proposed a middleware proxy system, called m-Links. It retrieves Web documents using HTTP and delivers a suitable user interface to a variety of small wireless devices. Chi and Cao [22] and Mohan et al. [6, 23] have presented specific issues of multimedia adaptation, in which they stressed efficient data reuse on the proxy (Chi and Cao) and dynamic adaptability of multimedia according to a variety of different accessing devices (Mohan et al.).

Both content extracting and content reformatting adaptation require a high real-time overhead to transcode streaming HTML source code [22, 24]. It is hard to compare the performance of these two approaches because the performance depends on the algorithm used. One significant limitation of the content reformatting approach is that it does not eliminate redundant or noisy information from the desktop Web document. Therefore the adaptation results sometimes seem awkward from the user's point of view [4]. For this reason, we have employed the content extracting adaptation approach. Our system extracts relevant information by using structure based rules when it receives HTTP response messages.

3. WMS Architecture

The Web to Mobile Server (WMS), a content extracting adaptation system, mediates between various clients and Web servers. When it receives an HTTP request from a client's device, it detects the delivery context, including device type and servicing carrier. Figure 1 illustrates WMS system architecture. The WMS system consists of three sub-modules: the content extracting rule script; the script engine; and the markup language converter.

- The content extracting rule base stores rules for content extraction, which are created by the user with the rule management program. The rules include personalisation information and display structuring information.
- The rule engine reconstructs contents by using content extracting rules and the delivery context. The result of the rule script engine is independent from the markup language.
- The markup language converter transforms the rule script engine's result to the appropriate markup language according to the delivery context.

Figure 2 illustrates the operation process of the WMS system. Users may access the WMS system via desktop devices, mobile phones, and other mobile devices (❶). The WMS system obtains the delivery context, including the user's mobile device information (e.g. display size and colour) and carrier information, and requested URL information, using the protocol detector. After obtaining this information, the WMS system requests URL information from the Web server (❷). The corresponding Web server generates a HTML response message and sends it to the protocol detector (❸). The protocol detector then passes this HTTP response message to the selector with delivery context information.

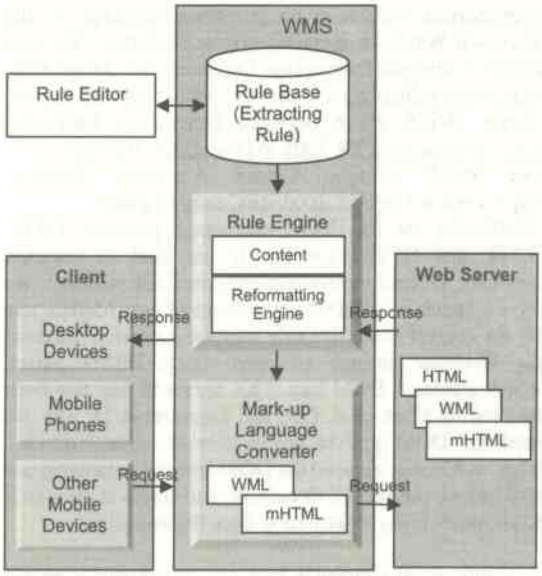


Figure 1. WMS System Architecture

The selector chooses relevant information from the HTTP response message by using content extracting rules and returns this result to the protocol detector (❹~❻). The protocol detector in turn sends this information to the translator. The translator performs appropriate markup language transformation, image transformation, paging and cashing (❼~❾). Lastly, the WMS system sends this result to the user.

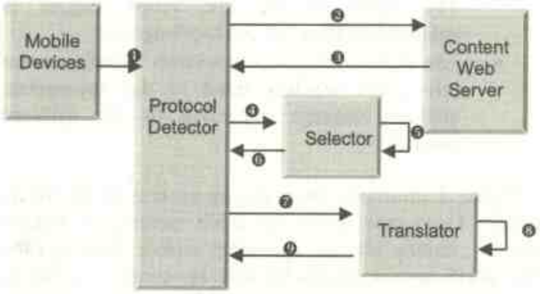


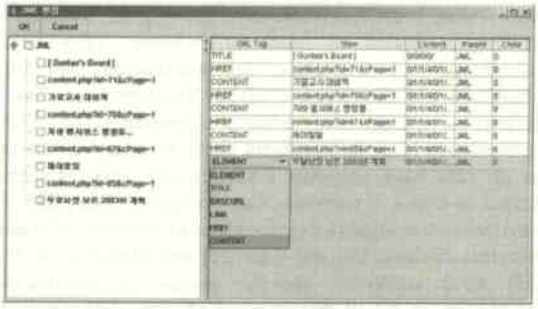
Figure 2. WMS Operation Process

4. Implementation

This Section describes an empirical implementation example of the WMS system. In this implementation, the WMS system mediates a Web Service request from a mobile device to the desktop Web Service. The system is implemented with Java programming language on the Microsoft Windows operating system platform. We used UDDI (Universal Description Discovery and Integration, http://www.uddi.org/) for service publishing and service search, WSDL (Web Services Description Language, http://www.w3.org/TR/2005/WD-wsdl20-20050803/), and SOAP (Simple Object Accessing Protocol, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/) for the XML messaging protocol. HTTP, SMTP, and HTTP over TCP/IP are used as transport networks. Overall implementations are J2EE standard. We used a specialised DOM (Document Object Model) tree for the content extracting rule creation process. We passed the Webpage through an open source HTML parser, which creates a DOM tree. This approach has also been adopted by Chen et al. [25] and Gupta et al. [8, 14]. We used the DOM tree to create content extracting rules, whereas Chen et al. used the DOM tree for organising and dividing up the HTML document, and Gupta et al. used it to eliminate noisy information from the Webpages

Figure 3 (a) illustrates how the administrator makes content extracting rules using the DOM tree. If the administrator submits a URL to be accessed, the HTML parser represents the parsing result as a DOM tree and shows the HTML sources in the source viewing window. Each node has enclosed content text and if the administrator selects one node to be served to a mobile device, it is saved in a mobile service list and used for service description for mobile devices. We used JML (Java Markup Language) editor to assign XML tags to these selected mobile service items. The types of XML tags include `TITLE`, `BASEURL`, `LINK`, `HREF`, `CONTENT`, and `ELEMENT`. Figure 3 (b) illustrates how the administrator can assign XML tags to the selected items. The selected items are displayed on the left side and the administrator can select an XML tag type from the XML tag list (on the right side).

(a) DOM Tree - Mobile Service Item Selection



(b) JML Editor – Assign XML tags
Figure 3. Content Extracting Rule Creation

Figure 4 shows a WSDL file that is produced by applying this content extracting rule. The markup converter module uses WSDL files to generate appropriate markup language like WML, mHTML, or cHTML, according delivery context information.

Figure 4. Content Extraction Result after applying Content Extracting Rule

Figure 5 illustrates the browsing shot as accessed by a mobile phone simulator. When users request mobile Web service from the wired Web site (Figure 5 (a)), the WMS system detects delivery context information and requests the wired Web Server, generates WSDL files by applying content extracting rules, and finally generates an appropriate wireless markup language file by using delivery context information. The WMS system's processing result is displayed in the user's mobile device (Figure 5 (b)). Though this example only shows how a user can access a wired Web site via a mobile phone, users can also gain access using a variety of other devices (e.g. PDAs).


(a) Wireless Service Request Page


(b) Response Message from WMS System

Figure 5. Example of Wireless Information Request to Wired Information Web Server

5. Conclusion and Further Work

In this paper, we have proposed a method that extracts mobile service content from wired Web documents by using content extracting rules. The rules are generated using a specific DOM tree and assigning its specific nodes as mobile service items. The JML editors are then used to assign XML tag to these items. By applying these rules, the system generates WSDL files on the fly by which appropriate wireless markup files are generated.

Although we have successfully implemented an intermediate content extracting adaptation system that supports dynamic content publishing according to delivery context, we still need to study the following issues:

- We have only explored physical context as delivery context in this research. However, we need to consider other context information such as communication context, user context, computing context, and time to provide more personalised mobile services.
- Although performance is absolutely critical in the content adaptation system, we have not conducted any performance evaluation. Therefore, we need to research the communication performance effect of the content adaptation system and minimization of computation overload for the content extraction process.
- There are various implementation issues associated with the HTTP form processing such as the HTTP "post" method processing, session management and java script processing. These forms of implementation should also be administered for a more complete content adaptation system.

References

[1] Butler, M., et al., Device independence and the Web. IEEE Internet Computing, 2002. 6(5): p. 81-86.
[2] Laakko, T. and T. Hiltunen, Adapting Web content to mobile user agents. IEEE Internet Computing, 2005. 9(2): p. 46-53.
[3] Butler, M.H., Current Technologies for Device Independent. 2001, Hewlett Packard.
[4] Hwang, Y., J. Kim, and E. Seo, Structure-aware Web transcoding for mobile devices. IEEE Internet Computing, 2003. 7(5): p. 14-21.
[5] Lum, W.Y. and F.C.M. Lau, A context-aware decision engine for content adaptation. IEEE Pervasive Computing, 2002. 1(3): p. 41-49.
[6] Mohan, R., J.R. Smith, and L. Chung-Sheng, Adapting multimedia Internet content for universal access. IEEE Transactions on Multimedia, 1999. 1(1): p. 104-114.
[7] Knutsson, B., et al., Architecture and Performance of Server-Directed Transcoding. ACM Transactions on Internet Technology, 2003. 3(4): p. 392.424.
[8] Gupta, S., et al., Automating content extraction of HTML documents. World Wide Web, 2005. 8(2): p. 179-224.
[9] Brezillon, P., Context in problem solving: a survey. Knowledge Engineering Review, 1999. 14(1): p. 47-80.
[10] Chen, G. and D. Kotz, A Survey of Context-Aware Mobile Computing Research. 2000, Dartmouth College: Hanover, NH, USA.
[11] Schilit, B., N. Adams, and R. Want, Context-aware computing applications. Proceedings. Workshop on Mobile Computing Systems and Applications (Cat. No.94TH06734), 1995: p. 85-90.
[12] Siau, K., E.P. Lim, and Z. Shen, Mobile commerce: promises, challenges, and research agenda. Journal of Database Management, 2001. vol.12, no.3: p. 4-13.
[13] Kaasinen, E., et al., Two approaches to bringing Internet services to WAP devices. Computer Networks, 2000. 33(1-6): p. 231-246.
[14] Gupta, S., et al. DOM-based content extraction of HTML documents. in International World Wide Web Conference. 2003. Budapest, Hungary: ACM Press New York, NY, USA.
[15] Rahman, A.F.R., H. Alam, and R. Hartono. Content extraction from HTML documents. in First International Workshop on Web Document Analysis (WDA2001). 2001. Seattle, Washington, USA.
[16] Bar-Yossef, z. and S. Rajagopalan. Template Detection via Data Mining and its Applications. in WWW 2002. 2002. Honolulu, Hawaii, USA.
[17] Lin, S.-H. and J.-M. Ho. Discovering Informative Content Blocks from Web Documents. in SIGKDD '02. 2002. Edmonton, Albert, Canada.
[18] Yi, L. and B. Liu. Web Page Cleaning for Web Mining through Feature Weighting. in Proceedings of Eighteenth International Joint Conference on

Artificial Intelligence (IJCAI-03). 2003. Acapulco, Mexico.
[19] Yi, L., B. Liu, and X. Li. Eliminating Noisy Information in Web Pages for Data Mining. in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2003). 2003. Washington, DC, USA.
[20] Schilit, B.N., et al., Web interaction using very small Internet devices. Computer, 2002. 35(10): p. 37-45.
[21] Metter, M. and R. Colomb. WAP Enabling Existing HTML Applications. in 1st Australasian User Interface Conference. 2000.
[22] Chi, C. and Y. Cao. Pervasive Web Content Delivery with Efficient Data Reuse. in 7th International Workshop on Web Content Caching and Distribution (WCW). 2002. Boulder, Colorado.
[23] Moran, T.P. and P. Dourish, Introduction to this special issue on context-aware computing. Human-Computer Interaction, 2001. 16.
[24] Canali, C., V. Cardellini, and M. Colajanni. Cooperative Architectures and Algorithms for Discovery and Transcoding of Multi-version Content. in 8th International Workshop on Web Content Caching and Distribution. 2003. IBM T.J. Watson Research Center, Hawthorne, NY USA.
[25] Chen, Y., W. Ma, and H. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. in 12th international conference on World Wide Web. 2003. Budapest, Hungary: ACM Press New York, NY, USA.

Authors

박길철(Gil-Cheol Park)
1979-1983 한남대학교
1983-1985 숭실대학교대학원(석사)
1994-1998 성균관대학교대학원(박사)
1985-1990 삼성종합기술원
1991-1996 대교컴퓨터
1996-1998 한서대학교
1998-현재 한남대학교 교수
2005.1-현재 Visiting Professor of Tasmania State Univ., Australia

김양석(Yang-sok Kim)
1987-1995 서울시립대학교
1994-2001 현대정보기술
2001-2002 E -2 Corporation
2002-2003 Tasmania State Univ., Australia
2003-2004 Tasmania State Univ., Australia (Master of Computing)
2005. 2-현재 Tasmania State Univ., Australia 박사과정 재학 중

강병호(Byeong-ho Kang)
1982-1988 부산대학교
1988-1990 Tasmania State Univ., Australia (Master of Computing)
1990-1995 New South Wales Univ., Australia (PhD)
1995-1996 히타치 연구소,일본, 초빙연구원
1996-1999 호서대학교, 교수
2000-현재 Tasmania State Univ., Australia, 교수