Journal of Web Engineering, Vol. 13, No.1&2 (2014) 087-096

C1: AN AUTOMATED ONLINE EDUCATION MANAGEMENT SYSTEM BASED ON AN OBJECT-ORIENTED APPROACH

S.C. NG, T.S. LEE, C.K. WONG , F.Y. LEE

School of Information Technology, SEGi University,

No.9, Jalan Teknologi, Taman Sains Selangor Kota Damansara, PJU 5, 47810, Petaling Jaya,

Selangor, Malaysia. ashleyng@segi.edu.my

Received October 2, 2012
Revised September 22, 2013

Communication and information are integral parts of the education process and thus the integration of Information and Communication Technologies (ICT) in education management systems comes naturally. Many business organisations, schools and universities are gradually increasing the use of ICT to reduce costs and improve the efficiency of administration. This paper presents the implementation of an automated online education management system, named Cybernetics 1 (C1). C1 was tested as an operational tool to support the education process of undergraduate students. The four major modules in C1 include the registration of new students (inclusive of retrieving and updating of student profile), creation of module assessment structures, module marks entry and generation of academic transcripts. C1 was developed based on the object-oriented approach. This system successfully eliminates the problem of poor data sharing and data inconsistency as encountered by most existing education management systems. In particular, the normalisation process has significantly reduced the problems of data redundancy in the database.

Key words: Education management system, Information and Communications Technology (ICT), object-oriented approach, normalisation.
Communicated by: G.J. Houben & Q. Li

1 Introduction

Information and Communication Technologies (ICT) are diverse sets of tools and resources that the society in general uses extensively to communicate, create, disseminate, store and manage information [6]. Since communication and information are two important aspects in the education process, it inherently motivates the extensive use of ICT in education management systems. Application of ICT in education serves two main purposes [7]: for the administration of routine tasks, and for instruction. A survey carried out in United States by the National Centre for Education Statistics [25] showed that the usage of Internet by education staff in US public schools for administrative record keeping is 34% after the creating of instructional materials (39%). The application of Internet to accomplish semi-administrative task such as the use of computers to communicate with colleagues was 23%. It was observed that collaboration is highly needed in academic work environments for the improvement of the learning experience as well as in the increase of efficiency in administrative tasks. Based on the preliminary research recent five years, the major pitfalls for human-based education management

systems are poor information sharing among departments and disassociation of data outside the system [15, 19]. The latter problem can result in much time and effort consumed for tracing the source data if a minor data input mistake is detected. Nevertheless, the problem of superintended service objects [11, 21] remained on most of the existing computer-based education management systems. In view of the need to improve the effectiveness of the current education management system, Cybernetics 1 (C1) was developed to serve as an alternative for automating most of the processes in the management of undergraduate students. Besides maintaining the consistency of input and storage of data, C1 exhibits a high level of flexibility in allowing the lecturers to customise the assessment structure of subjects in accordance to a particular degree's program structure. C1 was implemented using an object-oriented approach, which is a known software engineering concept in the software development process to ensure that development can be done in a timely and cost-effective manner. This study aims to present the implementation of C1 and the feasibility of adopting an object-oriented approach on web application development, particularly that which pertains to the development of an education management system.

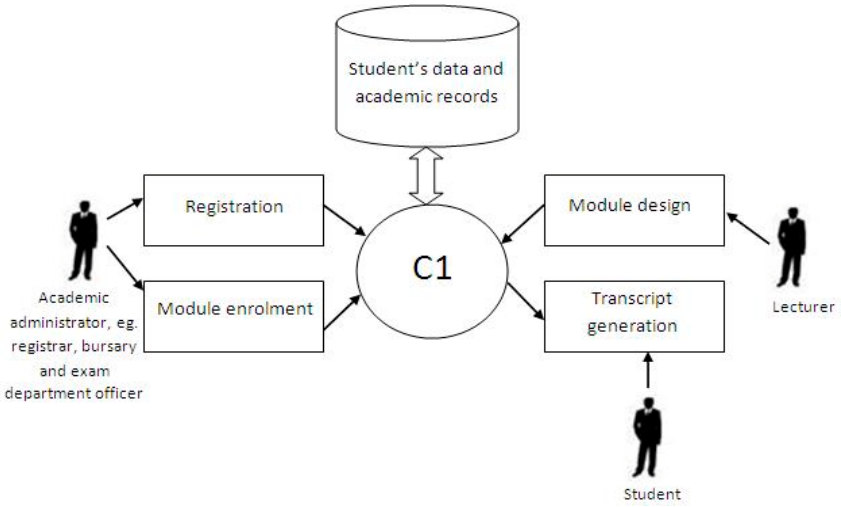


Figure 1 Implementation schema of C1 as an automated online education management system

2 Related Works

Since the concept of networking for data sharing and connectivity was realized, researchers started to explore the technology to use networking such as LAN, WAN and WWW as a medium to manage information system in an organization. As a result, new applications and methodologies are introduced, with the majority being typically web based [14]. Web-based applications are flexible. The implementation of web application is not platform specific. Users can access the application by using web browsers. Three prominent examples of web-based application in education are education management system (EMS) [11], course management system (CMS) [29] and learning management system (LMS) [10, 27]. EMS is a platform that supports both collaborative learning and institutional

administration and management tasks. Nevertheless, the operation nature of the system still depends on the institution needs. In web-based system development, the trade-off between re-inventing the wheel and utilizing existing system framework has always been the difficult choice for developers. Comparison on possible solutions is thus needed. Takago et al. [29] developed a CMS to unify the multiplicity modules in a department as well as the diversified human interfaces of the existing system. In the previous work of Andreica and Agachi [3], an integrated web-based application has been designed to serve the purpose as a high accessible and user-friendly tool for quantitative data analyses, syntheses and evaluations. Another similar work by Duan and Zhang [11] is a system with the characteristics of high integration, unified data platform, flat organization's structure and decision support feature. It can be observed that these in-house web-based applications were designed to solve the problems of non-centralized information across departments, poor infrastructure of data sharing and integration, different requirements and processes among departments and management levels [10, 17, 23]. Despite customization, the major concern to develop an EMS is the amount of time spent on the development process.

Besides the custom-developed EMS as discussed, a number of ready-made framework such as WebCT [12], Moodle [26], SAKAI [2] and ILIAS [20] have been integrated into the educational environment. Adopting existing system framework saves significant amount of development time and effort. However, since the main aim of these system frameworks (e.g. WebCT, Moodles) is to encourage student's online learning behavior, their fitness to serve the education management purpose has yet to be explored. Rößling [27] found that Moodle is lack of automatic assessment, feedback and program plagiarism detection features to support instructors. Al-Ajlan and Zedan [1] described the diversity and sophisticated features of which Moodle provides. But due to the complexity of the LMS, it has been evaluated as a complex platform for normal users and it needs to be operated by experts or experienced technicians. It is time consuming for the instructors and students to learn and adapt to the system framework. Training is thus required for different level of users. Moreover, the notation and methodology of existing system framework most of the time do not precisely match those of the course. In terms of flexibility, although instructors are able to create classes and upload content to the server, the system does not provide enough flexibility for the instructor to configure the assessment breakdown component (e.g. a module is split into 30% coursework assignment, 20% time constraint test and 50% final examination). Due to this fact, it is difficult for instructors to trace and synthesis the class performance. The admin are not able to generate constructive report for the class or course performance evaluation. To enhance the institution and student's teaching-learning experience, an automated online EMS (C1) was developed by using object-oriented approach.

3 Object Oriented Approach in Web Applications Development

The combination of navigation through heterogeneous information space and query operations via cross-platform access has made World Wide Web (WWW) the platform for a new generation of information systems. Web applications are based on the concept of thin clients and centralised maintenance that facilitates the real-time deployment of software updates at minimal cost [13]. While web-based applications have gained popularity due to their deployment advantage on client-server platforms, the development of web-based applications is still mostly ad hoc [16]. In other words, there is no single structured or systematic development methodology for web application development that is supported by suitable development tools. Most of the web application development and management

highly rely on the individual developer's knowledge, experience and approach [22]. Web application development typically encompasses four aspects: functionality, content, hypertext and presentation. Traditional software engineering methodologies are deficient in the modelling hypertext and web-based content. Incorporating hyperlinks into the interface would be a problem for web developers as the traditional software development methodologies do not provide the notion of hyperlinks and content modelling. Additionally, the lack of abstraction capabilities in these methodologies caused difficulties for the developer in reusing previously developed artefacts [31]. Difficulties also exist in the construction of frameworks capable of capturing the common features of classes used for rapid development and customization.

The concept of abstraction is critical to web application development as web applications are complex due to the blistering grow of application size and its evolving nature. Reusing previous design knowledge could minimise the error rate and ensure the efficiency of the development process. The object-oriented approach is a software development paradigm that models real world phenomena such as objects and classes [8]. Object-oriented principles such as abstraction, modularity and encapsulation facilitate the reuse of prefabricated components. In web applications design, object-oriented model uses components as a uniform concept to model the web entities at arbitrary levels of granularity [5]. For example, the content on a web page can be captured as an independent design artefact unit. It is noted that these components have such flexible grain size that they can model web entities as small as individual hyperlinks or layout fragments in CSS and XSL. These components can associate with other components to model aggregation (has-a) or inheritance (is-a) relationships. In addition, they can also be associated with other complete resources such as HTML documents and scripts that generate web documents. This flexibility is well-suited for the web application development process, which requires an approach based on high-level design and low-level implementation [13].

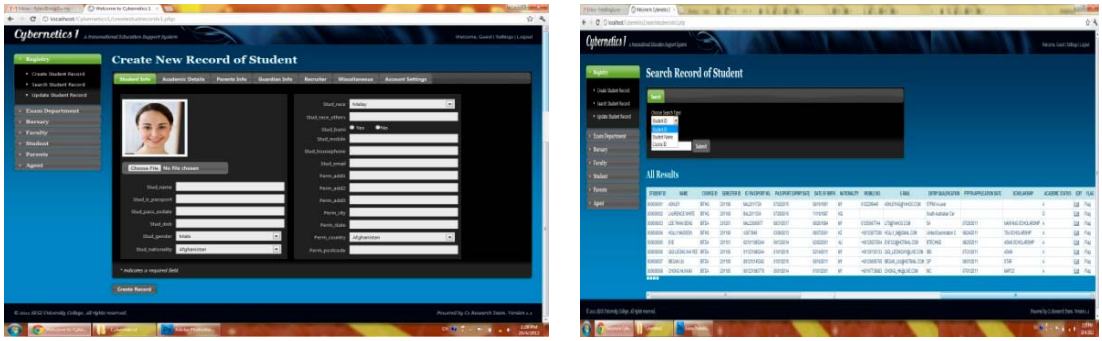
4 C1 System Description

C1 is an automated online education management system (C1) developed using the object-oriented approach. C1 links the registry, bursary, faculty and examination departments and synchronises the data shared among them. The actors of the system are academic administrators, lecturers and students. This system serves different purposes depending on the requirements of the actors. The academic administrators will use C1 to process and store students' data and academic records. Lecturers on the other hand, will use C1 as a tool to design the module assessment structure. Students can check their academic statuses and results from this system (Figure 1). There are four major modules in C1: (1) registration (2) module assessment structure design (3) module enrolment and (4) transcript generation. C1 models the operation of undergraduate students' education process flow, as proposed by Ng et al. [24]. At the beginning of first semester, the Registrar will create an account for each new student upon the admission of students into the institution. The registration module monitors the admission, termination and suspension of students (Figure 2).

The module assessment structure design component (Figure 3) allows lecturers to define the assessment structure based on a particular degree programme's approval documentation. Lecturers can dynamically alter the assessment structure of subjects online upon the updates to the degree programme's approval documentation. Students then enrol in the subjects of their current semester via

the module enrolment component. By keeping the enrolment records, C1 is able to generate student attendance lists for the lecturers.

Multiple-criteria queries are implemented to quicken the task of accurate record retrieval. Prior to examination, C1 verifies the student's payment status and the subjects registered so that the examination department officer could decide if a student is eligible to sit for the final examination. At the end of the semester, lecturers will enter coursework and examination marks of the students into C1, in according to the assessment structure that has been defined earlier in the semester. The transcript generation module computes the marks of each subject and generates the grades that students achieve. Students can view their semester results by downloading the electronic transcript (*e-transcript*) from C1.

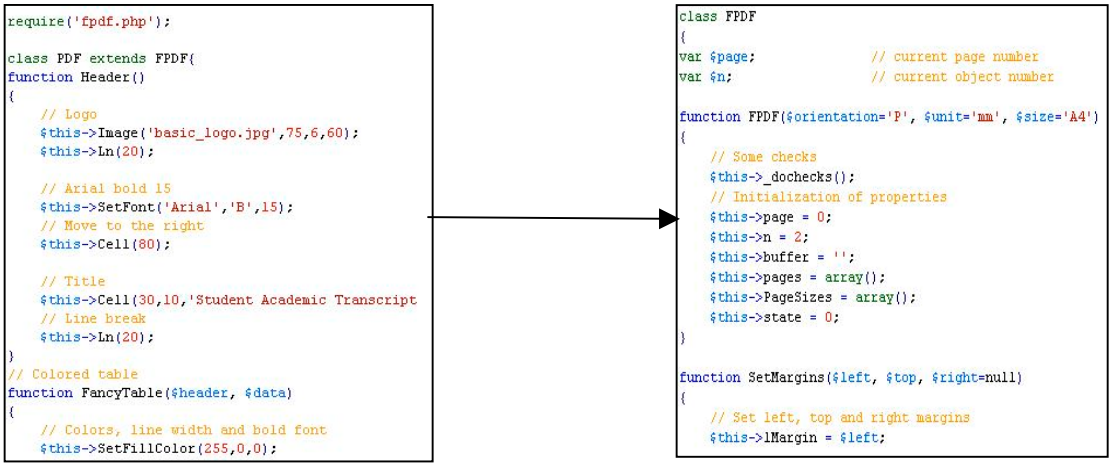


(a) (b)

Figure 2 GUI (a) new student registration page (b) data component retrieval page

(a) (b)

Figure 3 GUI (a) module component design page (b) assessment structure definition page



```
require('fpdf.php');

class PDF extends FPDF{
function Header()
{
    // Logo
    $this->Image('basic_logo.jpg',75,6,60);
    $this->Ln(20);

    // Arial bold 15
    $this->SetFont('Arial','B',15);
    // Move to the right
    $this->Cell(80);

    // Title
    $this->Cell(30,10,'Student Academic Transcript
    // Line break
    $this->Ln(20);
}
// Colored table
function FancyTable($header, $data)
{
    // Colors, line width and bold font
    $this->SetFillColor(255,0,0);
```

```
class FPDF
{
var $page;               // current page number
var $n;                  // current object number

function FPDF($orientation='P', $unit='mm', $size='A4')
{
    // Some checks
    $this->_dochecks();
    // Initialization of properties
    $this->page = 0;
    $this->n = 2;
    $this->buffer = '';
    $this->pages = array();
    $this->PageSizes = array();
    $this->state = 0;
}

function SetMargins($left, $top, $right=null)
{
    // Set left, top and right margins
    $this->lMargin = $left;
```

Figure 4 Inheritance relationship between *PDF* (child class) and *FPDF* class

5 Object-Oriented Programming

C1 was developed by using the PHP language [28] and Javascript that supports the object-oriented approach. Contrary to the conventional paradigms that separate data structure and procedures, object-oriented programming wraps procedures inside the data structure and defines it as an object or class. This method, known as *dynamic binding* enables the data to be passed to the appropriate procedures at runtime as well as allowing the programmer to control the access to the data by other classes [1]. Segments of code that perform similar tasks can be generalised as a class so that objects instantiated from the class can call the same procedures from that class without redefining the procedures.

For example, during the C1 implementation, the transcript generation module (*PDF* class) was defined directly based on the definition of *FPDF* class. Thus, the *PDF* class (child class) can use the functionality of the parent class, *FPDF* via the inheritance capability of object-oriented programming (Figure 4). This programming approach benefits the implementation of C1 in the way that the parent class can always maintain their state information while many child classes can be instantiated dynamically by the parent class, thus providing the ability to reuse program codes. With this, interactions between objects such as message sending can also be supported. It was discovered during the testing process that the *PDF* class achieved faster printing time in comparison to the common method of importing the dynamic link library—one used in existing information systems

6 Database Normalisation

C1 was deployed centrally with all the data stored at centralised facility. Users can access the data through web browsers. For the data model of this system, tables were created based on the attributes extracted from the forms used in the manual system. The main aim for the structure of the C1 database

was to minimize data redundancy. To eliminate the update, deletion and insertion anomalies, normalisation of the tables was performed. The database of this system was normalised using the normalisation method introduced by Boyce-Codd [18]. Generally, normalisation is the process of breaking up larger relations into many smaller ones based on a set of rules [30]. The process involves identifying functional dependencies among the attributes in the table. New relations will be generated if there are attributes which are not directly dependent on the primary key. Theoretically, there are 5 normal forms for the process of normalisation in which the un-normalised table progresses from the First Normal Form to the Fifth Normal Form [9]. For the C1 database, all the relations were designed to comply with 3NF. This achieved the main aim of removing anomalies and improving the maintainability of the database for future updates.

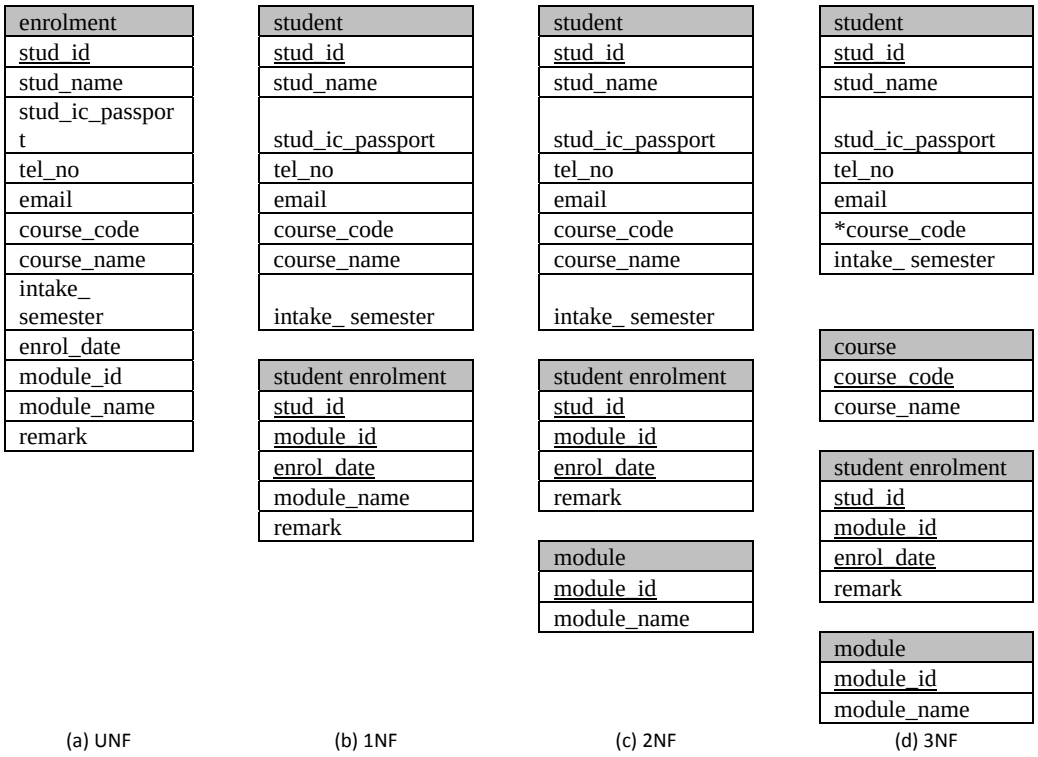


Figure 5. Normalisation process based on the student enrolment form

Figure 5 demonstrates the normalisation process for a small part of the system's database, which was based on the student module enrolment form. The un-normalise relation, as shown in Figure5 (a) was first converted to 1NF (Figure5 (b)) by removing the repeating attributes. The partially dependence attributes was then removed from 1NF to transform the relation into 2NF. Finally, the 3NF as shown in Figure5 (d) was achieved by removing transitive dependence attributes in 2NF (Figure 5(c)). There were in total 4 relations produced at the end of the normalisation process of this particular

example. For the overall C1 database, there are in total twelve relations created after the normalisation process.

7 Conclusion

This paper discusses the implementation of an automated online education management system (C1) using the object-oriented approach. C1 links the registry, bursary, faculty and examination departments and synchronises the data shared among them. This method successfully resolves the problem of poor data sharing and data inconsistency among the departments. There are four major modules in C1: registration, module assessment structure design, module enrolment and transcript generation. Serving as an academic support tool, C1 allows lecturers to design their module assessment structure. Upon any restructuring of module assessments, lecturers can alter the assessment structure flexibly online. In addition, C1 supports the real-time generation of student academic transcript (*e-transcript*) based on the automated grading function. Two methods were utilised for generating the e-transcript: importing the dynamic link library that most of the existing information systems are utilising and creating a class for formatting and printing the transcripts. The latter outperformed the method of importing the dynamic link library when it comes to achieving a shorter time in generating the e-transcripts online. Normalisation has also been carried out on the twelve tables to eliminate the redundancy and inconsistency of data in the C1 database. For future improvements, the C1 research team intends to incorporate decision support features to enhance the functionality of the system as an effective education management system.

Acknowledgements

This work is supported by the SEGi Internal Research Fund under the project no. SEGi/2011/TK/04/1.

References

1. Al-Ajlan, A. and Zedan, H. (2008) Why Moodle. In Proceedings of the 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, 58 – 64.
2. Albarrak, A.I., Aboalsamh, H.A. and Abouzahra, M. (2010) Evaluating Learning Management Systems For University Medical Education. In Proceedings of the International Conference on Education and Management Technology, 672 – 677.
3. Andreica, A. and Agachi, S. (2009) Academic Research by Means of An Integrated Software System. In Proceedings of IADIS Multiconference on Computer Science and Information System, 150 – 154.
4. Arisholm, E., Briand, L.C., Foyen, A. (2004) Dynamic Coupling Measurement for Object-Oriented Software. IEEE Transactions on Software Engineerin 30(8), 491-506.
5. Bertino, E., Camossi, E., and Bertolotto, M. (2009) Multi-Granular Spatio-Temporal Object Models: Concepts and Research Directions. Moira C. Norrie and Michael Grossniklaus (Eds.). Berlin: Springer-Verlag, 132-148.
6. Blurton, C. (1999) New Directions of ICT-Use in Education. World Communication and Information Report, Paris: UNESCO, 46-61.
7. Cairncross, F. and Pöysti, K. (2006) *ICTs for Education and Building Human Capital.* International Telecommunication Union.
8. Capretz, L.F. (2003) A Brief History of The Object-Oriented Approach. ACM SIGSOFT Software Engineering Notes 28(2), 6.

9. Chikkamannur, A.A. and Handigund, S.M. (2011) Design of Normalized Relation: An Ameliorated Tool. International Journal of Computing & Information Sciences 9(1), 28-33.
10. Darbhamulla, R and Lawhead, P. (2004) Paving the Way Towards An Efficient Learning Management System. In Proceedings of *the 42th Annual Southeast Regional Conference.*
11. Duan, R. and Zhang, M. (2007) Design of Web-Based Management Information System for Academic Degree and Graduate Education. Integration Innov. Orient E-Soc.(252), 218-226.
12. Ebenezer, J.V., Lugo, F., Beirnacka, B. and Puvirajah, A. (2003) Community Building Through Electronic Discussion Boards: Pre-Service Teachers' Reflective Dialogues On Science Teaching. *Journal of Science Education and Technology 12*(4), 397-411.
13. Hans-W. Gellersen and Gaedke, M. (1999) Object-Oriented Web Application Development. *IEEE Internet Computing* 3(1), 60-68.
14. Hansen, S. (2002) Web Information Systems: The Changing Landscape of Management Models and Web Applications. In *Proceedings of the 14th International Conference on Software Engineering And Knowledge Engineering*, 747-753.
15. Homeed, T.S.K. and Mahmood, A. (2006) Systematic Multimedia Courseware Development Using a Software Engineering Style Process Model. Information Technology Journal (5), 726-734.
16. Huang, G.Q., Mak, K.L. (2001) Issues in the Development and Implementation of Web Applications for Product Design and Manufacture. International Journal of Computer Integrated Manufacture 14(2), 125-135.
17. Huang, H. and Cai, Y. (2010) Web-based Design of the Management Information System for the Chemical Laboratory in the University. In Proceedings of the International Conference on Computer Design and Applications 5, 292 – 293.
18. Hussain, T., Shamail, S. and Awais, M.M. (2003) Eliminating Process of Normalisation in Relational Database Design. In Proceedings of the 7th Multi Topic International Conference, 408-413.
19. KaiLiang Liang and Zhigao Lan (2010) Local Universities of China Management Information System Integration Model Design. In Proceedings of International Symposium on Intelligence Information Processing and Trusted Computing, 18-20.
20. Lemnitzer, L., Vertan, C., Simov, K., Monachesi, P., Kiling A., Cristea, D. and Evans, D. (2006) Integrating Language Technology and Semantic Web Techniques in eLearning. In Proceedings of International Conference on Interactive Collaborative Learning, 666-672.
21. Li, H. and Q. Liu, (2008) Design and Implementation of Educational Information Resource Management System Based On SOA. In Proceedings of the International Symposium on Knowledge Acquisition and Modeling, 424-428.
22. Murugesan, S., Deshpande, Y. Hansen, S. and Ginige, A. (2001) Web Engineering: A New Discipline for Development of Web-Based Systems. Web Engineering, Software Engineering and Web Application Development, San Murugesan and Yogesh Deshpande (Eds.). London: Springer-Verlag, 3-13.
23. Mutholib, A., Gunawan, T.S. and Kartiwi, M. (2011) Development of FYP Online System for Outcome-based Education. In Proceedings of IEEE Region 10 Conference, 1312 – 1316.
24. Ng, S.C., Wong, C.K., Lee, T.S. and Lee, F.Y. (2011) Design of an Agent-based Academic Information System for Effective Education Management. Information Technology Journal (10), 1784-1788.
25. Rowand, C. (2000) Teacher Use of Computers and the Internet in Public Schools. Stats in Brief, DC: National Center for Education Statistics (Ed.).
26. Rößling, G. and Kothe, A. (2009) Extending Moodle to Better Support Computing Education. *SIGCSE Bull* 41(3), 146-150.

27. Rößling, G., Joy, M., Moreno, A., Radenski, A., Malmi, L., Kerren, A., Naps, T., Ross, R.J., Clancy, M., Korhonen, A., Oechsle, R. and Iturbide, J.Á.V. (2008) Enhancing Learning Management Systems To Better Support Computer Science Education. *SIGCSE Bull* 40(4), 142-166.
28. Suzumura, T., Trent, S., Tatsubori, M., Tozawa, A. and Onodera, T. (2008) Performance Comparison of Web Service Engines in PHP, Java and C. In Proceedings of IEEE International Conference on Web Services, 385-392.
29. Takago, D., Matsuishi, M., Goto, H. and Sakamoto, M. (2007) Requirements for a Web 2.0 Course Management System of Engineering Education. In Proceedings of the 9th IEEE International Symposium on Multimedia Workshops, 435 – 440.
30. Wu, Y.H. (2004) Normalisation Design of XML Database Schema for Eliminating Redundant Schemas and Satisfying Lossless Join. In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, 660 – 663.
31. Zhang, H.M. and Ben, K.R. (2010) Architectural Design of the Health Watch System with an Integrated Aspect-oriented Modeling Approach. In Proceedings of the International Conference on Computer Design and Applications (1), 624-628.