ISSN: 1545-679X

Information Systems Education Journal

Volume 8, Number 24 http://isedj.org/8/24/ June 9, 2010

In this issue:

Building an Integrated Student Information System in a K-12 School System

Annette Lerine Steenkamp
Lawrence Technological University
Southfield, Michigan, MI 48075-1058 USA

Abdelraheem Basal
Lawrence Technological University
Southfield, Michigan, MI 48075-1058 USA

Abstract: The task of managing an IT system in a school environment poses unique challenges. For example, one of the greatest challenges facing individual schools is the lack of integration between various information systems. The present situation in many schools is that there are many disconnected systems managing many different tasks. Systems with differing levels of functionality run independently of one another, causing multiple problems for the school's overall IT system. Many difficulties arise from inconsistently registered data, duplicate manual data entry, the extra time needed to manage multiple user accounts for one user, and non-productive time spent on technical support. In addition to these problems, the task of maintaining each individual system is time consuming. The challenges described above have prompted this research study. This study required an examination of the system architecture of several typical K-12 School systems, and a comprehensive understanding of the business and instructional needs of K-12 education.

Keywords: Learning Management System, Student Information System, Content Management System, Web 2.0, MVC Framework, SaaS, Cloud Computing

---

Recommended Citation: Steenkamp and Basal (2010). Building an Integrated Student Information System in a K-12 School System. *Information Systems Education Journal, 8* (24). http://isedj.org/8/24/. ISSN: 1545-679X. (A preliminary version appears in *The Proceedings of ISECON 2009:* §3554. ISSN: 1542-7382.)

This issue is on the Internet at **http://isedj.org/8/24/**

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: http://isedj.org/ • Contact person: Don Colton (editor@isedj.org)

This paper was in the 2009 cohort from which the top 45% were accepted for journal publication. Acceptance is competitive based on at least three double-blind peer reviews plus additional single-blind reviews by the review board and editors to assess final manuscript quality including the importance of what was said and the clarity of presentation.

Building an Integrated Student Information System in a K-12 School System

Annette Lerine Steenkamp
Steenkamp@ltu.edu

Abdelraheem Basal
abasal@datapublix.com

College of Management
Lawrence Technological University
Southfield, Michigan, 48075-1058, USA

Abstract

The task of managing an IT system in a school environment poses unique challenges. For example, one of the greatest challenges facing individual schools is the lack of integration between various information systems. The present situation in many schools is that there are many disconnected systems managing many different tasks. Systems with differing levels of functionality run independently of one another, causing multiple problems for the school's overall IT system. Many difficulties arise from inconsistently registered data, duplicate manual data entry, the extra time needed to manage multiple user accounts for one user, and non-productive time spent on technical support. In addition to these problems, the task of maintaining each individual system is time consuming. The challenges described above have prompted this research study. This study required an examination of the system architecture of several typical K-12 School systems, and a comprehensive understanding of the business and instructional needs of K-12 education.

Keywords: Learning Management System, Student Information System, Content Management System, Web 2.0, MVC Framework, SaaS, Cloud Computing

1. INTRODUCTION

More than ever before, K-12 schools are adopting information systems to improve and automate many processes that were once manual. With this large-scale movement towards IT systems and the increasing pressure on schools to use technology to improve student performance, many schools are looking for a learning management system as a way to enhance student learning: Recent surveys show that K-12 online learning is a rapidly growing phenomenon (Powell, 2008).

Learning management systems have already been implemented and used widely in institutions of higher education. Many U.S. states, including Michigan, require an online experience for all graduates (Michigan Department of Education, January, 2006).

The introduction of learning management systems into mainstream K-12 schools is expected to solve some problems in this education sector, but it also poses challenges, such as the major challenge of integrating systems. Many factors make integration of disparate systems used in K-12 schools a complex task. Different information technology architectures, software and hardware systems, network platforms, legacy systems, data structures, and applications already in place increase the complexity of the integration process (Perks and Beveridge 2003).

Such challenges apply equally to schools in small and large school districts working with

information systems. The failure to integrate their information systems is causing schools many headaches due to inconsistent data, duplicate manual data entry, extra time needed to manage multiple user accounts for individual users, and non-productive time spent on technical support. In addition, the task of maintaining all of the individual systems is time consuming.

Once a school system has all components of the disparate systems integrated, with the ability to access all of them from a single sign on (SSO), the tedious task of keeping track and maintaining the different systems will disappear. This paper explains research on building an integrated student information system with a Learning Management System, File Management System, and Content Management System for K-12 schools. The proposed system will achieve the highest level of data quality because a single source of data will be implemented and maintained. Section 2 presents the context for the research, Section 3 describes the research approach, Section 4 presents the approach to redesigning the architecture for the integrated system, with some modeling artifacts. Section 5 presents the demonstration of concept, a prototype), and Section 6 concludes with the summary and conclusions of the project.

2. RESEARCH CONTEXT

Need for Study

A solution is needed to address the issues discussed in the Introduction. This solution would include building an online environment in which students, parents, teachers and school administrators can share information and outcomes, while also complying with regulatory requirements and school policies in a transparent manner. The concerns of all stakeholders should be addressed in the form of an integrated enterprise architecture, which can result in a cost effective, adaptable and scalable solution. A clear roadmap of how to transform the individual systems of a disparate architecture into an integrated system based on a services orientation is also needed. Specifically, research should be aimed at designing a conceptual architecture for the next generation of integrated School Information Systems. This project was based on the Software-as-a-Service (SaaS) model utilizing Web 2.0 technologies.

Research focus

Addressing the problems stated above from an architecture point of view required an analysis of all systems in the K-12 environment. A Management Checklist (Perks & Beveridge, 2003) was used to identify systems integration problems in nine charter schools in southeast Michigan and northwest Ohio. The answers to these questions, presented in Table 1, lead us to take a holistic approach to examining the problems that schools face in the area of systems integration.

Integration related Questions	
Do your customers complain about your inability to maintain accurate and consistent information about them?	No
Do you find that changes required in one system manifest themselves in costly changes in other systems?	Yes
Do you find that integrating electronic information from customer and partners is costly and lacks integrity?	Yes
Do management information reports represent an inconsistent view of the current operational state of the organization?	Yes
Are there problems with internal business units communicating with each other electronically?	No
Has there been a lack of success in developing a corporate-wide shared knowledge base?	No
Do IT projects sponsored directly by the business exhibit integration and quality problems when introduced into the IT environment?	Yes
Is the integration between legacy and contemporary systems ineffective and costly?	Yes

Table 1: Management Checklist Answers to Integration Problem

Background of the Study

At the time of writing, several disparate systems are being used in K-12 charter schools in southeast Michigan. They are summarized in this section. The current system consists of several applications. A summary of these applications provides some indication of each system's functionality.

Learning Management System (LMS): The main functions of this system include course management, the ability to upload content, online student-teacher interaction, and student grading.

File Management System: This system allows all files to be stored and accessed from one secure server. The schools use the MS Windows 2003 file server. The main challenge facing this system is that its users need to link these file types to their websites and to access their files from home or off-the school campus.

Library System: This system keeps track of how students check materials out from the Media Center.

Special Education System: This system helps keep track of a student's Individualized Education Program (IEP), and progress made on a yearly basis. It also helps to track which students use different special services, such as speech, occupational therapy, and the school's resource room.

Student Information System: The web based Student Information System has been used since 2001 and is currently serving 9 schools with more than 300 staff and 3000 students. The SIS provides the functionality shown in Figure 1.

Figure 1. Current Student Information System Functionality

Assessment System: This system keeps track of student scores in various tests taken outside the local grading system.

Curriculum Management System: This system allows teachers to access material that has previously been taught, and helps them to map out the curriculum that should be taught for the upcoming months or year.

Cafeteria System: This system deals with student transactions involving meals, keeps files on free and reduced lunches, and is a Michigan State requirement.

Content Managements System: This system maintains the school website.

Problem Analysis

Multiple applications in school information systems create problems because of the lack of communication between them. The multiple systems collect the same data numerous times to meet the immediate need of each department. This constant collection of the same data creates errors, data-reconciliation challenges, and reporting discrepancies (Fruth, Larry, Michael, & Elizabeth, 2007). For example, if you have one teacher named Mr. Smith, he has account/user information in each system within the school. He has an account within the Student Information System, testing system, e-mail system, file management system and several other systems that might be needed. The fact of having multiple systems in one school creates a major dilemma when it comes to entering user information within each system. When one system is updated, the other systems are not, since the systems are not integrated. Appendix 1 shows how the various systems within the school interact with one another, and the types of relationships between them. All of the relationships are shown with interconnecting lines, and are all manual in nature.

The existing systems in the charter schools lack the following functionality:

- Student Connect module: A place where students will be able to login to check their grades and assignments online.
- Parents Connect module: A place where parents will be able to check their child's grades, assignments, discipline, attendance, learning materials, and progress online.
- Administrator/Staff Connect module: A place where administrators, teachers, and technical support staff can sign on to a single port where they can access all systems without the tedious task of remembering various passwords to sign

on for different school locations and sites.

- The system needs to support multiple data from multiple years and school rolls with the same user login.
- The bottom line: It is difficult for every school in this study to maintain their websites, and for individual content management system users to keep track of multiple user names and passwords.

To add to these requirements, there were multiple requests from school management teams to add the following sub-systems:

- A course and learning management system to enhance student learning.
- A website for every teacher.
- A course website to manage curriculum content.

Due to the complexity of the problem domain, the focus in this paper is on the four main systems, namely the Student Information System, the Learning Management System, the Content Managements System, and the File Management System, highlighted in yellow in appendix 1. These systems have similar cross functionality. For example, a student's information is repeated in every one of the following systems:

- Student information system
- File management system
- Learning management System
- Content management system

3. RESEARCH APPROACH

A positivist-empirical research approach was followed for this project, based on observation and interventions using several methods (Remenyi et al., 1998; Curtis and Cobham, 2002). This approach is appropriate for research into the phenomena, processes, and behaviors that are the basis of the K-12 educational environment. The action research paradigm formed part of the research approach since the primary author was engaged in the domain of discourse and participated actively in all the research processes.

The research life cycle is illustrated in Appendix 2, which depicts the research processes conducted in this study, and is summarized in brief as follows:

1. Identified and verified research problem and identified scope of research in terms of integrated school system.
2. Reviewed literature related to K-12 educational systems focused on the learning management systems, student information system, content management system and file management system.
3. Conceptualized insight into the research field of focus, based on literature and empirical experience.
4. Drafted the design of conceptual solution and verified the conceptual model.
5. Demonstrated the conceptual model by refining it into a conceptual solution and a prototype using:
    - Business case
    - Use cases models
    - Use cases specifications
    - UML diagrams for business scenarios
    - Class models
6. Evaluated prototype and refined conceptual solution.
7. Documented findings and drew conclusions.

4. REDESIGNING THE ARCHITECTURE

The architecture approach followed to redesign the integrated school system architecture consists of the following key tenets: the principles guiding the architecture (including overarching principles of the education domain of discourse and architecture viewpoint principles for each viewpoint of concern); the architecture meta-framework used when performing systematic viewpoint analysis; and the architecture process model and the supporting architecture methodology that is followed (Steenkamp, 2007). The approach is supported by several architecting tools.

Overarching principles pertinent to this research are:

- Interoperability
- Scalability

- Adoptability
- Ease of Use
- Affordability
- Use software as service for delivery model
- Service-oriented architecture
- Use of open source technology
- Use of object oriented methodology
- Use of model view controller development framework

Using an architecture meta-framework ensures comprehensive coverage of the problem space. The architecture process model and supporting methodology simplify architecture development and ensure that the redesign is accelerated by reusing available patterns, templates and standards (The Open Group 2003). The TOGAF 9 Reference Model was adopted as the architectural guideline, and provided a sound foundation for this research (The Open Group 2009). The intent is to ensure that the redesigned architecture allows for future growth in response to the changing needs of the K-12 Educational environment.

The architecture approach followed in this project is intended to help resolve the stated requirements and address the problems, by providing a framework for building a prototype that shows how the individual systems should be integrated into one integrated system. By focusing the Student Information System (SIS) as the main source of data, the process of building an integrated enterprise system will be simplified. Redesigning the architecture of the proposed school system will enable the building of the enterprise system.

The proposed system will integrate the overlapping functionality as shown in appendix 3, in addition to adding the other needed functionality mentioned in Sections 1 and 2. In considering the design of the solution to the research problem the authors adopted the "DRY" principle: "Don't Repeat Yourself," first introduced by Andy Hunt and Dave Thomas which was used in a Ruby programming language (Thomas & Hunt, 2000). Applying the DRY concept at the system level resulted in the proposed system illustrated in Appendix 4.

The rationale of integrating the Student Information System with the Learning Management System, the Content Management System and the File Management System is based on the notion that the Student Information System is the core system in the school, containing information about all stakeholders. This system covers the management aspect of the school, including enrollment, attendance, scheduling, and grading. To add to these functions, the Learning Management System helps the teacher to add resources toward student learning. The Content Management System plays a major role in helping schools to integrate their websites with their teachers' pages and the course websites. The system utilizes Web 2.0 blogs as a way of communicating and adding content to the teacher resources. With the process content creation, the need for file support motivated the creation of a File Management System. There is overlapping of functionality between all these systems. Some examples of this are grading, student schedule, and course selection. Another example of overlapping is the end user roles among the four systems. If we look closely at the administrator, who manages user accounts between all of the systems, it is clear that the teacher appears in both the Learning Management System and Student Information System.

5. DEMONSTRATION OF CONCEPT – THE PROTOTYPE

In developing a prototype of the proposed system the semantics of UML 2.0 diagrams were considered. UML 2.0 provides a comprehensive set of representation schemes for modeling software systems. These include:

Use case modeling: The use case context diagram in Figure 2 shows how the proposed system can be used with the other entities (actors, end users, system) in the abstract. First level use cases describe the functionality of the system which results in value added to the end users (stakeholders). Use cases enable end users to get a first impression of the intended functionality. The use case context diagram in Figure 2 illustrates an example of a teacher's interaction with the system.

Process diagram: A process model serves as the foundation for redesigning a process, or redesigning the organizational structure

to better support the process desired. In Appendix 5, the process of a teacher posting a blog is shown. Also appendix 6 shows the process of posting an assignment involving several factors, including teachers, parents, students, and the proposed system.

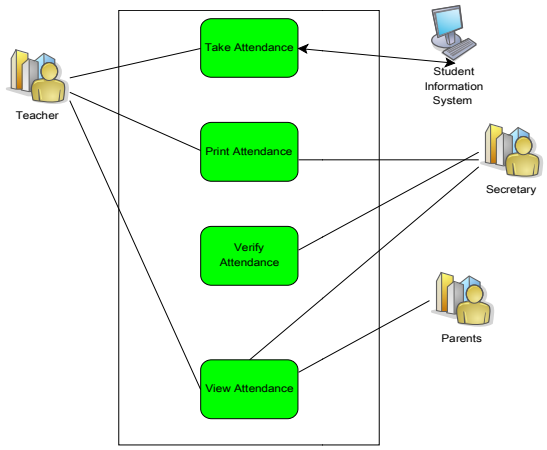


Figure 2. Teacher Taking Attendance Use Case

Meta-model of the Proposed System: In Appendix 7, a meta-model demonstrates the main classes and the relationships between them, in the form of a UML class diagram.

This section describes the phases of implementing a prototype of the proposed solution at a high level. The principle of agile software development was adopted and the solution was implemented using Extreme Programming (XP) as a software engineering methodology. This was done as part of action research, where the researcher was immersed and actively involved in the project. The process of iteration involved: process of planning based on requirements, obtaining and validating feedback from the various stakeholders, communicating the new requirements to the development team, testing, deploying of the new functionality, soliciting and receiving feedback from the end users, and communicating with the development team throughout the development cycle, as illustrated in Figure 3.

In the interest of simplicity, the feedback loops and iteration has been omitted from the diagram.

The following technologies will be adopted to develop the solution.

1. MVC Framework
2. Ruby on Rails
3. Apache Web Server
4. MySQL for Database
5. Amazon services for cloud computing, demonstrating the infrastructure as service (IaaS) platform for developing and hosting the proposed system.

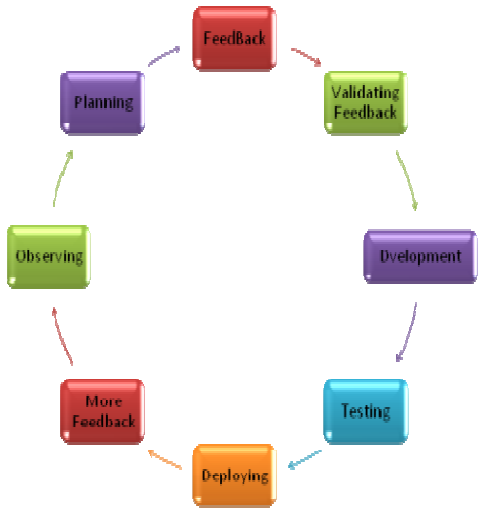


Figure 3. Proposed System Development Cycle

Web Application Architecture Framework. When looking to rebuild and interconnect the proposed system we will be using the Model View Controller (MVC), a web application architecture framework. This is the most important step and will come first in the remodeling of the new updated system.

Model-View-Controller (MVC) Overview.

The architecture of the MVC separates the application data model, user interface, and control logic into three distinct components. This separation is beneficial because if changes and modifications need to be made to one of the components, they can be done with little impact or harm to the other components.

Model: This is the domain-specific representation of the information on which the application operates. The Model is another name for the domain layer. Domain logic adds meaning to raw data, e.g. calculating the student's total absent days.

View: This renders the model into a form suitable for interaction, typically a user interface element. MVC is often seen in web applications, where the View is the HTML page and the code that gathers dynamic data for the page.

Controller: This responds to events, typically user actions, and invokes changes in the Model and perhaps the View. In addtion, the Controller is responsible for mediating between View and Model. It must translate View events, which originate from user input, to Model operation (Veit & Herrmann, 2003).

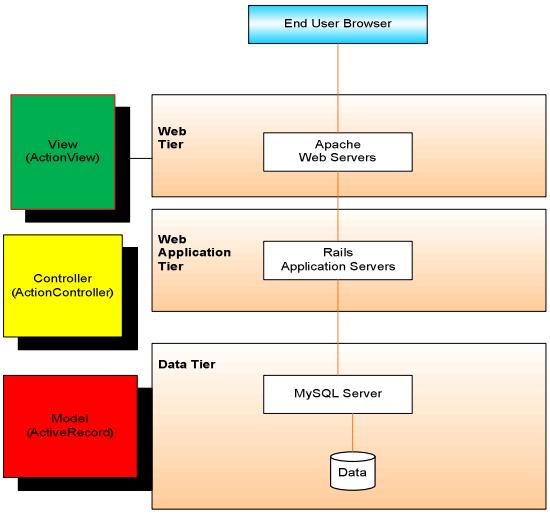


Figure 4. MVC Framework with Ruby on Rails

MVC is often thought of as a software design pattern. However, MVC encompasses more of the architecture of an application that is typical for a design pattern. Constructing an application using MVC architecture involves defining three classes of modules domain objects (such as student, class, attendance) that hold all the business logic and know how to persist themselves to a database. The Controller handles the incoming requests (such as Save New User, Update Student Record, Show Teacher Account) by manipulating the Model and directing data to the View. Figure 4 shows the conceptual model of MVC according to the user's viewpoint.

Ruby on Rails Development Framework

Once the MVC has been chosen for the architectural framework, the next step is to choose a development framework. This development framework should be compatible with the MVC architecture framework. After we examined the available development frameworks, it was clear that the best choice for the proposed system would be the newly developed open source product Ruby on Rails as the application framework. This web application framework builds web application frameworks according to the MVC. A rail is a full-stack framework for developing database-backed web applications built with the object-oriented language Ruby, according to the Model-View-Control architecture pattern (Chak, 2009).

One of the main principles of Ruby on Rails is "Don't Repeat Yourself (DRY)." In essence, this principle means that the information put into a system should not be duplicated.

Deployment

The deployment of the proposed solution implementation will use cloud computing, including Platform-as-a-Service (PaaS) for the development environment, Infrastructure-as-a-Service (IaaS) for hosting the proposed system , and Software-as-a-Service (SaaS) as the delivery model. Figure 5 shows the conceptual model, which demonstrates the end user viewpoint and the developer viewpoint in relation to the proposed system.

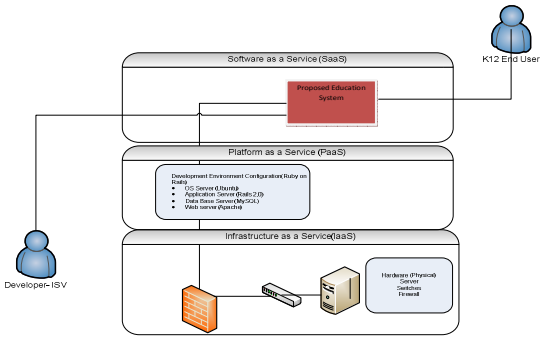


Figure 5. Deployment of Cloud Computing - Conceptual Model

6. SUMMARY AND CONCLUSIONS

The intent of this study was to examine the Student Information Systems in several charter schools in southeast Michigan and northeast Ohio, and to create an IT solution that would solve many of the problems endemic in the current systems. Some of the main issues with the current systems were identified, and suggestions and changes for a better and more efficient system emerged. The main outcome is an integrated system which eliminates the need for multiple disparate systems that were causing difficulties and inefficiencies for the school systems involved. The redesigned system allows all stakeholders to log in and access the Student Information System, the File Management System, the Learning Management System, and the Content Management System. New technologies available for web application development were identified and adopted into the redesigned solution. Using open source Web 2.0 technologies to integrate four disparate systems into one system allows users access to all of the school's services and systems with a single sign-on step. The proposed system described in this paper will be further developed by taking the next steps and implementing the prototype into a live system in the schools that participated in this study.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support and involvement of Global Educational Excellence, the Service Provider with the K-12 schools' management, and the input of all stakeholders during this research project.

8. REFERENCES

Aulbach, S., Grust, T., Jacobs, D., Kemper, A., and Rittinger, J. (2008). Multi-tenant databases for software as a service: schema-mapping techniques. *In Proceedings of the 2008 ACM*

Basal,A .(2009). Service Oriented Approach in Information Technology Enabled Education Systems

Basal, A. and Steenkamp, A.L. (2009). An Integration Approach to IT Enabled K12 School System, submitted to *ISECON2009*, Washington DC, United States.

Chak, D. (2009). Enterprise Rails. Sebastopol : O'Reilly Media Inc.

Curtis, G. and Cobham, D. (2002). Business Information Systems: Analysis, Design and

Practice, 4th edn., Pearson, Essex.

David Thomas, A. H. (2000). Programming Ruby: the pragmatic programmer's guide. Boston: Addison-Wesley Longman Publishing Co.

Dougiamas, M., & Peter, T. (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. *Journal of Educational Multimedia and Hypermedia* , 171-178.

Ferretti, S., Mirri, S., Muratori, L. A., Roccetti, M., and Salomoni, P. 2008. E-learning 2.0: you are We-LCoME!. In Proceedings of the 2008 international Cross-Disciplinary Conference on

Web Accessibility (W4a) (Beijing, China, April 21 - 22, 2008). W4A '08, vol. 317. ACM, New York, NY, 116-125. DOI= http://doi.acm.org/10.1145/1368044.1368070

Fruth, L., Larry, C. a., Michael, S., & Elizabeth, L. (2007). The Right Data to the Right People at the Right Time:How Interoperability Helps America's Students Succeed. Council of Chief State School Officers and Data.

ISO/IEC15288, 2002, Systems Engineering – System Life Cycle Processes. 2002-11-01.

Jacobson, I., G. Booch and J. Rumbaugh, (1999). The Unified Software Development Process. Addison-Wesley.

Gerard, B., & Marinos, A. (2009). Digital Ecosystems in the Clouds: Towards Community Cloud Computing. in Digital Gianpaolo, Carraro; Fred, Chong. (2006, October). Software as a Service (SaaS): An Enterprise Perspective. Retrieved July 17, 2009, from MSDN Architecture Center: http://msdn.microsoft.com/en-us/library/aa905332.aspx

Ecosystems and Technologies Conference. IEEE Press, 2009.

Gu, Q., & Lago, P. (2008). SOA process decisions: new challenges in architectural

knowledge modeling. *ACM Special Interest Group on Software Engineering* , 3-10.

Hew, K. F., & Brush, T. (2007). Integrating technology into K–12 teaching and learning: Current knowledge gaps and recommendations for future research. *Educational Technology Research and Development* , 223–252.

Harmon, P. (2003). A Manager's Guide to Improving, Redesigning and Automating Processes. Morgan Kaufmann.

Harris, A.L. and Rea, A. (2009). Web 2.0 and Virtual World Technologies: a Growing Impact on IS Education, *Journal of Information System Education, Vol. 20(2).*

M. A. Roth, D. C. (2002). Information integration: A new generation of information technology. *IBM System Journal , 563.*

Mahmoud, Q. H. (2005, April). Web Services. Retrieved 11 1, 2008, from sun microsystem: http://java.sun.com/developer/technicalArticles/WebServices/soa/

Mark, G., & Cindy, G. (2006). Integrating Technology For Meaningful Learning. 2006.

Michigan Merit Curriculum - Online Learning. (2006). Retrieved from http://www.michigan.gov/documents/mde/OE_Companion_Doc_12-06_184084_7.pdf

Perks, C. and Beveridge, T. (2003) Guide to Enterprise IT Architecture, Springer, New York.

Powell, A. (2008). K12 Online Learning: A Global Perspective. In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008 (pp. 2353-2380). Chesapeake, VA: AACE.

Remenyi, D., Williams, B., Money, A. and Swartz, E., 1998, *Doing Research in Business and Management*, Sage, pp.73.

Rienzo, T. And Han, B. (2009), Microsoft or Google Web.0 Tools for Course Management. *Journal of Information System Education, Vol. 20(2).*

Shen, Jia and Eder, L.B. (2009), Intentions to Use Virtual Worlds for Education. *Journal of Information System Education, Vol. 20(2).*

Steenkamp, A.L. (2007), An Approach to Developing IT Architectures, The Open Group Enterprise Architecture Practitioner' s Conference, San Diego.

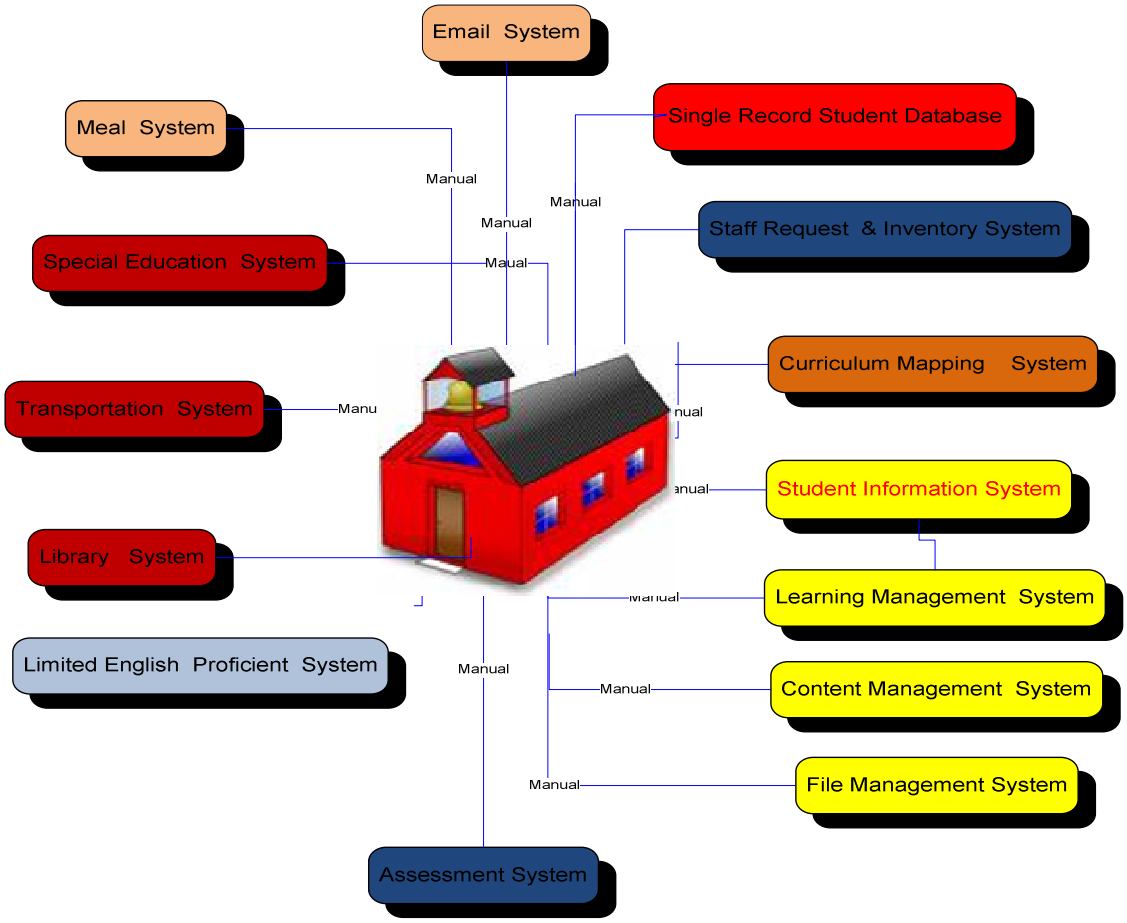
Steenkamp, A.L. and McCord, S.A., 2007, "Teaching Research Methodology for Information Technology", Journal of Information Systems Education, Vol. 18, No.2.

Schooldude. (2008). Retrieved July 15, 2009, from Schooldude: http://www.schooldude.com/elements/media/article_file/The_Unique_Challenges_Facing_the_IT_Professional_in_K12_Education.pdf

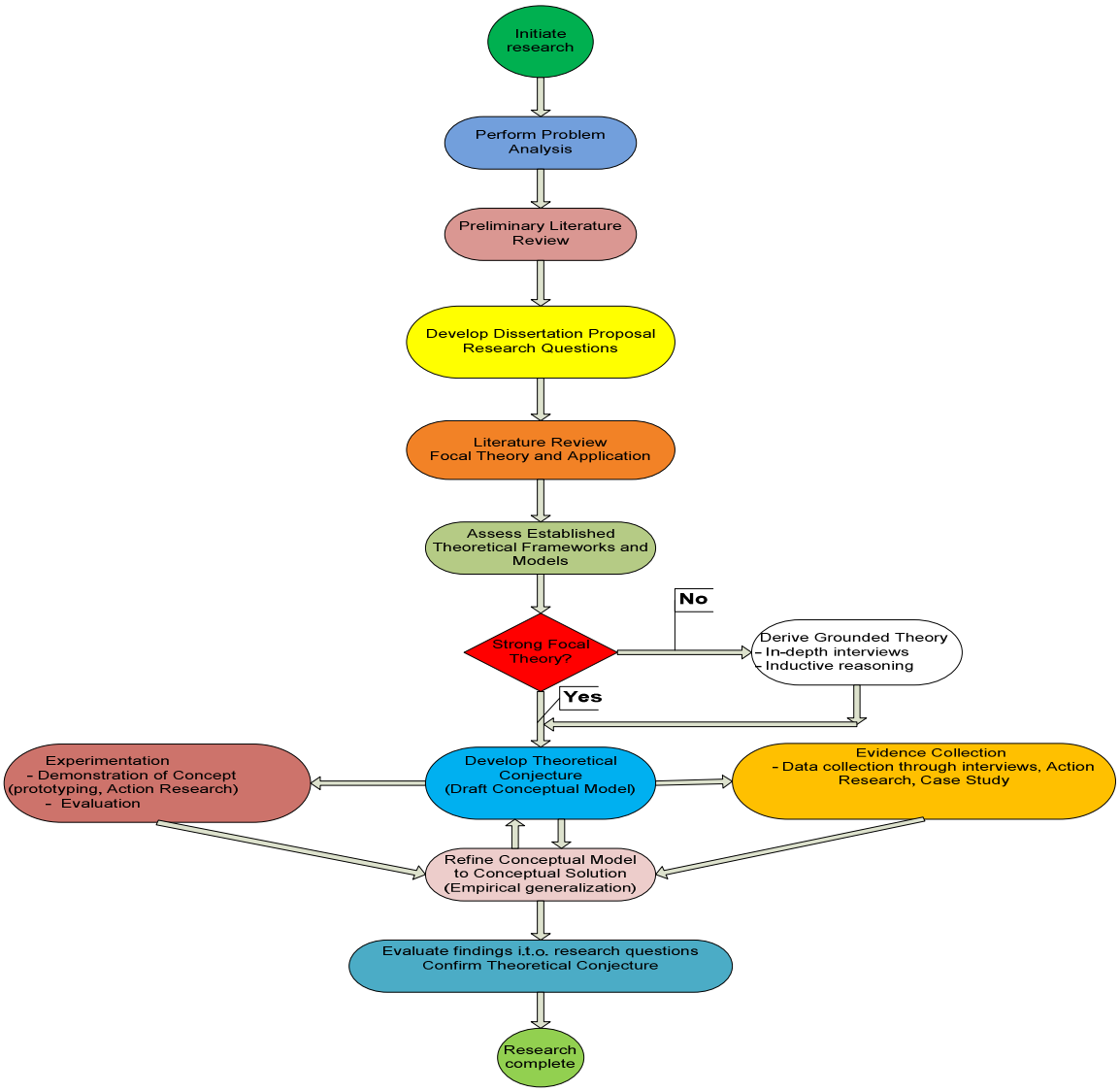
The Open Group (2009). TOGAF Enterprise Edition 9. Retrieved June 29th, 2009 http://www.opengroup.org.

Weil, P. and J.W. Ross, 2004, IT Governance, How Top Performers Manage IT Decision-rights for Superior Results, Harvard Business School Press.

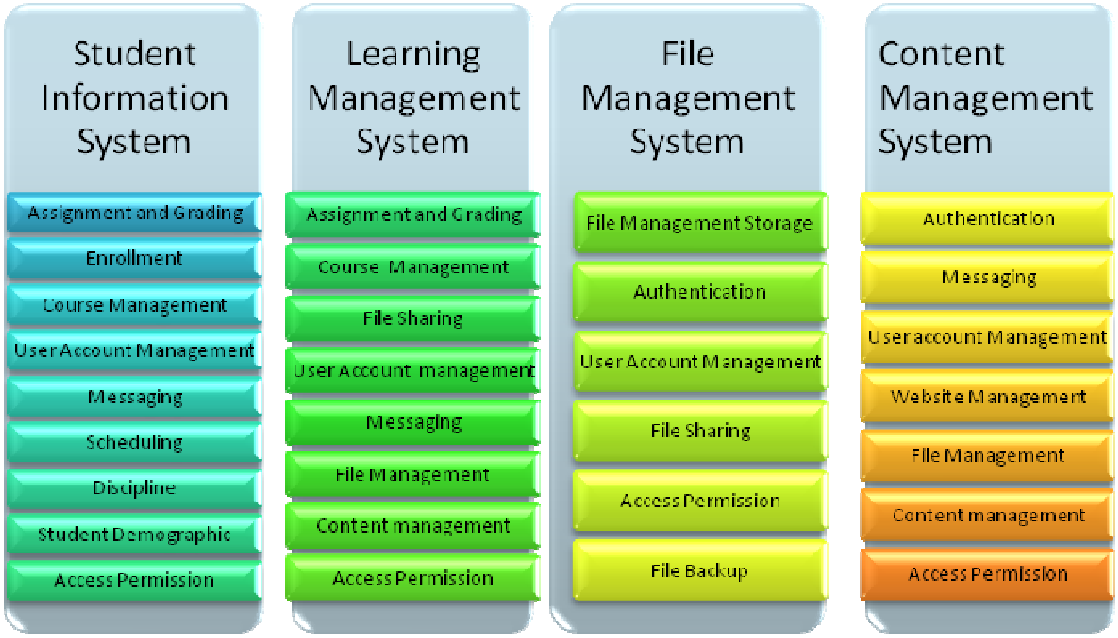
Appendix 1. Typical Applications in the K-12 School Environment



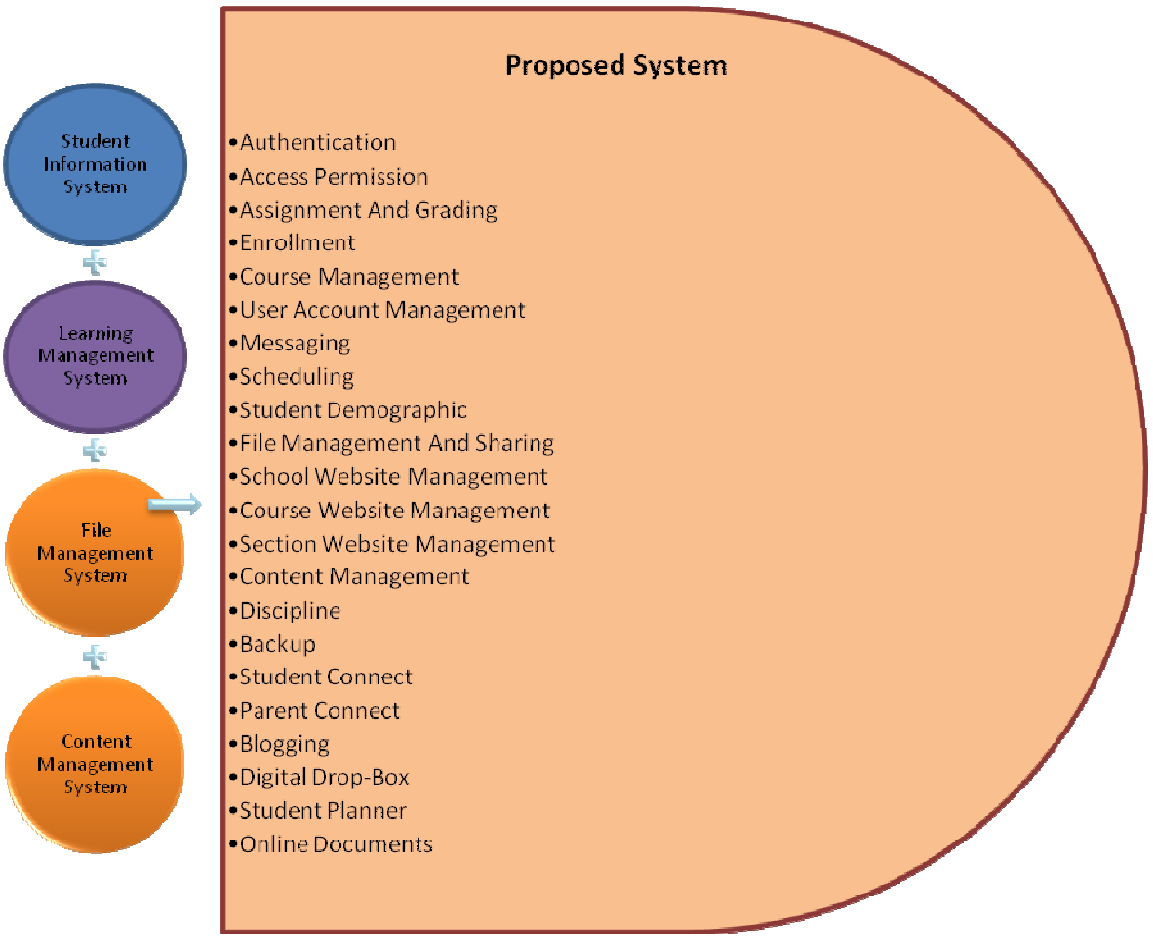
Appendix 2. Positivist-Empirical life cycle



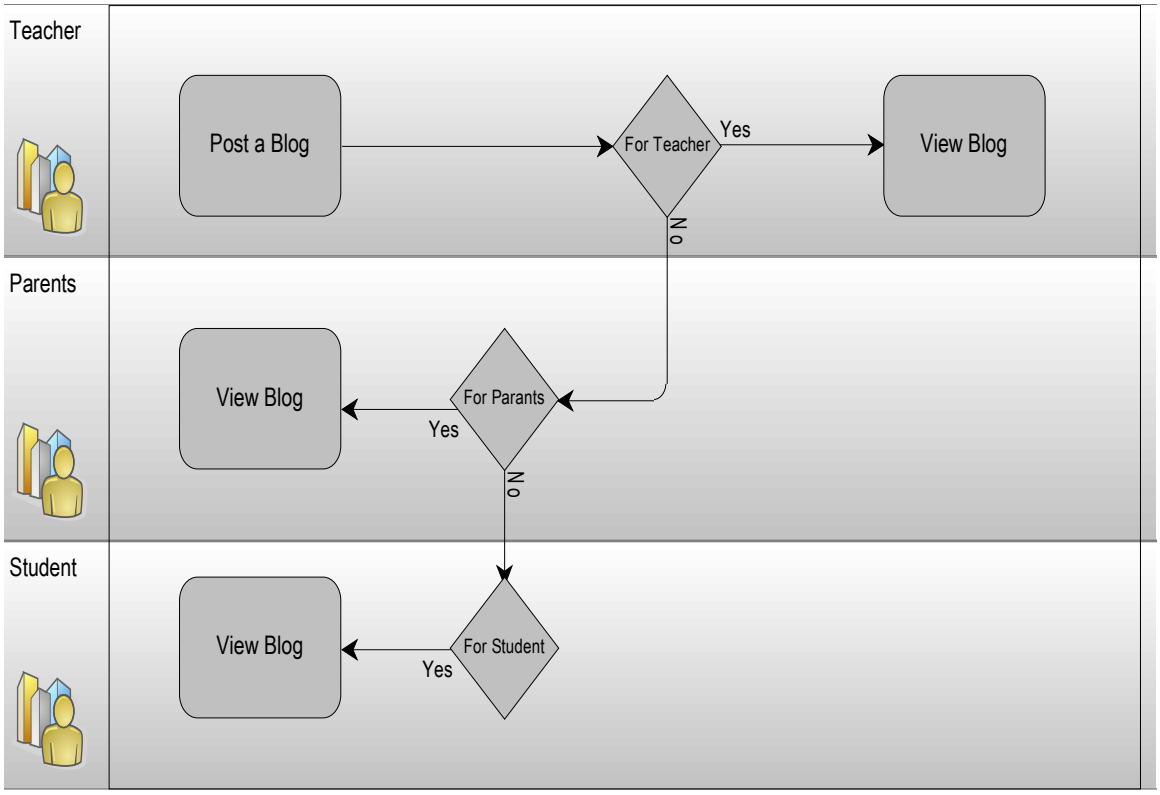
Appendix 3. Typical SIS, LMS, CMS, and FMS functionality in K12 School



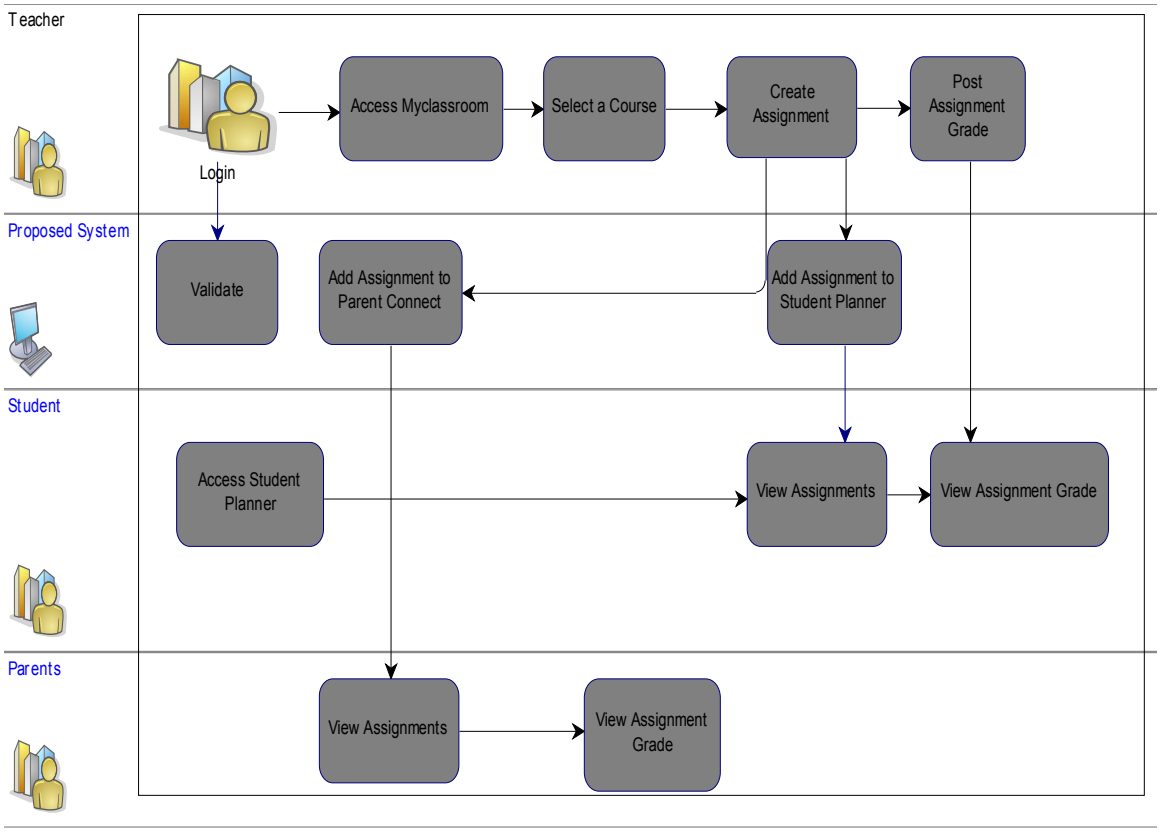
Appendix 4. Proposed Conceptual System with Integrated Functionality



Appendix 5. Blog Posting Workflow



Appendix 6. Process Diagram of Posting Assignment



Appendix 7. Meta Model of the propoesd System.