

ATHABASCA UNIVERSITY

Teacher Decision Making in a Ubiquitous Learning Environment

BY

ALEX MOTTUS

A Thesis project submitted in partial fulfillment

Of the requirements for the degree of

MASTER OF SCIENCE in INFORMATION SYSTEMS

Athabasca, Alberta

August, 2013

Creative Commons, 2013

Dedication

I dedicate this project to my wife and children who have patiently allowed me to disappear for large periods of time to work my studies. My daughters who remind me constantly of the wonders that learning with technology can represent. My wife who brings creativity and life to the realm of digital interfaces showing me that sometimes you have to look beyond simple logic to create something that engages the mind while it learns. Also to my parents who showed me the value of generalists approach in reading, science and art which forms the basis for an open mind, the first and most valuable tool in education. Finally to my father who passed away before I had a chance to complete this paper. I wish we had had the chance of one last debate...

Abstract

The ability for teachers to support students in ubiquitous learning environments has proven to be very challenging due to the limitations caused by physical distance and asynchronous communication methods. This has been further complicated by the dynamic nature of the learning process and a dramatic rise in the number of digital resources combined with those found in the real world. This has generated a vast amount of information in terms of learning interactions, mobile anywhere, anytime access, peer associations, and demographic materials. This plethora of information has made it extremely difficult to isolate critical facts and refine meaningful patterns. While data mining, adaptive learning systems and other education support techniques have provided a number of tools they have yet to create a universal method of assisting students. These mechanisms have also focused on replicating the expertise provided by teachers rather than supporting them. These techniques also tend to be highly complex requiring specialized skills just to use them thereby limiting accessibility. Given the challenges of automating educational processes this paper proposes a mechanism for reconnecting teachers with their students.

To accomplish this, an educational decision support system has been proposed in the form of an interactive dashboard to empower teachers with the ability to access student learning process information along with machine generated recommendations. The design has also focused on ways of expressing data mining outcomes using methods which have highly visual components in order to promote ease of access. This research has developed one potential teacher decision support system design for use in a ubiquitous learning environment. This has been augmented by also identifying the tools suitable for constructing such a dashboard. The prototype from this research has provided a proof of concept which can be used to guide future work and derive feedback from teachers and researchers alike.

Acknowledgements

Thank you to Dr. Kinshuk, my supervisor on this project for what has been a great journey into the world of research which I have come to have a whole new appreciation for. I would also like to thank Dr. Graf and Dr. Chen for acting as the review panel and providing some great feedback on the project. In addition I found the commentary and discussion with the NSERC/iCORE research group invaluable, as they welcomed me into their academic community.

Contents

Table of Figures.....	v
Chapter I.....	1
Introduction.....	1
Purpose.....	1
Research Problem.....	1
Organization of Chapters.....	3
Chapter II.....	5
Literature Review.....	5
Supporting Ubiquitous learning.....	5
Learning Analytics.....	8
Visualizations.....	12
Summary of Literature Review.....	14
Chapter III.....	15
Methodology.....	15
Assumptions.....	15
Scenarios.....	16
Students in Need.....	16
The Class Perspective.....	17
Examining Groups.....	18
Student Perspective.....	18
Teacher Decision Making.....	18
Requirements.....	19
Architecture.....	20
Information Requirements.....	22
Specifications.....	22
Chapter IV.....	25
Beginning Development.....	25
Underlying Framework.....	25

Review of Middleware Database.....	26
Working with the Data	26
Implementing Visualizations.....	27
Selecting Visualization Types	28
Dashboard Implementation	29
Dashboard Pages – Highlights that Need Attention.....	29
Dashboard Pages – Student Specific Elements	32
Dashboard Pages –Class Summary	37
Dashboard Pages - Groups	39
Prototype Function	42
Validation	42
Chapter V.....	45
Discussion.....	45
Support Systems and Experts	46
Remaining Gaps	47
Future Vision.....	49
Chapter VI.....	51
Conclusion.....	51
Future work	51
Works Cited.....	53
Index I – PALD	59

Table of Figures

Figure 1- High Level Architecture Diagram.....	21
Figure 2 – Example of Organizing “At Risk” students	30
Figure 3 – Nodal map of Students and Tutors.....	31
Figure 4 – Summary of Class Summaries and Resource Needs	32
Figure 5 – Student Demographic and Learning Object Preferences	33
Figure 6 – Sample of Student Location.....	34
Figure 7 - Example of Domain Knowledge Visualization	35
Figure 8 - Student Learning Style	36
Figure 9 – Sample Recommendation Object.....	37

Figure 10 - Sample Class Visualizations.....	38
Figure 11 - Pie Chart and Scattergram of Student Progress.....	39
Figure 12 - Groups Selection, summary and bubble chart	40
Figure 13 - Heat Map and Recommendations/Intervention Actions.....	41
Figure 14- Student Marks in Line and Table Formats	41

Chapter I

Introduction

Purpose

Ubiquitous learning has become commonplace with computers and networks growing to provide access to vast arrays of educational resources. This has been further enhanced with the addition of mobile technologies eliminating barriers of location and time. From Learning Object Repositories (LORs) to generic basic web pages, opportunities for learning are almost as great as the Internet itself. Modern learning environments have grown to encompass physical resources that can be utilized in conjunction with those found in digital spaces. As students and teachers moved into digital and real world environments, the old classroom model has become less of the norm but many of the supports have not moved with them. While learning resources are plentiful, the guidance and structure to make effective use of those resources can be porous at best. Thus, the goal of this research is to examine a way to allow teachers to observe and support their students more effectively in a distributed learning environment. A situation in which students may be working at home, as a group in a coffee shop, or in a mobile scenario where they can be engaged in learning using smart phones. This degree of mobility also encompasses interaction with physical materials found in museums, libraries, or in nature itself. The critical element of all these examples is that the teacher is not physically present even though learning objectives are being addressed.

Research Problem

In terms of modern education there has been a shift away from older models of educational delivery defined by having a teacher and student(s) in the same room. There are many reasons this shift has occurred; flexibility of schedule, inability to physically attend, or even a preference for individual learning opportunities. The most common scenario has been a student working at home using a web browser interface to access digital resources found in a Learning Management System (LMS) like Moodle. Upon closer review this component only represents one piece of the educational environment that a student encounters. There is no restriction to prevent a student from accessing other resources found freely on the Internet with a simple search. This type of exploration can lead to a variety of sources offering both high and low quality options with very little guidance as to their effectiveness. Course content can provide suggestions but students, especially those seeking extra support, are likely to move outside of the regular bounds. Increasingly students are also using mobile devices which add another layer of communication and resources. Smartphones and similar equipment make Internet access time and place irrelevant allowing individuals access to multimedia objects and other information sources. These can be linked to physical elements by a variety of means

including QR codes, RFID and GPS location to provide a hybrid approach to leveraging digital and real world information simultaneously (Nelaturu, Kambham, Karna, Parupalli, & Mandula, 2010). It is this universal access that immerses a student in a truly ubiquitous environment that spans learning objects in all environments, regardless of where they reside. This has created an environment where learning options are plentiful but with a notable lack of contact and subsequent support from teachers.

Many of the traditional teaching methods rely heavily on physical proximity where teachers can observe and evaluate their students directly. In an online or distributed scenario this is no longer possible. This problem is further compounded by the explosion of educational options that students are presented with in terms of resources and learning choices. This means that not only is it difficult to physically observe students but it is also virtually impossible to predict their behaviour in terms of what learning resources they will select. One irony of this situation is that there is no shortage of raw information about students. Individual learners generate hundreds, if not thousands, of interactions as they click, browse, and otherwise interact with physical and digital artifacts. Likewise Student Information Systems (SIS) and LMS also regularly collect demographic, assessment, and activity data related to each student. Unfortunately, getting access to this data and manipulating it into a useable form is not an easy task. The term “big data” is often used to refer to large repositories of information which while plentiful, can be unstructured and so large that they make processing extremely problematic. There is also the challenge of aggregating disparate data structures that can come from a wide variety of sources. To address the problem of generating useable information, educational data mining (EDM) has emerged as a unique research area. It has taken pattern recognition mechanisms from a variety of fields and focused them on learning processes. This in itself is a complex task requiring considerable expertise in data management and algorithms to begin creating results that are useful. Even the outcomes can be highly complex and require a great deal of training to understand and leverage in an effective way.

Further compounding the problem is the time sensitive nature of supporting learning. As students move through courses, assessments, and other constructs they can encounter a range of difficulties which would benefit from the assistance of a teacher. While some students may self-identify by asking for help, this would seem less likely to occur in an environment where access to a teacher or other educational assistance can be difficult to achieve. Basic problems occur if the student does not know how to contact a teacher or if the teacher is currently unavailable which can cause significant pitfalls. If a teacher is to proactively assist a student they need access to learning information that is both real time and historical as they require context along with the awareness that a problem is occurring.

Work in and related to the field of educational data mining has developed a variety of mechanisms for identifying patterns found in student learning. These range from

developing learning styles to finding students at risk so that researchers and teachers can make decisions and drive future interactions. Decision support systems found in business provide context as well as identifying specific issues. In the same way a system designed to support educational decision making should provide an overview of the situation. Simple outcomes such as flagging students needing help are relatively intuitive but the supporting information may not be. Since it is the outcome that provides educational value, it makes sense to abstract the teacher from the data mining actions and simply present the result. Any supporting information also needs to be displayed in the same way, in order to make it easy to understand.

The scenario described above suggests the need for a tool that provides teachers with a variety of information about what their students have done as well as current situations. It should be noted that teachers are experts in the process of learning but not in computer science disciplines such as data mining and software creation. Thus they require a tool which can abstract teachers from the complexities of pattern generation and render them in a way that is intuitive and can be manipulated with relative ease. Any such system should be able to go one step further and provide suggestions for action in order to make teachers aware of options while giving them the flexibility to choose.

To address the problems presented, a prototype of a dashboard is proposed to illustrate one potential mechanism for improving the connection between teachers and students in a ubiquitous learning environment. A dashboard has been selected as it is a highly visual tool for representing information which can be used as a reference point for current and future discussions on student support frameworks. By generating a preliminary design based on simple student support scenarios it is possible to create a “proof of concept” for a decision support system which can be validated by implementing the requirements identified.

Organization of Chapters

The rest of this paper follows through the initial review and analysis to implementation, discussion, and final conclusions. In particular, a review of related literature and the contributions of various fields are presented in Chapter 2 to capture an overview of the current state of the art. The review is followed by an examination of the proposed solution and the analysis of the requirements needed for implementation. This includes a summary of the requirements as well as a set of scenarios which are all contained in Chapter 3. The project implementation phase and the evolution of the design are subsequently addressed to demonstrate the outcomes of the project in Chapter 4. The focus of the paper shifts to a discussion of the project and the insights gained from its creation which is covered in Chapter 5. This includes an overview of remaining questions and the direction this researcher sees the field evolving. Finally, there is a summary of the paper which draws

together the conclusions and identifies areas for future work in Chapter 6 drawing the paper to a close.

Chapter II

Literature Review

Within the context of ubiquitous learning environments the challenge of retaining individual interactions that form the underpinning of most human learning has been an area of focus for several researchers. These include peer to peer associations along with the student-teacher links that support and guide the learning process. In response certain streams of research, such as Intelligent Tutoring Systems (ITS), have striven to enhance the abilities of learning systems to mimic the teaching process. There has also been a complementary vein which has worked to allow the teacher to function within the online learning process in a more effective manner. The later process can be seen in the field of Adaptive Learning Systems which seeks to leverage the strength of computer based learning while maintaining some of the benefits of teacher-student interactions. In both situations, student information must be extracted in a real time manner that can allow an instructor to monitor the process of both student groups and individuals.

Before addressing the future needs of ubiquitous computer mediated learning environments, a focused review needed to be carried out on the paths that have brought researchers to their current stages. An examination of the high level roots of digital learning was necessary to identify the foundations that modern components have been built upon. The evolution of LMS's, Intelligent Tutors, Learning Companions, and Adaptive Learning Systems (ALS) are critical to understanding the implications of designing an educational decision support system. These areas have also focused on various learner modelling methodologies and enterprise architectures that have been used to support increasingly large populations of online learners. Recent research has also examined the new reality of ubiquitous mobile learning that needs to be contextually driven by location and real world learning objects. This has further added to the complexity of an already challenging topic. Only after careful review of these concepts was it possible to look at the future of learning and possible tools for supporting student needs with both computer assisted methods and teacher driven interfaces aimed at restoring the learner-instructor relationships.

Supporting Ubiquitous learning

A number of different avenues have been pursued by researchers over the years looking for the best approach to delivering educational services. Many of these methodologies have similar aspects allowing different areas to support each other directly and indirectly. With this in mind, it made sense to take a close look at some of the primary approaches to see how they have developed and to look at their strengths and weaknesses. In this manner, the best features could be identified along with the challenges that needed to be addressed going forward. The areas that represented some of the most significant

contributions were; the LMS, ITS, Learning Companions, Recommender Systems, Simulations, and Adaptive Learning Systems (ALS).

Computer supported learning achieved a new level of sophistication with the introduction of the LMS. Within these environments, digital content could be served from a single location, broken into manageable chunks complete with assessments while providing unique user profiles to students and teachers. The possibilities created by these learning environments were revolutionary as they advanced content delivery into a new age digital education. Very quickly good results were demonstrated from the use of these types of systems (Mitrovic, 2006). Just as LMS's provided great leaps forward it was also evident that they had certain limitations. Early implementations were primarily web servers optimized to provide learning and teaching information. Essentially early online courses were digital books that could be consumed by remote users via browsers (Bargel, Schrock, & Roller, 2012). In many cases this is still the situation today with the addition of multimedia content, sometimes referred to as hypermedia, to provide both sound and video components (Brusilovsky, Sosnovsky, & Shcherbinina, 2005). What these online courses very clearly lacked, and in many cases still do, was the ability to learn from student activities and tailor the experience to the individual's needs. This was a somewhat surprising discovery as one of the functions that most LMS perform very well is to collect vast amounts of information about student interactions with digital content (Graf & Kinshuk, 2006). Generally this has been captured in the form of logging files which collect everything from clicks/taps and other responses to the amount of time students spend on pages or interacting with specific content. The assessment results from various activities have also been catalogued to form an incredibly rich source of raw data. As LMS's became increasingly common the question turned to what to do with these repositories of information and how to handle it effectively.

Of all the learning mechanisms, the ITS are perhaps the most recognizable from popular fiction sources. The ideal of a computer or robot that effectively replaces a human teacher is one of the great promises of computer science. This could be considered the educational equivalent of the Turing Test as ITS strive to create the same value as having a live instructor. Research in the ITS area has provided significant contributions in learner, domain, and expert system modelling (Mitrovic, Martin, & Suraweera, 2007). The reality of these systems is that they are incredibly complex, requiring a vast amount of information in order to capture the same degree of domain knowledge as an expert human counterpart. In addition, these systems also need to be able to evaluate student needs in order to facilitate proper communication or to provide learning resources that meet learner's changing requirements. To do this, ITS frequently make use of questionnaires and student modelling techniques based on learning interactions between the student and system (Latham, Crockett, McLean, Edmonds, & O'Shea, 2010). While these systems have proved effective at improving student results and engagement, they

still have a long way to go in order to fully replicate a human teacher (Mitrovic, 2006). Added to this issue is the level of effort that is required to program them. This has proven very time consuming even with experimental work on automated interviewing methods for capturing knowledge and skills from human domain experts (Hwang, Chen, Tsai, & Tsai, 2011). Other methodologies such as using constraints and fuzzy logic to make ITS systems more generic have simplified the programming but not enough to make them realistic on a large scale (Mitrovic, Martin, & Suraweera, 2007). These systems also tend to be very subject specific requiring large scale modifications to apply existing systems to new subject areas. Whether these limitations prove to be temporary, as computer system solutions grow in sophistication is yet to be seen. Work with intelligent agents and other architectures may well overcome these difficulties in time but not in the immediate future.

A similar line of research can be found in the area of learning companions. These systems focus on providing learners with a partner for exploring and gaining knowledge about new topics. This has shown to reduce learner confusion in virtual environments resulting in better outcomes. These systems also seek to address the social aspect of learning as they provide individual students with a source of communication that they use to develop new ideas (Wu, Miao, & Shen, 2012). Learning companions can be set at different levels of expertise depending on the skill level of the student. They can be used to evaluate student progress by providing false information to see if the learner can detect the mistakes (Chou, Chan, & Lin, 2002). Research on these systems has expanded the understanding of student learning as well as the mechanisms involved in providing useful feedback and evaluation. Learning companions do not provide as much direct intervention as they are designed to be more subtle in their approach. While they are certainly useful, there are limitations to how much impact they can have on the learning process. In some cases they are used as a supplementary component for modelling and ITS approaches (Limoanco & Sison, 2002). Results from automated systems can be compared to those identified by expert users but that is also usually done in some sort of mechanized fashion (Brтка, Radosav, & Brтка, 2010). This refers to the fact that there are few validation mechanisms in place to allow a human expert to determine if a system generated adaptation is correct. Learning companions generally lack a validation mechanism, meaning, if they make an invalid assessment they may not be able to autocorrect themselves (Chou, Chan, & Lin, 2002).

Another type of automated learning support is represented by the use of recommender systems. The recommender approach has its roots in the management of content and, while they may develop knowledge about the learner, their primary purpose is to decide what information to present and in what format. The latter explores some interesting concepts as the very nature of computer supported learning is aimed at providing information in visual and auditory formats as well as text. These systems can also take into account the type of device used by the student (Ahmed, Chang, & Kinshuk, 2012).

For example, mobile users require a low bandwidth alternative for large, bandwidth intensive artifacts which would not work well on their devices (Verbert, et al., 2007). Recommender systems are generally made up of two primary components; a repository of content that is tagged in some sort of meaningful way and a mechanism for predicting the preferences of the learner. The preference engine is generally developed in one of two ways; either by examining past behaviour to determine likes and dislikes or by modelling the learner to try and group them with other individuals whose preferences are already known. Recommender systems have been highly successful in e-commerce contexts and as a result have expanded into the mobile world extremely rapidly. As a result, recommender systems are of particular interest as they are already bridging the gap between content and location. This relates directly to the idea of being context aware of location which is rapidly gaining traction in the education world. Museum environments have been used in the type of research to demonstrate location and recommender support in an educational scenario (Wu, Chang, Chang, Liu, & Heh, 2008). It should be noted that that recommender systems are simply facilitators in the online learning situation. By definition, recommender systems simply provide advice, leaving the final decision to the student who needs to self-regulate their learning (Rodriguez-Cerezo, Gomez-Albarran, & Sierra, 2011). This means the student has the benefit of a supported learning environment while still retaining the ability for personal choice. These can include mechanisms that evaluate user preferences, identify content, as well as mediating communication methods.

Adaptive Learning Systems are somewhat unique in the context of this review as they tend to take a more interdisciplinary approach. Rather than focusing on specific delivery these systems take a more holistic view of the learning experience and the how the different elements interact to influence the student. This approach is also dynamic by its very nature as the assumption is that the student is constantly changing as they acquire and potentially forget information (Virvou & Chrysafiadi, 2006). This represents a number of challenges as it puts the onus of understanding and detection on the learning system and the teachers. There is also an emphasis on providing content in different streams to be able to accommodate different student needs and attributes. This lends itself to number of different architectural models although the underpinning is still that of a strong student modelling approach coupled with learning objects and mechanisms for changing approaches in real time. Adaptive learning systems generally put more emphasis on having an educator involved as they seek to provide information for use by a human agent. This suggests a hybrid approach which leverages the expert skills of a teacher augmented by a robust learning system jointly providing educational support and opportunities (Graf & Kinshuk, 2009).

Learning Analytics

Any review of modern educational support systems would be remiss without a section on data mining and its development in the realm of learning systems. The systematic analysis

of information is an endeavour which has been pursued for years by businesses striving to take vast repositories of customer and transactional data and convert it into aggregated elements that can be used to identify current trends and predict future outcomes (Uthayakumar & Sarukesi, 2011). While this type of work does not necessarily draw a direct line to education, it provides some interesting insights into the problems and solutions that arise when dealing with massive data sets. It has also promoted the idea that decision support systems are an important part of being able to support student learning. These systems draw on a variety of data mining outcomes in order to provide sophisticated information analysis. Research has shown that these types of systems have been highly effective at evaluating student progress and determining what content best suits a learner's needs (Hatzilygeroudis, Giannoulis, & Koutsojannis, 2005). Another research area that has made considerable use of large scale data analysis is that of scientific inquiry. An excellent example is the field of genetics where some of the techniques used in identifying patterns in genes could be applied to education data analysis (Ninomiya, Nakayama, Shimizu, Anma, & Okamoto, 2007).

Through the discussion of LMS's and approaches to learning, it has become evident just how much information has been collected during the learning processes. Different computer logs and other data capture mechanisms run constantly as individual students navigate through various learning environments. The process of converting all that raw data into meaningful chunks has proved to be a much more complex process that has been the subject of considerable investigation in the past few years. The fields of Learning Analytics and Educational Data Mining (EDM) have both become common terms for identifying patterns in student information and related log files. These lines of research have formed useful supports for the development of expert systems which can assist teachers when they are well-designed (Hwang, Chen, Tsai, & Tsai, 2011). A number of practical considerations have been focused on, which form the basis of pattern identification. Perhaps the most basic example has been to form associations between a variable such as an action or environmental element and a resulting outcome. This represents a simple but very useful mechanism for using data to identify predictive processes and support future actions on the part of a teacher or researcher. Sequential patterns have been another useful outcome that has been derived from large data sets (Romero, Ventura, Zafra, & de Bra, 2009). The ability to identify similar activity chains has made it much easier to predict future behaviour based on subsequent actions. This has been seen quite commonly in online navigation activities as certain "clicks" may result in a very distinct pattern of activity. Both of these types of analysis have been used to develop recommender systems which seek to provide options customized to an end user. These provide greater insight into system user activities which has a very strong parallel to the activities seen in some learner situations. The clustering or classification of common events is another important approach to data mining. This is where common items are dynamically identified by evaluating their various attributes or properties. While

highly flexible, this type of analysis has been shown to be subject to false positives as groupings may be identified which are not useful (Zhou & Xu, 2010). Classification, on the other hand, has normally been done by using experts to develop attributes clusters found in common groups which can then be used to categorize data elements. This too has had its challenges as distinct groups may be aggregated resulting in the loss of unique characteristics. One of the most powerful tools that can be generated by data mining has come in the form of association rules which can be used to tie different pieces of information together (Tang, Chen, Lin, & Cui, 2012). These have generally been generated dynamically but like clustering can be subject to a number of false positives. The complexity of association rule analysis can be very high as it can involve a number of different pieces of information and sequences that work together to form a meaningful pattern.

While the emergence of data mining has its roots in business and scientific analysis it has become quite clear that educational information has added a number of specialized nuances. In education, the usefulness of simple queries has proved fairly limited (Chanchary, Haque, & Khalid, 2008). While these may point out basic correlations, most learning activities generally contain of a number of steps that can contain interactions with content, peers, and instructors. Complex learning behaviours could also involve reflective activities completely internal to the student themselves. By contrast most business activities have been measured by simple “transactions” or events that have been defined quite clearly by a purchase, click, or other activity. Within the context of learning, the acquisition of a skill or concept could take place over a prolonged period which might take several attempts. Invalid knowledge could also be acquired which would require “unlearning” of a concept which must be replaced by the correct one. Also supporting the idea of the transient nature of learning has been the process by which a piece of information can be forgotten resulting in the need to relearn it. Being able to recognize and act on these changes has presented an incredibly challenging aspect that is not seen in any other field. As learning processes evolve raw data can become a living entity that can change and evolve incrementally and by quantum shifts which occur as a student achieves more advanced degrees of understanding.

When looking at information analysis in education, another important consideration has been the time frame in which different researchers have selected their topics. Many of the early investigations focused on improvement of online courses. In particular, much of the research looked at links between learning objects and the assessment results that were generated by their use. This was critical for improving courses and forming links between pedagogical ideas and the outcomes that were created (García, Romero, Ventura, & de Castro, 2011). Unfortunately this has most often been done after courses have been completed which means that it only helps future students. As the EDM and Learning Analytics (LA) fields have matured, other avenues of research have begun to look at ways

to provide better educational support and to identify students at risk. The goal has been to make LMS's proactive and provide educators with better tools for assisting students. This trend has indicated that researchers have been moving to more complex pattern identification in order to refine generic data mining practices to a more educational perspective.

In the past few years EDM and LA techniques have been used to predict student behaviour, to group common activities or actions, and to identify relationships between different variables. From that basic set of objectives, a vast range of strategies and implementations have emerged. From a pattern analysis point of view, there are a number of different techniques that have been used. Some of the most common methods used have been clustering, decision tree construction, rule induction, neural networks, and logic programming (Brtka, Radosav, & Brtka, 2010). An example of one use of these methods has been the attempt to group learner behaviour into some sort of predictable classification. Those activities could represent a set of specific actions, a preference for certain types of learning objects, or a tendency to spend more time on a learning action. The initial identification of these patterns has been handled in two primary ways. The first method was with the use of experts who can either submit patterns to search for or to identify patterns that the system can then built rules for and attempt to match. This approach worked well with constraint based systems which can be programmed with extensive rules sets (Mitrovic, Martin, & Suraweera, 2007). It has also provided a validation mechanism as human expert and machine driven results could be compared and contrasted for effectiveness (Brtka, Radosav, & Brtka, 2010). The second method use has been the induction based method. It too could use experts by interviewing then and developing rules. Or it could simply scan and attempt to identify interesting patterns. One of the challenges with this approach has been that not all patterns are useful. Quite often unique sequences appear that have no real meaning. Those can be difficult to avoid or eliminate as they could consume a great deal of computing and user time (Zhou & Xu, 2010).

Another basic but effective student support mechanism has been to determine the time required to complete certain learning components. If a section is particularly hard to understand or the concepts are not properly explained, a much longer learning sequence could occur. Naturally the more complex the learning process the more difficult it becomes to analyze. As mentioned earlier, one of the challenges of educational analysis is that a learning "transaction" may not occur in a single session. Instead it might occur over repeated attempts by the student. This means that pattern creation might involve aggregating multiple learning sessions to capture activity involved in the acquisition of a specific concept. This type of analysis has frequently been seen supporting recommender systems which strive to provide suitable options for learning by analyzing user logs for current and past behavior patterns (Puntheeranurak & Tsuji, 2005).

Visualizations

From an end-user perspective, how data has been mined is not nearly as important as its presentation. It does not matter whether aggregated information has been selected for educators, businessmen, scientists, or any other expert group. The ability to formulate it in a manner that can easily be understood is critical. Work by data visualization researchers (Fry, 2008) has moved across many fields but with the same goal of providing information that is meaningful and timely. There has also been research completed to support the belief that there is general a need for the increased use of visualizations in data analysis (Ali, Hatala, Gašević, & Jovanovic, 2012). Given the complexities of data mining it has proven necessary to abstract the data consumer from that process or at least give them the option to bypass the messy details of chunking large information sets (Zorrilla & García-Saiz, 2012). While power-users might still wish to involve themselves in the finer points of refining information sets there are a larger number of experts who focus on information consumption in order to obtain specific results (Hu, Almansoori, Kannan, Azarm, & Wang, 2012). This can be considered particularly true in education where the course delivery and management take up the majority of a teacher's time. Even in highly developed online environments marking and other processes require a larger commitment in terms of working hours.

The primary uses of visualization tools and dashboards has been to support decision making allowing experts to quickly leverage large data sets in order to assist with day-to-day activities as well as long term planning (Lavrac, et al., 2007). It has also been helpful to look at what decisions dashboard users need to make. From an operational point of view, trends and status updates are both important aspects which have provided experts a real time awareness of where the information they are monitoring is going (Mahendrawathi, Pranantha, & Utomo, 2010). The identification of exceptions has also proved important as those elements which stray outside the norm can offer quick flags as to what is going wrong or identifying new patterns that may be emerging (Sachin & Vijay, 2012). One of the other advantages has been the ability to drill down from general to very specific information. The ability to look at a general data set and then select a specific element that is of interest is a very powerful tool in the hands of an expert user. In this type of scenario, it might also be necessary to have multiple educators working with the same cohort of students. This has led to the need for a collaborative approach which has also been facilitated by the use of visualizations (Tchoua, et al., 2010). Given the range of requirements a highly flexible approach has been needed to support the types of visualizations used dashboard implementations (Yigitbasioglu & Velcu, 2012).

Taking the observations listed above specifically to an educational context these tools could allow educators to monitor classes in a highly efficient manner (Kent, Carr, Husted, & Pop, 2011). Even multiple classes could be tracked which is an important consideration as student to educator ratios are typically much higher in online environments. An

example of a simple visualization would be to compare multiple assessments across different cohorts using graphing techniques (Friedler, Tan, Peer, & Shneiderman, 2008). This allowed instructors to evaluate how effectively different students find educational material and observe what kinds of interactions they had. It has also included information about where and how students access learning objects. This has become a much more important detail as hybrid courses which combine online and real world aspects have quickly become the majority of course offerings (Li, 2009). Another aspect that has been deemed extremely important has been the ability to identify **at risk students** or those who might require a teacher's intervention to assist them. These different types of needs generate two of the streams that have emerged quite quickly from the use of dashboard style tools. One has been the ability to observe how a class as a whole has been performing and to view global trends and activities within that group. The second part has been the ability to select specific students or cohorts to identify specific issues and problems. These different mechanisms must be capable of being analyzed separately or together in contrast to determine how best to meet learning needs. Underpinning this type of analysis has been the ability to develop student and pedagogical models which could also be presented visually to evaluate performance on both a student and class level.

In terms of how information should actually be presented, there have been a number of mechanisms identified. Graphs have been used for years to visually present information and can very easily be applied to a dashboard scenario (Martin & Di Bernardo, 2008). Pie, histograms, bar, and similar charts have effectively represented concepts such as assessment results, course progress, and use of different learning objects. These have also been used to present relational data such as time spent with different tools and success in achieving learning outcomes. Concept and neural maps have been expressed graphically to show linkages and the number of links between different objects, students or other relevant nodes. In terms of location, maps have been used in concert with GPS and RFID technology to provide a physical representation of where students were located. This has been related to student's later activities and courses progress to provide a more holistic view of their activity. Other styles of presentations have also been used such as the use of word maps to provide a concept view of what students are doing and which elements they have made the most use of (Gomez-Aguilar, Conde-Gonzalez, Theron, & Garcia-Penalvo, 2011). Scattergrams have been utilized to show not only clustering of information but to quickly illustrate those exception points which fall well outside normal results. The representation of student specific information has been handled in similar ways. For example, different student activities or traits could be mapped to different graph types. This could also be related to similar students and their results to predict future performance. Student pace and social interactions could also be good indicators or success and might lead the instructor to provide suggestions that can proactively assist the student in their studies. Sociograms and neural nets could be used to show linkage

between different elements. Another example where these visualizations could be used would be to represent navigation patterns as well as more complex trait associations.

Summary of Literature Review

The literature review has examined some of the extensive research that has been done in a number of fields seeking to leverage the vast amount of information captured in LMS and related systems. Although large repositories exist, in terms of historical data from past cohorts and from real time student logs generated every day, this wealth of information has had limited influence on learning environments. Work in LA and other fields, has built upon the research previously completed on modelling, categorizing, and evaluating various information patterns. ITS and other automated approaches have attempted to use this information to improve systems abilities to provide real time support for educational requirements (Dai, Bao, Yu, & Wang, 2009). These machine-based solutions have not addressed the role of educator and instead seek to replace much of that relationship. While artificial intelligence style approaches may eventually allow systems to simulate a human teacher, that outcome does not appear to be likely in the immediate future. Due to the complexities of learning activities, it seems more appropriate to support educators with tools that make them more aware of student activities. In that way, the challenges of supporting distance education students could be dealt with utilizing tools that currently exist rather than seeking to invent new solutions.

The recently defined field of EDM has worked towards developing goals and mechanisms for leveraging valuable learning data stores. Most of their work has focused on the actual manipulation of information rather than its presentation. LA on the other hand has focused on visualization elements but not necessarily with a data mining laymen in mind. This has formed an important distinction as the assumption many systems make is that users are experts in data mining. While that might be true it has limited the ability to share information with individuals whose expertise may lie in other related areas like teaching. This has formed a critical limitation as one of the goals of support systems should be to disseminate information between experts and communities (Zhang, Kaschek, & Kinshuk, 2005). This is not a criticism of EDM and LA but a reflection of the complexity of the work they carrying out. As discussed earlier, the mining of educational information has been considered extremely complex with highly unpredictable time windows. At this point, the main concern of these fields has been extracting valuable information related to individual systems rather than focusing on ubiquitous learning environments as part of a larger decision making process.

While other areas of research have certainly contributed to the support of ubiquitous learning events at this point the basic concepts have been covered. The next step was to examine the dashboard solution in greater detail in order to determine requirements and expose scenarios that were used to drive the actual development.

Chapter III

Methodology

Given the absence of research specific to decision making tools in a ubiquitous educational environment it was necessary to start with basic elements. This investigation involved providing a working example from which to build and discuss the merits of different solutions. In the case of this research, a prototyping exercise was selected to demonstrate what is possible and to begin the process of identifying elements that may be missing or incomplete. This was advanced by using a dashboard concept which is quite common in decision support systems in the realms of business and systems monitoring. The term “proof of concept” is frequently used for this type of software development as it provides an example which stakeholders can discuss and provide feedback on. This phase also involved feedback from a panel of academics that reviewed and commented to provide suggestions from both a research and practical standpoint. Other individuals have provided impromptu suggestions as the software was developed and presented at in a variety of research discussion groups. In terms of development and architecture, the prototype construction was developed around a set of fundamental assumptions and scenarios which the research targeted as key requirements.

Assumptions

In order to understand the construction and evolution of the project there were a few key assumptions made to simplify the prototyping process. First of all an optimal situation was proposed; one in which bandwidth and hardware limitations were not deemed important. The thought was that over time these issues would be resolved mitigating any programming challenges stemming from these attributes. This did not preclude good development practice but simply recognized that network speeds and hardware performance have been improving at a significant rate. A related assumption was that educational data mining techniques would be able to provide the information needed. The reason for such a dramatic statement was to abstract the prototyping from the complexities of data acquisition and refinement. Considerable research will be necessary to collect and aggregate information but the prototype itself is a consumer of data mining output and thus forms a discrete layer on its own.

In terms of philosophy, an open access approach was taken for a number of reasons. Given the ubiquitous nature of student interactions it was deemed necessary that data would have to flow as a free resource. Any such tool would need to draw on a variety of information sources without encumbrance. This was complemented by choosing technologies that are free and openly accessible on the Internet. The objective here was to create an example that could easily be used, replicated, and modified in order to promote future work. This also recognized the importance of selecting tools that meet community standards in terms of implementation and documentation. A prototype should set the stage

for formal development thus making these considerations important even though they were not critical to the successful outcome of this project.

A final assumption was to take a western perspective to the development of the prototype. By this, it was simply meant that the colors and shapes used were based on the researchers “norms”. For example, red was considered a caution color where green has been considered positive. Shapes like “stars” were also considered to be points of importance although other items like squares and triangles make be less obvious. Legends were also identified as being important to determine meaning and provide support to the end user. This was not done to limit the design, merely to recognize that the use of certain colors and shapes might not be culturally universal. The long term requirement would be to make these elements customizable for each teacher. For example, the use of red as a warning color might not translate to other cultures even though it is commonly recognized as an “alarm” color in western culture.

Scenarios

To generate the layout and design of the prototype, a set of scenarios were identified based on other research work and the underlying objective of providing teachers with information about their students. This was also analyzed with visualizations in mind to attempt to identify those mechanisms which provide the most intuitive displays of information. These scenarios were also built with the goal of providing recommendations in mind. This meant the information needed to support the final suggestions so that teacher would be able to make meaningful decisions. In the context of determining requirements this was aimed at solving specific education issues. For example the system might identify a student who has quit working because they are unable to acquire a certain competency. In this situation suggestions would be made to assist a teacher in selecting a mechanisms or strategy to help the student.

Students in Need

The first scenario drew on work by a group from Purdue and by Macfadyen and Dawson relating to at risk students (Arnold, Tanes, & King, 2010) (Macfadyen & Dawson, 2010). Being able to identify students having learning difficulties provides a very powerful tool as it allows teachers to focus their energies on those students in greatest need. It also should also allow them to drill down on specific information very quickly in order to maximize their time efficiently. The proposed scenario examined how this would work in a dashboard environment where multiple objects could be used to display information. When the teacher first opened the dashboard they should be presented with up-to-date information on which students require assistance and some general context on what their circumstance is. Two groups very quickly emerged; those who self-identified by requesting help and those identified as being “in need” by the system. Since these represented different groups they each needed to have their own visualization objects so

the teacher can easily differentiate. Those students flagged by the system needed to be placed into different categorizations. To reflect these groupings specific examples were selected for the purpose of demonstration by the prototype. The idea of stalled students was used; those who were no longer progressing in the course or who had not accessed course material in a number of days or weeks. Another group was represented by those who were simply deemed as at risk due to student profile attributes such as low course prerequisites or low social interactions. By providing these descriptions, teachers could be guided in their review of other information. Another component of the in dashboard display was the ability to see when students become “active” by accessing learning resources. This involved activities such as “logging on” onto the LMS at which point they needed to be identified if they fell into the “at need” category. That way a teacher would know if an opportunity for intervention had presented itself. A final component of the initial screen was to look at how the class fit together in terms of relationships. This entailed a visual representation of which students established peer relationships along with details of their general attributes. From this very high level overview the teacher would also need to be able to select individual students for deeper analysis. Thus, once a student has been “targeted” it should be possible to focus on them specifically to garner more information about their past preferences, location, recent activity, and learning styles. Finally, the teacher should also be presented with a set of options as to what could be done to assist a student. It was also determined that these suggestions should not be prescriptive as teachers function as experts in education who are ultimately responsible for selecting the proper course of action.

The Class Perspective

A second scenario looked at examining classes as a whole. This meant that dozens, hundreds, or even thousands of students could be represented in a series of visual objects. These data graphics would need to provide summaries that illustrated the different groups which comprise the class as a whole. This type of page or tab was required to give the teacher a general sense of the class; the make up in terms of learning styles, knowledge levels, progress, and general satisfaction. The various objects in this group also needed to have a tighter level of interaction as manipulating visualization should influence some of the others. In this way, the teacher would be able to adapt the view and be able to identify groups or individuals of interest. Once again, the navigation needed to allow for the selection of individuals or groups for deeper analysis.

Another perspective on the class concept was to look at how the entire group was progressing. This view was made more complicated by the potential for open intake which allows students to start courses at any time rather than as discrete cohorts. This was partially alleviated by grouping students in ranges of start dates which could be grouped dynamically. High level statements about the class as whole were also defined looking at satisfaction, knowledge level as well as progress. Further elements that required attention

were also grouped at a class level to identify when students started falling behind or developing other learning needs that put extra stress on educational resources in a class situation. This scenario drew attention to that fact that while content resources are quite large, support resources quantified by teachers or teaching staff were not.

Examining Groups

There were two categories of groups proposed for the purpose of this project; pre-defined groups from projects or other learning activities and those created dynamically as a result of selections taken from visualizations representing the entire class. Pre-defined groups could be evaluated and provided with suggestions for future actions a little more easily as they tend to be at similar stages in their course work. This also made other comparisons simpler such as looking at knowledge levels, learning styles, and risk factors. By contrast, examining ad hoc groups was about discovering patterns or looking for commonalities that may be impacting learning success. While it would be possible to provide suggestions, those would need to be as real time as the group creation. In a situation where data can be vast and complex this presented a number of difficulties. Due to this observation it was not considered viable, thus it was decided that no recommendations would be illustrated by the prototype for the impromptu groups.

Student Perspective

The prototyped solution has been developed as a teacher tool but this does not mean the student perspective should be ignored. Instead it is important to recognize that the actions guided or suggested by a decision support tool are designed to have a direct impact on student learning activities. This is important as the proposed system mitigates the risks of taking a fully automated approach. There is always a teacher who ultimately makes the final decision and while fallible they have the ability to evaluate their decisions on an ongoing basis. They can also establish direct communication where possible to gain additional information the system may not have access to. From a student's perspective this means they can receive support in a number of ways including a variety of communication methods and learning resources. As the student works through the learning material they can be assisted directly by teachers, tutors, and peers as they are met with the various challenges presented by course work. They can be more subtly supported by means of information that is provided to them environmentally in the form of different learning options. This should result in a more robust learning environment that the student is able to navigate and explore with the support mechanism needed to make them more successful.

Teacher Decision Making

Returning to the earlier statements guiding this project, the ultimate purpose was to allow teachers to support their students. This was determined to mean decisions would be focused on current and future learning challenges identified by the system and the teacher. The prioritization of these issues was the first consideration as teacher's need to

be able to compare and evaluate different learning activities to see which require immediate attention. These needed to be broad categorizations which could be quickly branched into discrete choices. Once the teacher has made certain selections they need specific information about learning problems. An example would be identifying a student who is having difficulty understanding a concept. Around that issue are a number of other details such as how long they have been attempting to understand the concept, what resources they have used, and what their understanding level of previous components has been. Once that information has been displayed, a series of options can be presented to allow teachers to select the actions that could be taken. It should also be noted the learning problems can be very time sensitive, if a student has difficulty grasping an idea it can easily have a negative impact on related concepts that are subsequently presented. They may even develop an inaccurate understanding that would need to be unlearned and relearned at a later time. This can have a dramatic impact depending on the length of time that transpires between a learning problem and its resolution. The worst case scenario is, of course, the outright failure or withdrawal of the student which essentially stops the learning process all together. In this situation, the teacher needs to be made aware of the problem, provided with supporting information, and given a set of potential solutions within the context of a relatively condensed period.

Requirements

To this point, the description of the prototype has been at a very high level looking at guiding views and general areas of functionality. In order to generate a full working example, a set of discrete requirements were necessary to base development on. Teachers have been described as experts in the field of learning but not necessarily in those areas that more closely align with computer science concepts. This specifically refers to the challenges of data mining and software development which are critical to the construction a decision support tool. This means that the system needed to be highly accessible with a limited need for training or skill acquisition in order to use it.

- Highlights that need attention
 - Ability to categorize, organize, and display students who need assistance
 - Notify the teacher as students log on and become available
 - Show recent help requests
 - Indicate students who are at risk
 - Provide an overview of student connections to one another
 - Summaries of class as a whole: progress, satisfaction, and knowledge level
 - Capacity and use of resources in the class as a whole
 - Ability to select specific students to see additional information
- Student specific elements
 - Summary of student: who they are, program, contact information
 - Preferences for learning objects

- Location information
- Domain knowledge of student
- Recent student activities
- Learning style evaluations
- Potential recommendations for teacher to work with student
- Class summaries
 - General location information – assume multinational class
 - Knowledge levels of the students in the class
 - Learning styles – strong, moderate, balanced
 - Progress of students – assume different start times
 - Status of students by how they are progressing; behind, on or ahead of schedule
 - Show high level averages of class performance
 - Ability to select groups or specific data items to “drill down” on specific information
- Groups
 - Summarize group with quick details
 - Location of group members
 - Summary of knowledge and progress
 - Potential recommendations for teacher to work with group
- General items
 - Use of a variety of visualization techniques
 - Ability to move and remove elements
 - Break information into discrete views or pages
 - Ability to link elements together; action in on impacts others

Architecture

The listed requirements made up the functionality of the system but the choices of technologies and how to implement them were just as important. The purpose of a prototype is normally to demonstrate the effectiveness of underlying design and related infrastructure choices which represent expressions of the projects general objectives. In the case of the dashboard prototype, there were several high level objectives which formed guidelines for the selection of development options. One of the initial aims was to make the prototype as open as possible to allow others ease of use and access. This precluded proprietary tools that, while potentially helpful, would have restricted future researchers from utilizing the design. There was also a need to make the system modular in a way that allowed other researchers and developers to add and modify the design. The prototype was also designed to be easy to setup and run in as many environments as possible in order to make it usable in a number of situations. Given the online nature of the environment that the prototype was created to support it also made sense to utilize web based technologies as much as possible. With these guiding statements a number of

technologies were selected. Given that web based technologies were preferred, an HTML5 standard was selected to see if an entirely browser based interface was feasible. That led to the selection of a basic web server to handle programming logic and serve of pages. To this end the Apache web server was selected which can be freely downloaded from the Apache foundation (The Apache Software Foundation, 2012). That provided additional development flexibility as well. The Apache web server works well with the Tomcat application server along with a number of PHP server implementations. This allowed for “add on” other options in case functionality from other programming languages was needed.

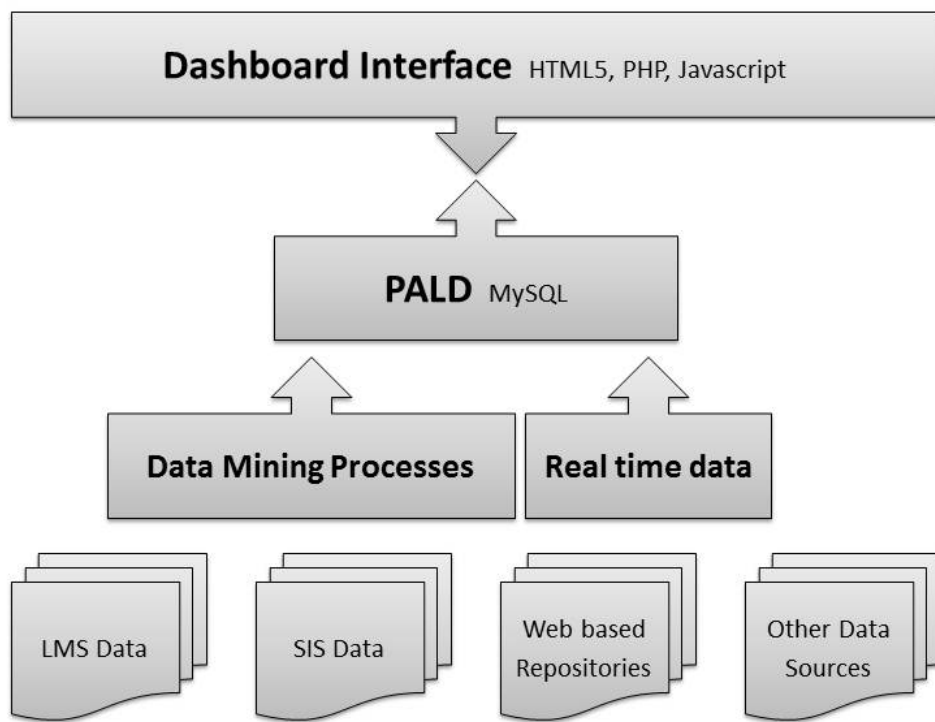


Figure 1- High Level Architecture Diagram

From an information use perspective a middleware database was proposed. This essentially created a multi-tiered application in which the intermediary database forms a discrete layer between the display and the raw information that is found in LMS, web logs, and other sources as shown in figure one. This provided an important distinction as it fundamentally abstracted the dashboard development from the complex details of data mining and other large scale data manipulations. While this design represented a relatively simply construction it still achieve the goal of illustrating a more modular

approach which could be linked to different solutions on the data analysis side. The prototype could just as easily have been linked to Hibernate or some other middleware capable of managing data connections. This was not done to marginalize the data component but simply to recognize its complexity which placed it out of the scope of the project.

Information Requirements

One of the most important considerations that needed to be determined for the dashboard decision support system was what type of information needed to be displayed. For the most part the data used had to be aggregated, meaning that it summarized large chunks of raw data from sources such as LMS logs or Student Information System (SIS) databases. Given the use of a middleware database discussed in the architecture section it was possible to simulate data that could be produced from data mining activities. Certain data constructs needed to reflect this meaning, for example, student understanding was expressed as knowledge levels rather than assessments. In this situation an assessment or collection of assessments existed as discrete pieces of information which needed to be merged with other sources of similar data to create a harmonized understanding to provide greater value and validity. There also needed to be a blend of both real time and historical data to form a more robust data portrait of students, groups, and classes. Notably, historical data should undergo greater aggregation as longer timeframes would be available for performing data analysis. By contrast, the closer to real time information became the less refined it would be such as in the case of student activity and location data. These would essentially be examples of discrete facts that would have meaning for the teacher and student in the current context.

Specifications

One of the advantages of working with a prototype was that specifications tend to be far less stringent in terms of performance and functionality. This did not entirely preclude these considerations as any proof of concept development still needs to be capable of demonstrating a reasonable level of interactivity. There also needed to be the recognition that certain levels of speed and function would be necessary in a final product. From a performance standpoint, the responsiveness of the application needed to be consistent with any web application, with changes between tabs or pages taking a second or two. Changes within a page or visualization element needed to be virtually instantaneous as the data set would already be loaded. This might seem obvious but it had implications when examining complex data that required considerable manipulation prior to its actual display. That also had a direct bearing on the timeliness of the data itself. In a perfect scenario, information would always be real time and constantly updated in order to ensure that the application always reflects the most up to date representation of students and their learning. This was not deemed realistic as “big data” analysis can require hours or even weeks to process in order to identify critical patterns. This suggested a hybrid approach

where certain data elements were provided in near real time, within seconds or minutes. Any data in this category would need to be simple, representing discrete factors or simple patterns that could easily be processed and displayed. Location and basic tracking information were two such examples that could easily be drawn from data sources and shown in a graphical format. On the other side, elements such as learning styles and student preferences can take too long periods to model or evaluate. Fortunately, those types of patterns are quite stable meaning they change very slowly or rarely over time. Thus, once they were established the system could display them quickly and regularly with a relatively low risk of them being incorrect. It should be noted that within any such algorithm there exists a degree of certainty meaning that errors can occur although that margin should decrease over time.

In terms of rendering data, the graphical interface needed to meet a number of criteria that were both specific and abstract. The display needed to be clear and concise utilizing techniques which could translate large chunks of data into representations that could be quickly interpreted. Use of tooltips and other built in supports were also critical to making sure that features were easily accessible and that any specific functions were clearly represented. To this end, point and click features needed to be prominent to support ease of use and supply the ability to quickly change data perspectives and function. Various types of charts like bar, line, and pie have commonly been used in a number of applications which made them ideal for leveraging transferable skills learned in other software. More complex elements like interactive location maps, tree maps, scattergrams, and heat diagrams followed in the same vein although with added degrees of functionality. More importantly, these objects needed to contain discrete features allowing the teacher to select, filter, and sort them. There also needed to be related elements that worked together to create a larger picture. For example, a set of diagrams could be used to illustrate a student's learning preferences, activities, and knowledge level to give a teacher a holistic overview of learning behavior. Navigation was also critical as it made it possible to move down through data "levels" from global elements to more specific pieces of information. In this way, a teacher could move from an overview to the specific elements needed to make final decisions. Within the context of these visual elements, colors were also identified to be used to flag issues although these were used sparingly so as to avoid overwhelming the teacher. Red was selected as a commonly used color for this purpose, but if half the screen was this color it ceased to have any real meaning. Likewise popups and other disruptive elements were used as little as possible as they broke the natural flow of navigation and created distractions. In terms of functionality, the prototype did not need to represent fully developed interactivity but did strive to demonstrate basic implementations that were able to show the capacity to display and manipulate discrete visualization objects.

With the prototype being web based, it was also designed to be as generic as possible staying away from browser specific functionality which would prevent proper operation on certain platforms. Likewise proprietary programming elements were also avoided in order to support a Creative Commons approach that could be easily leveraged by practitioner and researchers alike. The modular approach also allowed future development to easily move between other back end data sources which could drive the prototyped visualizations. As mentioned in the architecture section, the goal was to separate the data manipulation and display layer as much as possible. This was considered at a very high level as some manipulation would still be necessary to make sure similar data formats were being used at the presentation level.

This concluded the analysis phase of the project providing the details necessary to drive the development component. With a set of requirements in place, along with series of guiding scenarios, it was possible to begin the implementation phase using the architectural components selected. This provided a strong basis for building a prototype, although many decisions remained to be made as different technologies, processes and visualizations were added to the construct. This also led to an assessment of whether the initial requirements were successfully implemented based on the analysis components.

Chapter IV

Beginning Development

As the transition began from analysis to development, it was important to move in a deliberate fashion to make effective use of limited development time. This meant leveraging existing technology as much as possible for the benefit of this research and in order to provide ease of access to subsequent development. Such an approach implied that the development phase also included experimentation utilizing a number of different tools to determine which were most effective and usable. This made it necessary to break the development into discrete components as much as possible in order to make units modular so they could be reused and replicated in other situations. Finally, this moved toward the final stage of working with the unified dashboard to provide a working proof of concept following the requirements defined and replicating learning support scenarios.

Underlying Framework

The proposed architecture of the prototype required a highly robust mechanism for displaying visual data. It also identified an idealized situation which involved a platform independent web based design. While a variety of web based tools exist in a range of different languages and libraries many of these come from either closed architectures or involve complex development environments. One of the other goals of the project was to make the solution easy to adapt with a readily available toolset any researcher or developer could access. With these objectives in mind, it was technologies like HTML, HTML5, JavaScript, and PHP that became logical starting points. These languages are all openly accessible with readily available development environments that can quickly be setup with little or no cost.

The initial focus was to see if it was possible to construct the entire interface using HTML5 techniques. This path was chosen largely due to the emergence of HTML5 as a cross platform standard. JavaScript as a primary tool of this mechanism offered a number of libraries that could be used to provide visualization objects. In fact, more than 25 discrete libraries were found in a relatively simple web search offering a range of options and functionality. Based on the number of objects offered by these libraries and their open availability three in particular were selected; D3, InfoVis, and Google Visualizations (Introduction to Using Chart Tools, 2013) (Graph Visualization Library With jQuery - Arbor, 2013) (Google Developers, 2013). These libraries provided a range of functionality including tables, pie, bar, and line charts as well as more complex representations such as scattergrams, gauges, heat maps, location maps, and node diagrams. These three library options contained the building blocks for developing a dashboard capable of demonstrating a number of different functions. While other resources may have offered comparable options, it was necessary to select a fixed number of resources given the time limitations of the project. Several weeks or months could

easily have been spent researching and evaluating different options. The number of libraries did give a clear indication that sufficient web based resources exist to build a wide range of multifunction visualizations for a dashboard.

The framework of the dashboard was also examined using much the same criteria as the visualizations. The example that was initially drawn upon was iGoogle, which demonstrates the ability to add, remove, and customize dynamic components within a tabbed interface (iGoogle, 2013). This provided a highly functional design as it demonstrated a flexible environment with the capacity to grow and contract based on the needs of the end user. Different code examples were found that provided similar functionality utilizing HTML and HTML5 techniques in order to keep it in alignment with the rest of the prototype (eksith, 2013) (Padolsey, 2013).

Review of Middleware Database

The second major element in the prototype architecture was the selection of a data source which could house the information necessary to illustrate the dashboards functionality. Existing work was leveraged in the form of the Personalized Adaptive Learning Dashboard (PALD), a data store which had been designed to reflect the information needed to support an adaptive learning environment (Index I) (Yang, Kinshuk, & Graf, 2010). This data store contained a number of complex constructs including learning styles, domain knowledge, concept maps, and student interests. It also housed discrete data elements such as location information, student demographics, learning objects, and course information. The purpose of this data store was to provide data elements which could effectively be translated into dashboard components to make it easier to display information based on data mining and aggregation activities. While this database did not contain all the elements necessary for the prototype, it provided an excellent starting point. In order to keep with the open technology requirement, MySQL was used to implement the database structure and to serve data to the prototype interface. A PostgreSQL implementation could just as easily been used and could easily be utilized in any subsequent replication of the project.

Working with the Data

In the initial creation of the dashboard visualization elements were introduced with static data generally in JSON format which is commonly used with HTML5 techniques. JSON data generally comes in static files or can be hard coded into JavaScript files. JSON data objects can reflect highly complex data structures incorporating headings along with parent and child information several layers deep. While this worked well for the initial demonstration of functionality it quickly proved problematic when trying to implement a more dynamic data flow. Within the context of programming, this was rendered as highly complex data strings which proved to be very difficult to build especially when pulling dynamically from a database. This also required the use of an extra “tier” of programs which were responsible exclusively for building JSON data elements. While such an

approach might be desirable for separating display, operational logic, and the database it introduced a very high degree of complexity. It also required additional programming for linking the layers together to provide a fully interactive experience. Given the prototype nature of the project and the time limitations this level of overhead was not realistic. It was also questionable whether a dashboard application would benefit from such a complicated design as its purpose was to render data rather than perform complex operations. Whether this would be true or not fell outside the scope of the current project but should be revisited in subsequent research.

As the project evolved, the design moved away from a pure HTML5 implementation to a hybrid solution which involved the introduction of a PHP component for accessing and loading fields from the database. PHP was chosen for as it has proven an excellent mechanism for querying and building data structures and provided a great deal of flexibility when implementing different visualizations. The PHP components were also built in such a way that the prototype interface could easily be converted back to an HTML5 mechanism utilizing JSON data. This provided the best of both worlds giving the prototype a quicker implementation while providing the opportunity for future evolution of the architecture.

Implementing Visualizations

As mentioned in the in the earlier section on libraries three options were selected for the initial implementation. In the first phase which involved static data; InfoVis, D3, and the Google API's were used to very similar degrees. As the project began to incorporate the database, this quickly shifted as certain challenges began to emerge. While the InfoVis and D3 libraries provided impressive visual elements they appeared to have been designed independently of data considerations. What this meant was that it became highly challenging to implement data changes or variations which could occur as new views of information needed to be implemented. They also came with very little documentation requiring considerable analysis in order to modify and enhance them. New functionality almost always needed to be built from scratch requiring extended periods of development for very little functional gain. With the Google API's these problems were quickly alleviated, the documentation proved excellent with a number of working examples that could be easily modified and implemented. These visualizations were also implemented based on a table object which could be loaded from either JSON data or from a traditional array structure. Multiple visualizations were also developed based on views of the same data object making linkage and interaction much more effective. This quickly changed the development path of the dashboard as a number of new visualizations were incorporated from the Google API library including a few which replaced examples of other sources. In the final review of the prototype, the Google objects made up roughly 80% of the visualization objects simply because they proved much more effective and accessible. They have been developed as part of Google's open access strategy, meaning they should

remain available for some time in their current form. Google is also committed to enhancing many of its projects which makes growth of the visualization library highly probable. This directly supports the goal of making the prototype accessible to a wider audience capable of easily replicating the functionality.

Selecting Visualization Types

In terms of the objects themselves, a number of different techniques were utilized to illustrate a range of functions. As with any good prototype, the aim was to demonstrate what was possible and provide examples of a number of different objects. The most basic of these was the table structure which, in spite of its simplicity, still remained on the most effective ways of organizing two dimensional data. To make these objects more versatile, they needed to be sortable and contain links that could be used to navigate to other parts of the dashboard. The ability to page and break information into meaningful subsets was also essential to illustrating how table objects could render complex information. These could also be linked to other visualizations to provide multifunctional interactivity. Basic graphing was the next logical step which was implemented in the form of bar and line charts which expressed tabular data with additional dimensions. For example, knowledge levels could be shown across time and learning objectives to give teachers a more effective view of the class's performance. Likewise, line charts proved an excellent way to show trending and compare the performance levels of different students.

In terms of more complicated elements, location data was rendered in a number of different ways. In the dashboard prototype three different types of maps were shown to illustrate how student location could be rendered. The use of tagging techniques showed clearly where students were located and when they were clustered in certain physical areas. This provided an excellent opportunity to demonstrate how the dashboard could assist in identifying opportunities to partner students who might not otherwise have such an option in a distance learning situation. Maps showing larger geographical areas as in a global heat map was also implemented to demonstrate how they could be helpful in showing where students were located in terms of countries on other granular locations. This also assisted in identifying potential communication issues or other disruptions that might occur as a result of geopolitical and environment issues.

In terms of relating students, a node map was also selected to show links between students. The implementation demonstrated in the prototype was quite simplistic even though the potential was quite large. Many different elements could be flagged in such an object including relationships, student attributes, types of interactions, and patterns of activity. Due to time restrictions these were only addressed in a very cursory fashion but other research has illustrated where this could be significantly expanded (Correa, Crnovrsanin, & Ma, 2012). Likewise, the use of knowledge trees was also introduced to show the organization of domain knowledge and the degree of student understanding of each section in a sample course. This was color coded to provide clearer queues using red

as a warning and green as a positive. This was based on a western approach and would need to be validated for other cultures.

Another high level visualization implementation involved the use of gauges which very quickly expressed aggregated information in a simple and meaningful way. In the prototype, progress, understanding level, and interest were displayed to give a teacher a simple overview of their class's status. While the implementation of such an object was relatively simple the representation implied a great deal of background information.

The other piece of functionality which was implemented came in the form of recommendations to the teacher. While not a particularly significant from a visualization standpoint, these elements represented an important statement about the goal of the dashboard tool itself. One of the purposes was to provide decision support so that teachers could provide a better educational environment to their students. By introducing the concept of recommendations and recording their outcomes, the system demonstrated the ability to develop intervention patterns of its own. This implied that the decision support mechanisms should collect and analyze data to become more effective at predicting what students need. Such a mechanism is adaptive but only in conjunction with a learning expert; the teacher.

Dashboard Implementation

Dashboard Pages – Highlights that Need Attention

With the examination of the requirements and specifications, a high level design emerged which focused on four key areas. The first was the idea of an overview area which could flag students in need and provide the teacher with the ability to perform a “triage” style of analysis on their class and constituent students. This type of page is frequently seen in dashboard applications which seek to draw attention to time sensitive elements. Within a business application these would apply to reorder levels, daily sales, and critical contacts. A similar approach was taken with educational needs which could focus on the primary recipients of teaching services: students. With this “critical item” idea in mind the first elements revolved around identifying students at risk and providing a mechanism for focusing on them quickly (Figure 2). To make these meaningful, categorizations were selected that could give teachers a sense of which needs they were addressing. In the prototype example, stalled students, and those at risk such as students with low prerequisites and low social activity were identified. These were implemented with an expandable table which could be selected to allow teachers to view lists of grouped students. Within these groups, it was also possible to select specific students to examine them individually. In this way, a teacher could quickly see what types of students need there assistance and in what quantity.

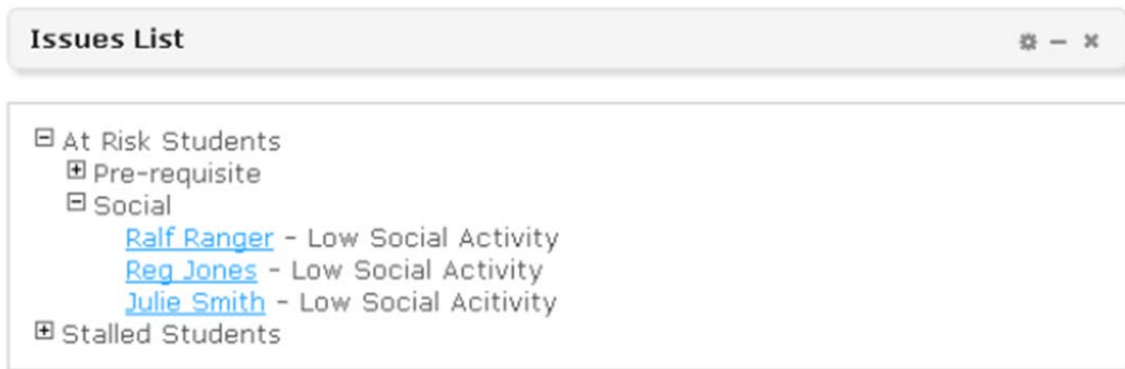
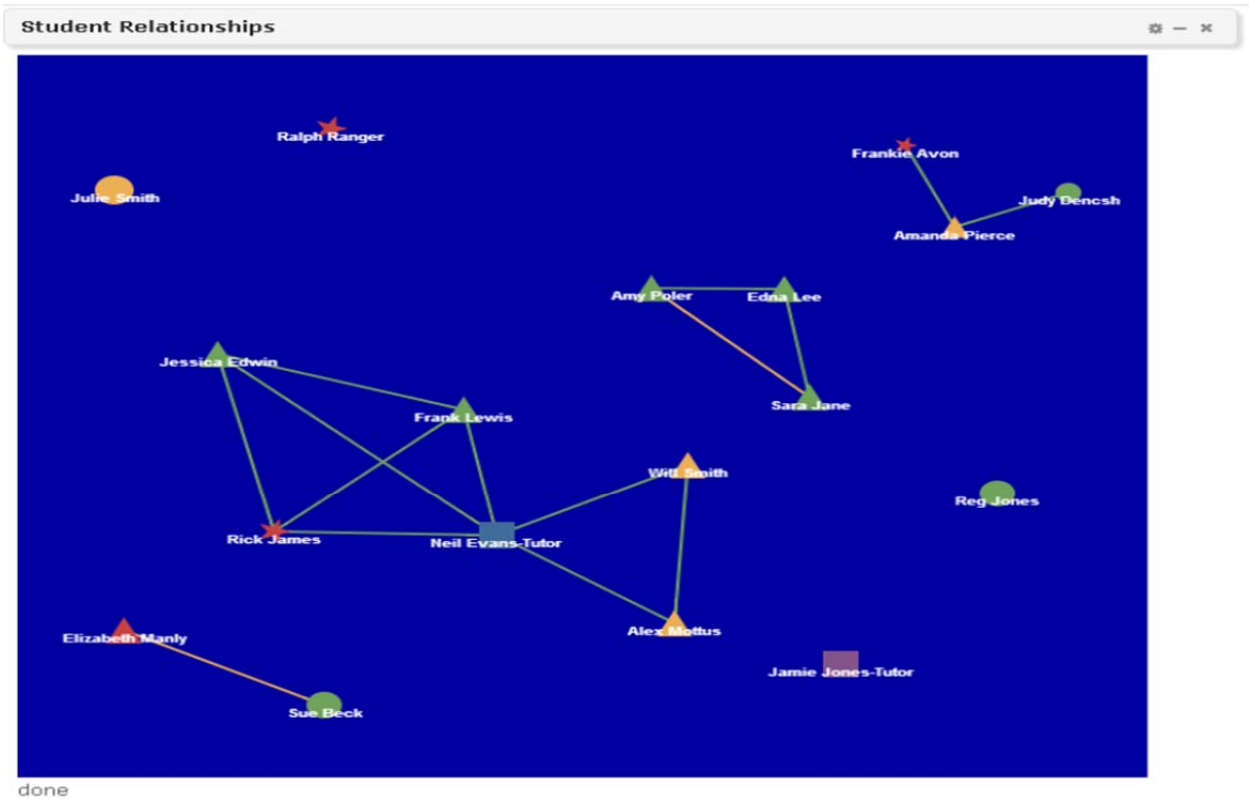


Figure 2 – Example of Organizing “At Risk” students

Another visual object was also added to summarize self-identifiers who had requested help. These students could have been grouped with the system generated categories but because they were self-identifying by using a specific contact mechanism they were split out into their own table. Contact mechanisms such as e-mail, phone, and text were indicated so the teacher would know how best to respond. The goal of these visualizations was to allow teachers to focus their attention on elements that were impacting a number of students, thus maximizing their time and resources. Take an example where a larger number of students have become “stuck” and as a result are no longer advancing through the course. Upon further analysis, the teacher then discovers that the stall point is focused on the third lesson in a course. The teacher could then focus on that part of the course to supply additional resources, send out group communications, or engage tutor and peer supports to focus on that part of the course.

To support the student groupings, contextual information was also selected to provide a very high level summary. A nodal map element was also included to allow teachers to observe very quickly the status of the class as a whole (Figure 3). This included peer interactions, contacts with tutors, as well as very high level information about the students themselves. Due to the complexity of this visualization a legend was also incorporated to relate colors and shapes to student attributes. This type of element demonstrates how several pieces of information can be combined in display component to give a teacher a high operational awareness in a very brief time period. Notably, this was done with a relatively small number of students (17) for demonstration purposes and would need to be refined for use with large classes of 100’s or more. It would also be relatively easy to add additional information about relationships by manipulating line lengths and thicknesses. At this point in the prototype, however, the lines simply represent basic associations or contacts.



Colors - Knowledge Level	Shapes - Student Type
Red: 0-69%	Star: Multiple Risk attributes
Yellow: 70-84%	Circle: Low Social
Green: 85-100%	Triangle: Behind Schedule
Purple: N/A	Square: Tutors

Figure 3 – Nodal map of Students and Tutors

(Legend – Red Star (visible in figure): student with multiple risk factor and an average below 70%, Yellow Star: student with multiple risk factor and an average between 70 and 84%, Green Star: student with multiple risk factor and an average above 84%, Purple Star: student with multiple risk factor and no recorded assessments, Red Circle: student with low social connections and an average below 70%, Yellow Circle (visible in figure): student with low social connections and an average between 70 and 84%, Green Circle (visible in figure): student with low social connections and an average above 84%, Purple Circle: student with low social connections and no recorded assessments, Red Triangle (visible in figure): student behind schedule in course and an average below 70%, Yellow Triangle (visible in figure): student behind schedule in course and an average between 70 and 84%, Green Triangle (visible in figure): student behind schedule in course and an average above 84%, Purple Star: student behind schedule in course and no recorded assessments, Purple Square (visible in figure): tutors)

Another summary item selected was a rolling log to indicate which students were currently available in the educational systems. This element linked very closely with an LMS component which requires students to log in thus identifying their presence. This visualization was also presented in a table format to quickly summarize a list of students and provide critical details such as whether they are also in the “at risk” categories. In this way a teacher would know if an opportunity or “teachable moment” had presented itself. This also illustrated an example of how real time information could be incorporated into

the dashboard alongside other elements that require more overhead in terms of processing.

Finally, the overview section was supplemented with two other components to give a teacher awareness of the demands being placed on educational resources by the class. This was selected to provide a very general summary of the progress, satisfaction, knowledge levels, and resources available to the class. These were implemented using gauges and a bar chart to make them as intuitive as possible. A teacher could use these to determine if the class was falling behind or had exceeded the current educational resources available (Figure 4). If one of these situations occurred, the teacher would then be able to notify administration or draw in other resources to make sure they were properly support. For example, if the demand for tutoring resources had been exceeded the teacher would then be able to demonstrate this critical need and request additional staff.

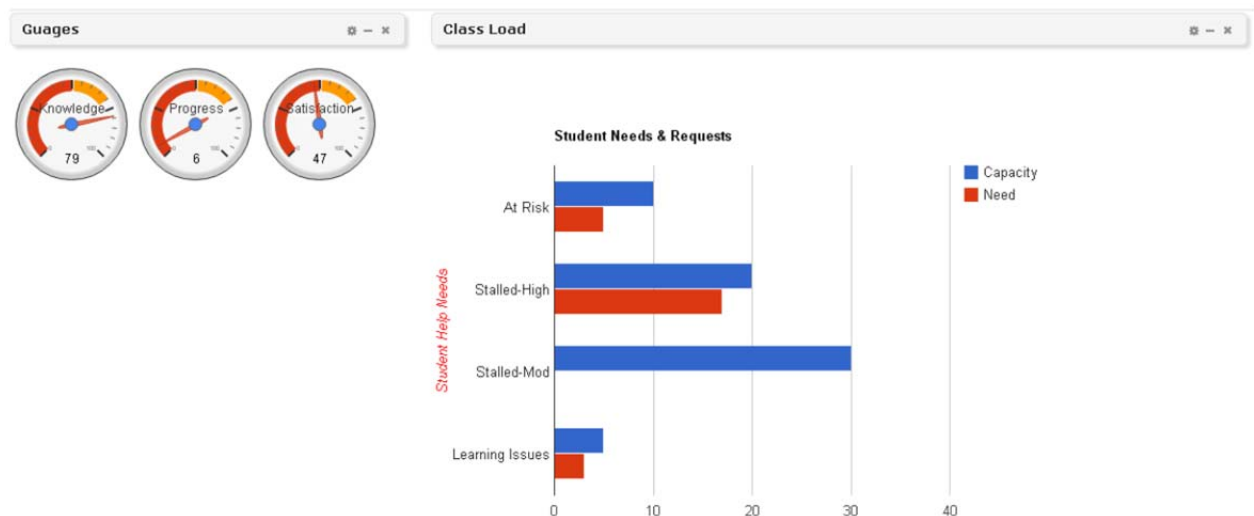


Figure 4 – Summary of Class Summaries and Resource Needs

Dashboard Pages – Student Specific Elements

The second “page” or tab extended the ability to focus on students “at risk”. It did this by introducing the concept of a student page which could quickly summarize student status, learning styles, and knowledge level. This applied to all students but could be driven by selecting a student from the overview page or by choosing a specific student directly. This style of page was chosen to provide a digital portrait of the student which would give the teacher the ability to view the results of complex analysis in unison with ongoing information on location and potential real time learning issues. The first element focused on a demographic overview to give the teacher a summary of who the student is, where they are in their studies, and a quick summary of any dominant learning style traits. This was complemented with a graphical object that summarized their use of learning tools according to different types; reflection quizzes, self-assessment tests, discussion forum

activities, additional reading materials, animations, exercises, examples, and real-life applications (Figure 5). This was mapped across the students last three courses (if applicable) to give the teacher a sense of the students trends in learning object use.

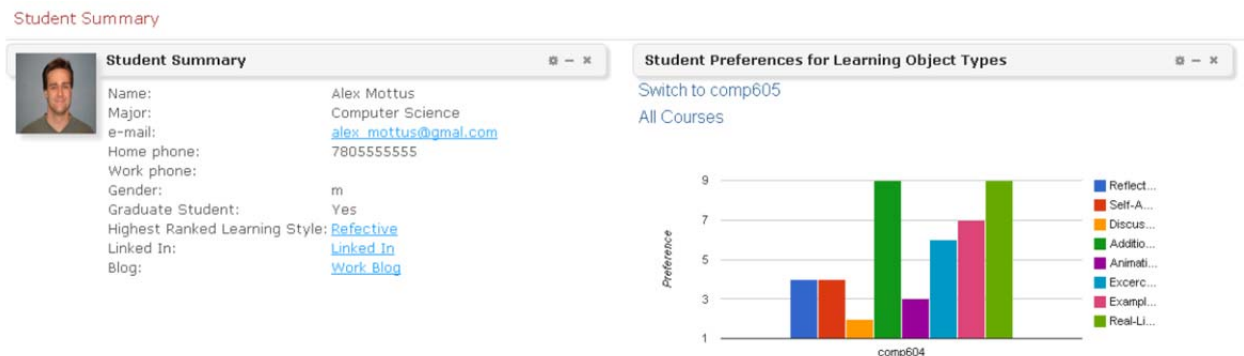


Figure 5 – Student Demographic and Learning Object Preferences

Location data was also included to provide the teacher with an overview of where the student was physically present. This was enhanced by showing what resources, including peers, existed in the immediate area. The student was shown as a different color icon to allow for quick differentiation along with tooltips for additional detail (Figure 6). This gives the teacher the ability to identify real world learning objects as well as the potential to form groups that could support learning activities. In terms of tabular data, a summary of recent learning activities were also included that provided a simplified log of key educational activities. Where possible these were also linked to provide access to blogs or shared assignments. This was added to give the teacher an opportunity to perform formative assessment if necessary.

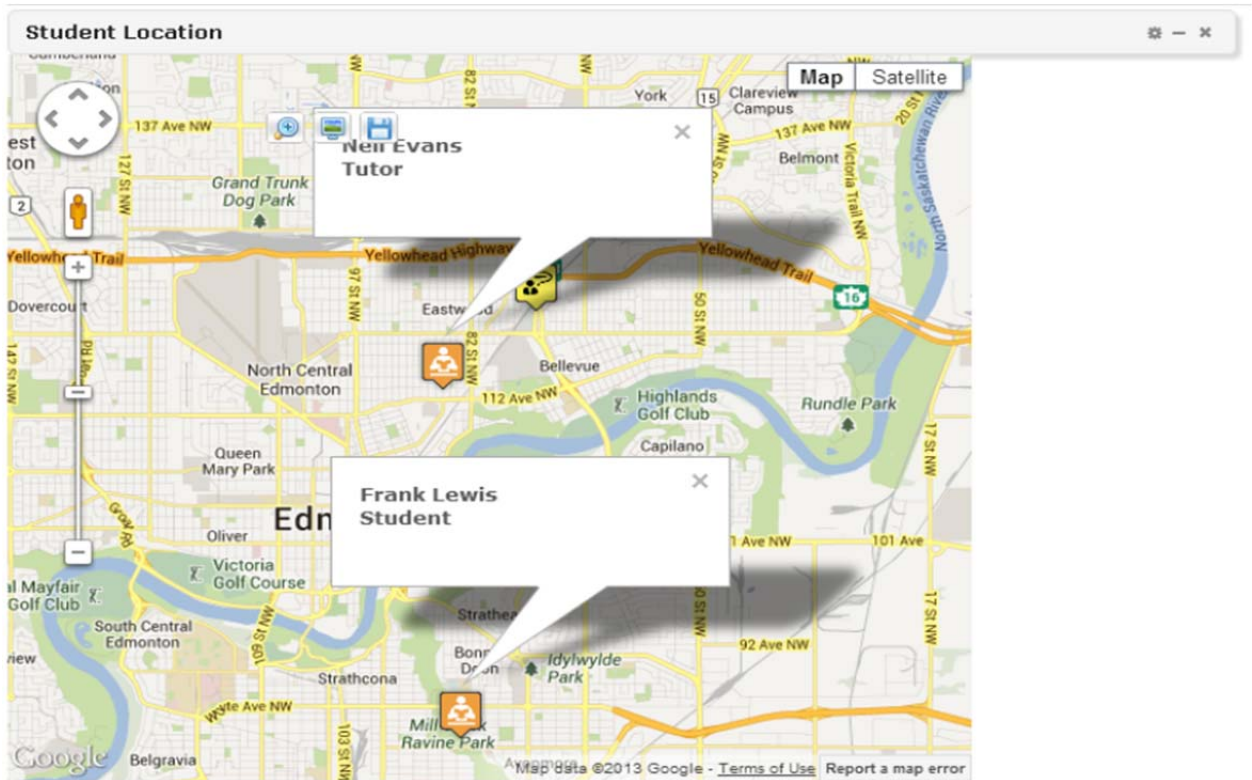


Figure 6 – Sample of Student Location

Another key element identified in the requirements was the ability to track student learning as they progressed through a course. To provide this functionality, a concept map of student knowledge across a series of learning objectives was incorporated as shown in the visualization below (Figure 7). This map was augmented with tooltips and the ability to click on “boxes” to drill down on specific learning objectives to see constituent items and levels of understanding. A color bar at the top of the visualization provided a legend to indicate where the colors exist across a range from red (no knowledge) to green (full understanding).

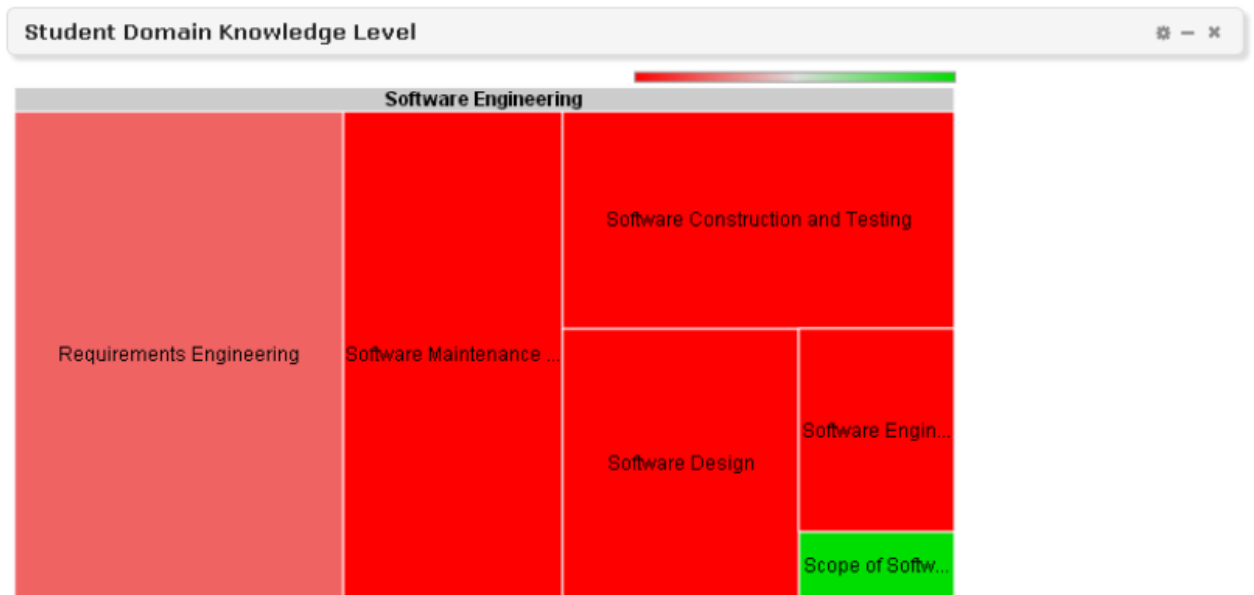


Figure 7 - Example of Domain Knowledge Visualization

(Bar above the words Software Engineer illustrates the range of student knowledge - red as very low understanding, grey as medium understanding and green as high understanding. An arrow appears on mouse over to show where in the range each block exists.)

The student page was also used to display learning styles and how they how evolved over time. This was implemented to reflect the Felder-Silverman model of ranges across the traits of Active-Reflective, Sensing-Intuitive, Visual-Verbal and Sequential-Global (Graf, Kinshuk, & Liu, 2008). This utilized a bar chart design with appropriate tooltips to display when and how the learning style evaluations had been conducted (Figure 8).

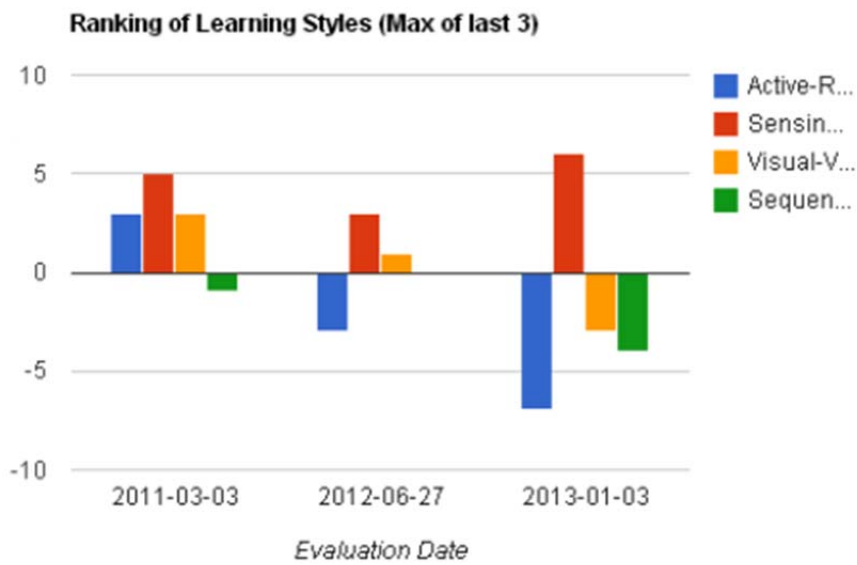


Figure 8 - Student Learning Style

The final element of the student page dealt with the recommendation component that has been discussed earlier. The implementation of this element was kept fairly simplistic at this point although its inclusion had significant implications. It was built to provide the teacher with the ability to select certain options and to comment on whether or not they had been successful (Figure 9). This was designed to be captured in real time as teachers make decisions. This provided options as well as capturing the decision outcome so they too could become part of the systems evaluation process. Ideally a repository of teacher choices should grow for future analysis.

Recommendations			
Date	Actions	Accepted	Outcome
2011-01-13	Phone Call	Yes	Success
2011-01-13	e-mail	No	
2013-02-14	Phone Call	Yes	Success
2013-03-03	Suggest Video	Yes	Success Indeterminate Unsuccessful
2013-03-03	Social Peer contact	Yes	Success

Figure 9 – Sample Recommendation Object

Dashboard Pages –Class Summary

In the dashboard implementation, a third page was added for the purpose of viewing the class as a whole. The goal of this page was two-fold as it provided an overview and the ability to select groups or individuals for closer analysis. The overview was also designed to demonstrate an up-to-date status of the classes learning outcomes and preferences viewed across the entire cohort. Within those elements, groups could quickly be identified and selected so as to provide the teacher with working subsets that could be analyzed in greater detail.

Class Summary

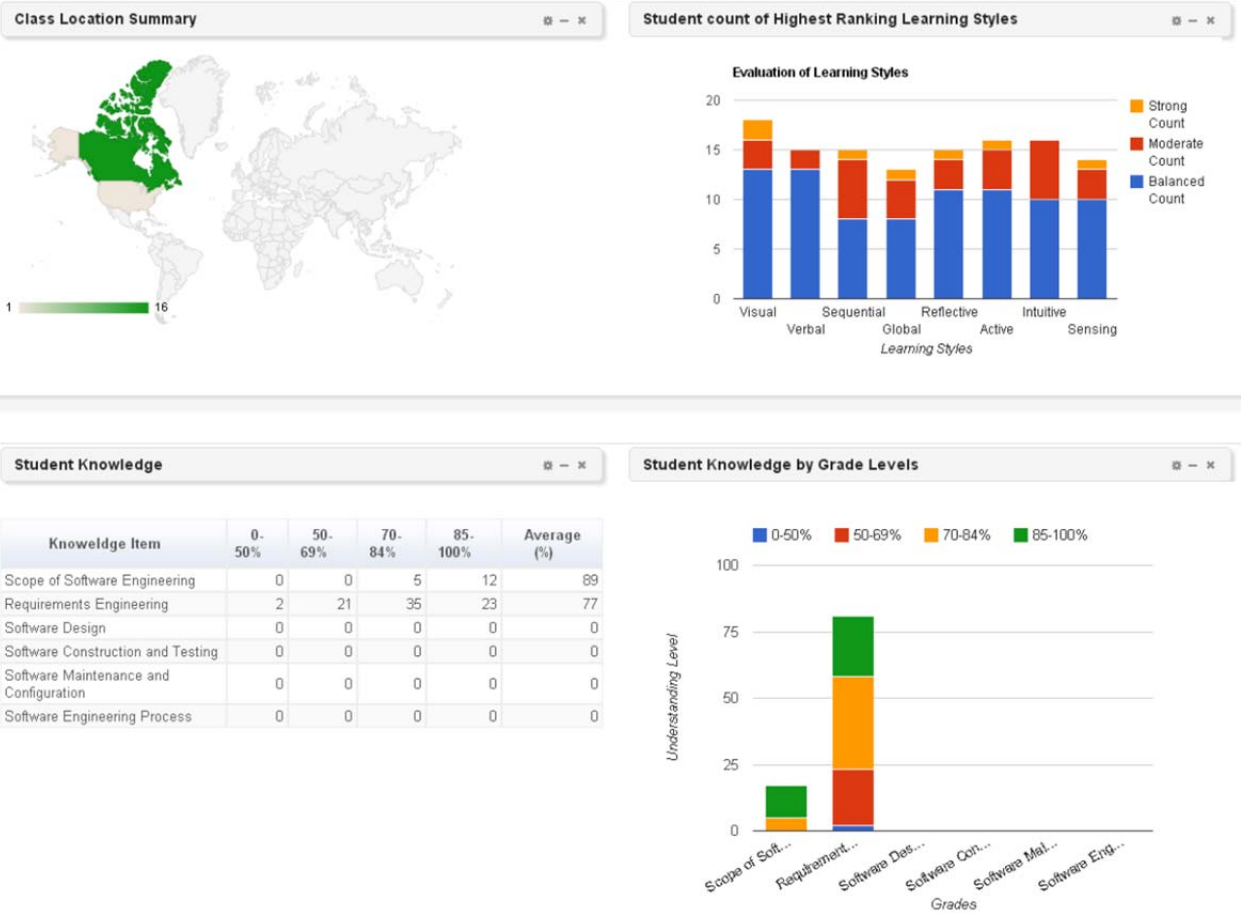


Figure 10 - Sample Class Visualizations

From the perspective of looking at the class as a whole, two concepts were focused on. The first was knowledge level which was selected to draw teacher's attention to how students performed on various learning objectives. Directly related was the progress of the student as they moved through the course. This was done by means of percentages and stacked bar charts which could be linked together to sort and select data (Figure 10 – top and bottom right). These created ranges which were used to make it possible to select groups that could be further analyzed.

Other elements were added to summarize information about the class and give the teacher a visual representation of progress and how the class was performing. This was done by means of a global heat map (Figure 10 – top left) which became darker based on how many students existed in each country. This could be particularly useful for those institutions which have global audiences. This could be broken down to a more localized map if students only came from a certain country, state, or province. Learning styles were also highlighted at the class level to indicate which learning preferences were strongest so that teachers could observe the makeup and needs of the class as a whole (Figure 10 – top

right). Gauges, also seen in the overview section were added here to illustrate class averages in terms of satisfaction, knowledge level and progress.

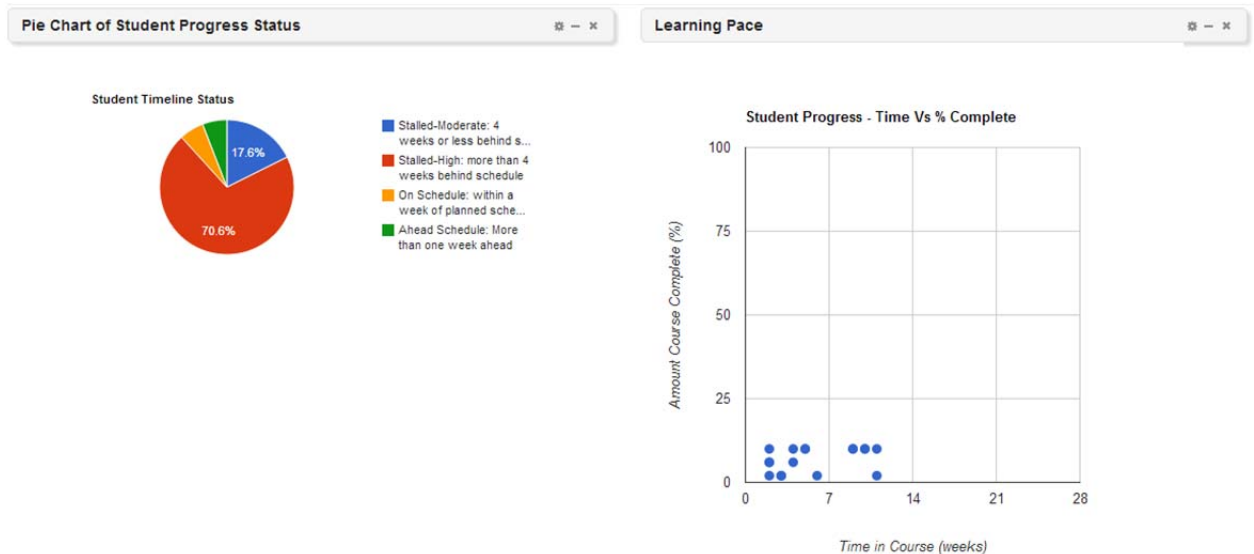


Figure 11 - Pie Chart and Scattergram of Student Progress

One of the requirements was to handle open intake, a situation where students could start the course at any time. This meant that cohorts had to be constructed based on date ranges which could be mapped. Two different visual objects were used to represent this type of information. One was a pie chart which could identify whether students were on track to complete the course within a 3 month cycle. Each slice in the “pie” was designed to indicate if students were, ahead, on, slightly behind, or very behind (more than 2 weeks) in the course. These formed excellent natural groups that could once again be selected as discrete cohorts. The second method of display was a Scattergram which mapped all of the students by time and number of learning objectives completed (Figure 11). This created a highly visual way of identifying “outliers” who exist outside the normal groupings of students. These individuals could then be selected, automatically loading the individual student page with their information.

Dashboard Pages - Groups

The final page was designed to provide a working area for investigating both pre-defined and selected groups. Both of these could be viewed in terms of the composite students. Two types of groups were available for review; one selected ad-hoc from the “Class” page and one from pre-defined groups available in a drop down list (Figure 12 – upper left). Groups in the first category could be selected but clicking on a component of a visualization such as a single “slice” of the pie chart. The latter were examples of project groups that had been formed as a result of class activities. The interface was designed to take the individual learners in these groups and quickly compare and contrast them by a variety of different attributes. In many ways this page was a hybrid between the class and

the individual student view. It was also important to note the flow of the different components. As mentioned earlier, the elements needed to work together and to direct flow logically between pages. The teacher could easily start in the class page, move to a group and end up examining a specific individual. Within these pages, discrete widgets were selected to provide information individually or in concert with other items.

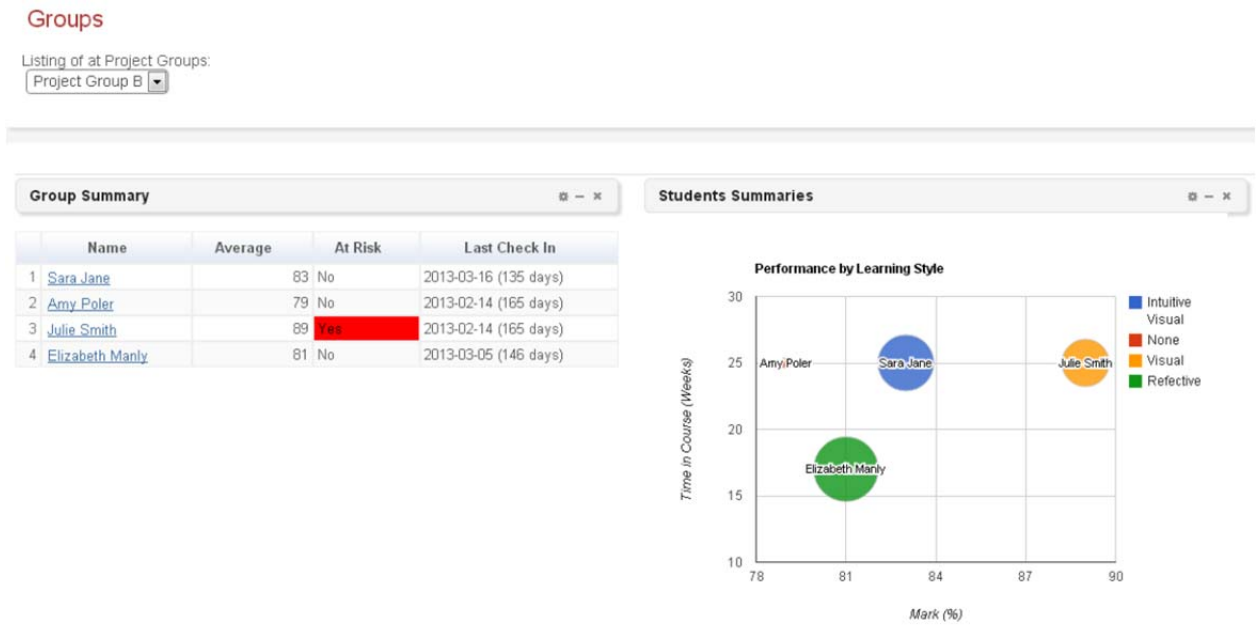


Figure 12 - Groups Selection, summary and bubble chart

Much like the class page, the group page focused on knowledge level and progress but with a greater degree of detail. A tabular format was used to summarize students in the group to show how they were progressing and with what level of understanding (Figure 12 – lower left). Specific students with the “at risk” identification were also flagged with a red color to draw attention. A bubble chart was also used to try and capture several attributes in a combination of visual techniques (Figure 12 – lower right). Circle sizes were used to illustrate understanding levels and satisfaction while colors were used to indicate learning styles. This was done within the context of X and Y axis which indicated time in the course and number of learning objectives completed. This was further augmented by a legend which was necessary due to the number of elements being covered. Tooltips were also used to provide more detail about each “bubble”. A map component was also included to show where students within groups were located combined with more detailed information about student performance and progress which was also displayed by means of a mouse over tooltip (Figure 13 – left side). As discussed in the requirements section, recommendations were also provided for predefined groups which the system could evaluate prior to the display in the dashboard tooltip (Figure 13 –

right side). This element was represented by the “Intervention Actions” object which was essentially the same as the one presented in the individual student section.

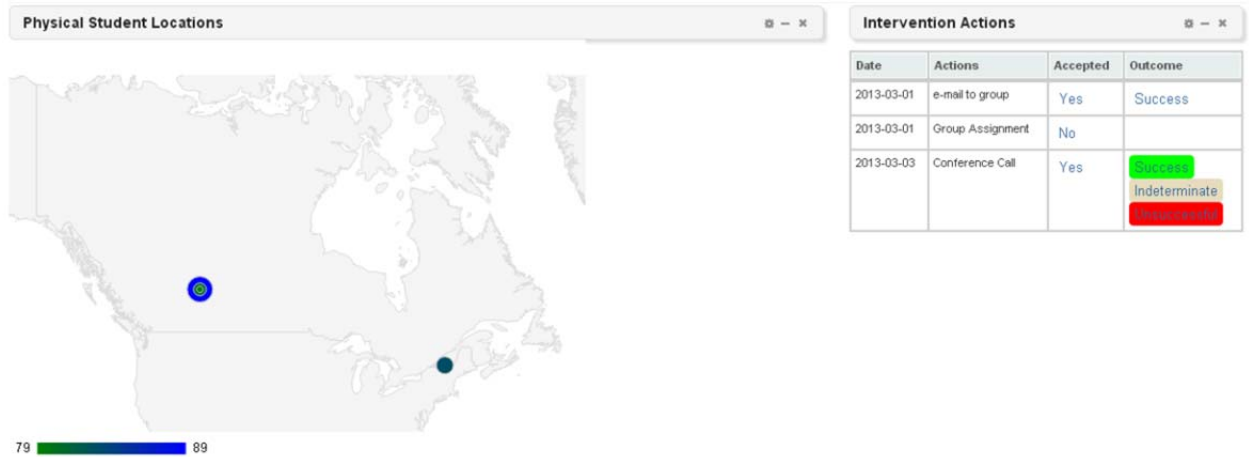


Figure 13 - Heat Map and Recommendations/Intervention Actions

Line and tabular data were once again linked to show how students were performing across a range of different learning objectives (Figure 14). This was made much more concise with the subset as it was able to directly compare and contrast students in specific groupings. One of the challenges in this type of analysis was to keep groups small enough to allow for discrete comparisons. It became very easy to select large groups which were too large to evaluate in context that provided a higher level of detail. This could result in information overload as large collections of details cease to have meaning at a certain volume.

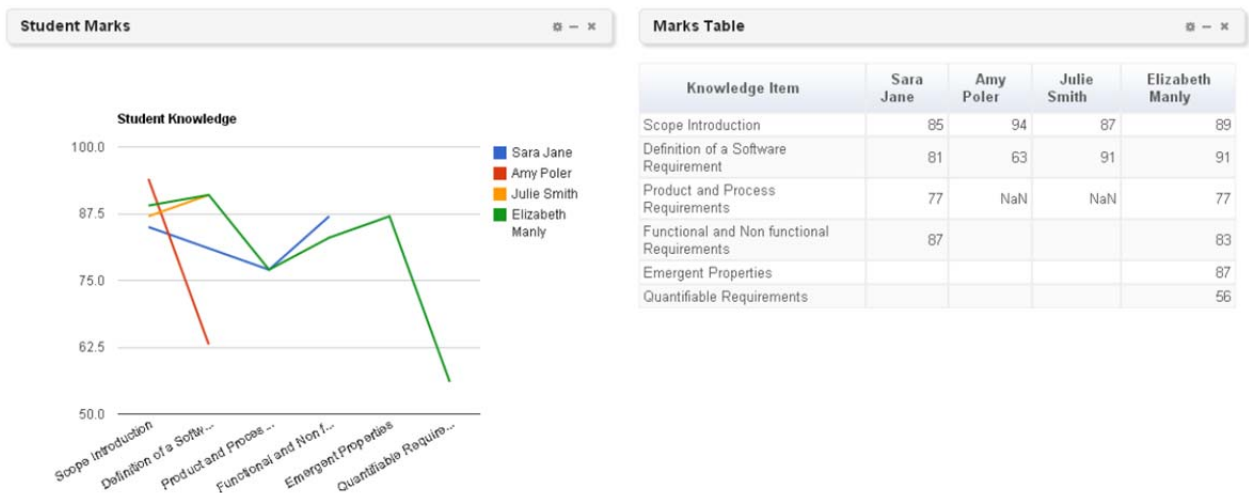


Figure 14- Student Marks in Line and Table Formats

Prototype Function

To this point the discussion has revolved around discrete components in terms of both data and visualizations. While these are important elements, one of the goals of the project was to unite them into a working environment which provided richer functionality as a result of interaction among the various pieces. Some of the functionality related to work flow has been mentioned in passing in the form of hyperlinks and navigation elements. The dashboard was designed to provide two paths for identifying interventions. One was direct in the overview tab which was meant to flag specific students so that the teacher could easily locate and support them. The second mechanism was to allow the teacher to drill down from the class as a whole into specific groups which could be worked with or to drill even further down to once again find individual students. This was supported by the visualization techniques that identified subsets in larger groups like sections of a pie chart which are clickable and allowed the teacher to automatically select the students represented by each element. This combined visualization and navigation in a way that would allow teachers to find and choose the students they were searching for.

One other important observation about the prototype construction was the need to work with both historical and real time data in the same panels. Student weblog and relationship information were two examples of such elements which could work together to allow a teacher to quickly evaluate student needs as those individuals became available for direct communication. This was a key factor as timely communication was identified as essential to providing proper support. The widget style discussed earlier made this type of implementation possible as different elements could represent discrete functionality within the same page. These could also be linked although this type of function was not implemented due to the time constraints of the project.

One of the interesting evolutions of the prototype design involved the use of data as an object. In order to link visualizations and the functions they provided, it became necessary to use highly refined data objects that were focused on specific elements. These information sources gradually evolved to support multiple visualizations on specific pages. In each of these cases, a data object needed to be constructed to drive the various visual components which became views of the data set from both an information and imagery perspective. This concept has been developed previously in data warehousing projects that have utilized data cubes or marts to create working blocks of information. This type of functionality was supported very well by the Google Data Table object which allowed for multiple views and columns of data which could be rendered in various visualizations after being filtered even further.

Validation

With the conclusion the development of the prototype it became necessary to evaluate whether or not it achieved the original objectives of the project. At a high level this was to provide teachers with a greater awareness of their student's activities. The aim was to

achieve this by providing details necessary for decision making along with recommendations for addressing learning difficulties. This also needed to be as intuitive as possible to make the prototype accessible to users with a broad range of skill levels.

To assess the effectiveness of the design and implementation, a number of examples can be provided to illustrate the manifestation of the guiding objectives. In terms of awareness, the dashboard provides a variety of details involving the basics of any inquiry; the who, what, when, why, and where. In terms of the “who”, “why” and the “what” the dashboard highlights students with risk attributes guiding the teacher to examine these more closely. This is done immediately by providing a summary of student’s challenges in the very first visualization. This highlights the focus and importance of finding students with learning difficulties. This is highlighted in System Messages area where students at risk are clearly flagged to indicate when they become available for direct interaction with the teacher. This is also on the student page in the demographic summary visualization so the teacher can quickly reference any learning issues. This is also done in the student summary component on the Group page with highlighting in red to draw attention to individuals with learning issues. From the “where” and the “when” a number of mapping objects are used to illustrate student location and their relation to other member of the class. These are shown on the student, class, and group page to insure that the teacher has a quick reference as in a variety of different areas. On the student page, the student location points are also tagged with time stamps to indicate when each location was last visited. The system messages visualization on the overview page also has a time stamp to indicate when students have accessed the system or become available. Recommendations on the student and group page are also time stamped to give the teacher information about when certain activities have occurred. Progress figures prominently as well especially on the class page where students can easily be compared on the scattergram visualization. From usability standpoint, point and click functionality are used exclusively and are supported by tooltips which provide additional information and instructions for using specific objects. The application also used colors as simple indicators using red and green to identify negative and positive conditions respectively.

In assessing the prototype design, the most important consideration was to compare how it addressed the scenario proposed for teacher decision making. This was selected to reflect the fundamental role of the dashboard as an education decision support tool. In the scenario outline in Chapter III the teacher needed to be able to prioritize student’s needs, review relevant information, and then select actions that could support student learning. The first page of the dashboard addressed the prioritization issue directly by identifying student problems and help requests immediately. This highlighted the two categories of students; those who self-identify and those the system flags based on various attributes. This mix sought to cover the basics of creating an effective support system by not only highlighting students but grouping them into meaningful cohorts. Supporting information

was also proved by implementing a mechanism for displaying real time events. This gave the teacher up to date information so they would be aware if a learning opportunity presented itself. To provide teachers with a highly visual representation of the class as whole a node diagram to create a summary of student relationships and interactions. Overall the overview page also drove navigation by making student names clickable so the teacher could select a student immediately for further investigation and support. This allowed the teacher to observe general groupings and then drill down to a specific student and from there to individual attributes.

One of the categories of the “at risk” students was those who had stopped progressing through the course for some reason such as becoming stuck on a learning concept. This provided a time context to allow teachers to observe the period involved for stalled students to acquire knowledge concepts. It should also be noted that students could be part of multiple categories so teachers can cross reference different learning problems. This allowed teachers to develop a sense of context about where students are at in terms of their learning challenges. Given the time sensitive nature of learning challenges this provided an important detail as it flagged teachers relatively quickly when students began to experience problems, even if they were not manifesting any other learning challenges.

Once a teacher selected a specific student, the dashboard presented them with details aimed providing specifics about a learning situation. The student’s preference for learning objects, their learning styles, location information, knowledge levels, and other details all become available for a teacher to observe a digital profile. The final piece of the decision support system was to provide a set of recommendations for resolving learning problems. These could also be simply to improve the student’s current learning situation although that would be of lower importance for students not experiencing specific problems. For those students with learning challenges that a teacher has focused on, both historical and current recommendations were supplied in order to provide operational awareness. The teacher can go on step further to identify whether or not to select recommendations and then subsequently indicate whether the action was successful. This addressed the teacher decision support scenarios requirement of presenting information and then providing support the actual choices that need to be made. This essentially encapsulated the learning support process as teachers could use the tool to identify students in need, consume relevant information, and finally make a decision aimed at improving the students learning situation. This also reflected a similar process that could happen in any class room but utilizing software as a medium rather than face-to-face interaction.

While functional validation would still be required as a follow up step, it is reasonable to conclude that the prototype does provide a potential solution. It meets the requirements identified in the analysis phase and demonstrates how the basic teacher decision scenario could be accomplished. This research can now be turned toward examining areas that were left unexplored and looking at where future work needs to focus.

Chapter V

Discussion

The construction of the prototype demonstrated that visualizations can be used to create a working mechanism for displaying a wide range of educational data. It also incorporated a recommendation element to illustrate support for decision making in the context of supplying student information. This included a tracking of that process to allow the system to develop its own interaction data in order to set the stage for a machine learning component. With the backdrop of the prototype the discussion can be moved back to the original goal of developing an effective tool for providing teachers with the ability to reconnect with their students. At this point such a mechanism does not provide all the necessary functionality to support student learning. However, it is reasonable to conclude that the components presented do in fact provide access to data mining outcomes in a way that is easier to digest. It also provided a unified mechanism for aggregating a number of different inputs and linking them together in terms of navigation. Further, it helped to identify the types of information that should be focused on when developing data mining algorithms. Complex concepts such as learning styles and the progression of understanding educational outcomes both presented key elements in the information that was displayed.

Part of the challenge of developing an educational support revolved around the complexity of understanding learning as a process. Learning incorporates a wide array of parameters which are inherently fluid. Thus, in order to understand and support the evolution of a student's learning process it became necessary to blend a wide range of behavioral data. One of the themes that emerged from the development of the prototype was a need for information items that were constructed based on a pool of supporting data. For example, rather than assessment data the concept of knowledge level was used through a set of visualizations. This implied that assessment information was simply one element that is used to determine a student's level of understanding. Other competency based activities could have been included along with the teachers own evaluations. Another high level concept that was referenced was the idea of students being "at risk". While visually identifying students as needing help was a simple task it represented a complex analysis of each student's state which could vary significantly over time. As a student engages or fails to engage in various activities their status could vary greatly to reflect their learning state at a given time. This spoke directly to the need for timely analysis which provides teachers with the appropriate details to make educational decisions. Another element which assisted in this process was the inclusion of learning styles which have been used in adaptive systems to drive evaluations and choices between options (Graf, Kinshuk, & Liu, 2008). These spectrums of student behavior could either represent a great deal of student data or have been derived quite quickly from questionnaires or other direct evaluation mechanisms. In the prototype, it was deemed

important to indicate where the data is being drawn from in order to provide the teacher with additional information on how certain elements were obtained. This highlighted the need for validation which is normally absent or limited in most automated implementations of learning systems (Mitrovic, Martin, & Suraweera, 2007).

Support Systems and Experts

A decision support system is by its very nature a tool that is designed to be used to provide systematic supports in a certain field (Bresfelean, Ghisoiu, Lacurezeanu, & Sitar-Taut, 2009). In this case, teachers who bring a range of instructional skills aimed at making learning more effective. One of the challenges of modern education as a whole is the ability for teachers to introduce technical elements into their standard practice. This is particularly true when teachers are required to develop a new set of advanced skills which may not pertain directly to educational delivery or support. If a new tool applies to this scenario it is much less likely that teachers will elect to utilize it. Instead they are more likely to use traditional means which do not require any specific or additional training to use effectively. In the case of the dashboard, the use of point and click functionality with tooltips and other prompts makes it possible for the system to be used with very little additional training.

Once teachers begin to use the system for making decisions they also become a critical source of information as their choices represents a unique viewpoint. They can have years of experience teaching in both traditional and online environments. They also have domain specific training in pedagogy and learning development which allows them to evaluate situations and make important choices. By observing and capturing these activities a decision support system has the capacity to develop machine intelligence around learning activities. Such a mechanism has an additional value as the teachers also become filters and validators for the suggestions the system makes. This is an element that is common in the development of automated systems like Intelligent Tutors but is rarely seen in the as an operational component when the system is regularly interacting with students. That can be problematic if a learning support system develops problems or misreads a situation. It also has implications for how systems evolve over time. The learning process is highly dynamic as students have the potential for intuitive leaps as they develop greater understanding of concepts. This can also be seen in complex patterns of learning which can involve both gains and losses of understanding over time. Due to these variable patterns a system which seeks to support students must be highly flexible, an attribute which is more closely related to human thinking than machine. This suggests that a hybrid approach to systems evolution should be taken where the human expert and the system jointly work to enhance each other and support students.

The use of a hybrid approach would also allow the system to be far more generic in its support mechanisms. One of the problems with automated support systems is that they tend to be highly domain specific. A large number of rules and subsequent programming

logic become necessary to map different situations that the student can become involved in. In this case, the system is simply observing and presenting observations that can be used or eliminated on the part of the teacher. An interesting element of this type of approach is that it could be used with an automated approach to allow teachers to monitor the operation of an Intelligent Tutor or Adaptive Learning System. In such a scenario, the decision support system would passively observe ongoing data mining and student learning activities leaving the more complex decision making interactions at the discretion of the teacher and other educational staff.

Remaining Gaps

Throughout the planning and development of the project, it has become increasingly evident that there are components related to the prototype that require further study. Strangely, the concept of providing decision support in an educational environment is an area that has received relatively little attention from a research perspective. This may well be due to the fact that it is an extremely complex problem which a number of constituent parts. While very specific problems, like dealing with at risk students and solving specific learning problems have been examined, there is still a distinct need for a holistic approach. As learners become increasingly independent and mobile they will need to be able to leverage new and emerging learning opportunities. This suggests they will move further and further away from a traditional classroom meaning that teachers will need to follow if they are to remain relevant to the learning process. Thus additional research is needed to reconnect the learning and teaching events in order to make both more effective. While the prototype illustrates that information can be delivered pictorially in a variety of ways it stops short of measuring the degree to which teachers could leverage such a tool. It also starts to consider the alternative methods for visualizing data but a more exhaustive analysis is necessary to evaluate the different options.

Another key feature that still remains to be explored is the ability to allow teachers to customize the types of visualizations presented in the dashboard. While the prototype addresses the idea of moving, removing, and displaying visualizations it stops short of actually allowing a teacher to begin configuring their own environment. For the purposes of demonstration, four main categories were selected but these might not be suitable for all scenarios. Instead, other predefined cohorts or monitoring of machine interventions might make more sense. For these types of scenarios, a more flexible approach would be needed which would allow a teacher to select and construct layouts that match specific views. The prototype does illustrate that this function is possible in terms of the number of visualizations and objects that can be used. There are also a number of portal implementations that demonstrate the ability to add and remove components dynamically. In these scenarios the user can create tabs or pages which allow them to build the discrete views which form the basis for inquiry. The key detail here is that visualizations need to be modular but within the context of a larger data set. This idea returns to the earlier

reference to data marts or cubes which present a chunk of refined data that can be viewed in several different ways. It should be noted the data cube construction itself still falls within the realm of data mining meaning that it would take expert skills to create. The purpose of the project in general was to abstract teachers from that aspect suggesting that some sort of machine mediated component would be necessary for creating those objects.

It should also be noted that there also needs to be some sort of mechanism to drive questions down through a data mining tier. This means that if teachers develop questions that can't be answered by existing data mining elements it should be possible to have new elements requested that could then be displayed through the existing visualization techniques. This implies a layered approach which abstracts display from data which is a common theme in multi-tiered development models. There is further support for this idea when talking about performance as it is important to minimize processing load as much as possible in the display tier. This would allow for effective display of the decision support tool in a range of devices including mobile. While that was not a specification for the prototype, it is important to keep in mind for any web based development. Just as students are making increasing use of mobile technology so are teachers who are gaining access to a richer array of physical tools. The focus on displaying information ties directly to educational data mining initiatives which seek to understand the different nuances of the learning process. If the goal is detecting learning difficulties and opportunities in a real time situation the focus becomes far more specific. In a world that is filled with large data sources of many disparate types, this ability to form filters is critical to identifying relevant information that can be extracted and refined to the point where it becomes usable. At this point, the data can become student or learning centric in order to provide the insights necessary to drive visualization that have meaning for teachers. Given the breadth of the field, this could well be a good thing as it forms a driver for relating data mining inquiry to real scenarios which can be measured for their validity. Certainly educational data mining has vast promise but unless it can be used to address learning problems in a timely fashion it has little direct value to students or teachers.

A related element is the need to link data mining concepts together in an operational hierarchy that can be rendered graphically. This is the final expression of the dashboard scenario and its promise of harmonizing different concepts. Once again the prototype illustrated how this could be done but it is far from creating a best practice scenario. While it begins that discussion, there are a number of considerations ranging from usability to functionality to promoting learning outcomes that need to be examined. These ideas also need to be considered in the context of how they could be implemented in a dashboard. There is the additional possibility that a decision support tool could be used in parallel with other systems, both automated and human, which already exist to supply and support learning opportunities. Dashboards are by their very design meant to augment other systems rather than replace them. This situation is no different as the dashboard

suggested sits atop many other components which deal directly with students. The prototype design was built to act as a facilitator that could improve other systems as well as assist the expert users who drive them.

Future Vision

When analyzing the outcomes of a prototype, the final element is to consider where it fits in the context of future work and the overarching goals of the research field. The use of a prototype methodology provides an easily accessible feedback mechanism that can be manipulated to add, change, and even remove functionality based on situation. In this manner, providing a concrete example gives teachers and researchers the ability to experiment and evaluate design features in a systematic and meaningful way. This takes the research exercise from a purely theoretical perspective to one grounded in actual practice which can be used to drive empirical investigation. The eventual intent is to provide a real tool that can be used by teachers on a daily basis to achieve greater awareness of their students and to help create a more effectively learning environment that not only rivals but can surpass the results found in a traditional classroom.

Completing a physical design provides a sign post or milestone in developing a final solution. In this case, the teacher dashboard prototype speaks directly to several critical items. First, it recognizes and supports the position of a teacher as an expert in the field who can provide important insights and essential supports to make learning successful in a distributed environment. This is an element that needs to be complementary to automated approaches which seek to provide educational supports in real time. This refers directly to the creating checks and balances to validate machine learning with an ongoing mechanism. This means that a teacher dashboard could and should be incorporated into AI approaches like Intelligent Tutors, Learning companions, and recommender systems. That is a notable element as these systems often include expert input in the development phase but neglect those benefits once the system is up and running. An advantage to teacher dashboard “Add-on” is that the system itself needs to be less complex as highly complicated decision logic can be off-loaded to the teacher who can assess student needs based on past experience, intuition, and other skills that are more applicable to human cognition. This highlights the interdisciplinary nature of the teacher dashboard approach as it is designed to enhance and assist the results in other areas of educational support and information development. While data presentation and visualization represents a discrete area of research it is one that relies heavily on the work in other fields like EDM. It is imperative that development of visualization techniques should be done with a high degree of awareness of data mining developments particularly in terms of developing complex constructs around student knowledge, risks, and progress. This is an important feature as it is important to leverage work in other fields rather than replicating it. Visualization and dashboard components are consumers of automated approaches and the results provided will only be as effective as the techniques they leverage. Thus, work in

AI learning techniques and data mining will automatically enhance any educational decision support tool by providing increasingly rich and dynamic information in manner that will be closer and closer to real time.

The development of a prototype for a teacher decision support tool is one small step toward an ideal scenario. In the long term, research in this area should strive to create software that can integrate with a variety of data sources and learning environment to leverage student activity occurring in both digital and real world environments.

Ubiquitous learning requires constant monitoring which can observe students in as many learning situations as possible to give teachers a unique insight into the process of learning. Given the human nature of learning itself there are many leaps, bounds, and pitfalls that can take knowledge acquisition in a vast range of different paths. Being able to observe and evaluate this process is likewise a highly human activity but it needs to be supported by effective technical tools that can combine and contrast large information sources to unite human experts and expert systems in an environment that makes learning support as ubiquitous as the learning itself.

Chapter VI

Conclusion

Through the development of the project there has been a focus on identifying the key features needed to support students who no longer learn within the contexts of traditional classrooms. Students now have the ability to use real world and digital learning objects to draw upon a rich variety of resources that can make education more effective.

Unfortunately, this also leaves them without the benefits of traditional supports provided by teachers who can act as sources of content knowledge but more importantly have developed expert skills in understanding the learning process. While automated mechanisms have managed to provide some “teacher-like” assistance they have not attained a level sufficient to replace the function of a teacher. It also seems unlikely that this will occur in the near future due to the complexity of the learning process. In response this project has proposed a mechanism for allowing teachers to observe and learn about their classes in order to find opportunities to interact with students. This was expressed as a decision support tool in the form of a dashboard which has been designed to display complex data in a manner that should make it easy to use and understand. In this way, a teacher could easily interact with the tool to investigate student, group, and class needs in a highly flexible fashion. The prototype decision support tool took this one step further by proposing a recommendation mechanism to give advice to teachers and make them aware of potential learning support options. This was also implemented to demonstrate an ability to capture teacher responses so that a fully formed system could learn from the choices made by expert teachers. In this manner, the prototyped solution has illustrated the functionality of an end to end system which consumed data mining inputs, displayed them in a visual format, and provided suggestions for rectifying educational problems. While only representing an introductory step, this mechanism shows good promise for providing a way to reconnect teachers and students in a ubiquitous learning environment. This research also presents a mechanism that could be highly complementary to any distributed learning environment including ones that are highly automated. Even in AI situations the ability to provide a teaching expert with access to real time information provides an avenue for systems improvement and interpersonal interaction which would otherwise be absence. In this way teachers are enabled within distributed learning environment to provide learning supports in a similar way to those found in face-to-face learning environments. This represents a key maturation point in ubiquitous learning environments as they evolve to develop mechanisms for rich learning supports.

Future work

The prototype clearly represents a starting point requiring future work to create a fully formed solution. From a practical software development standpoint, it will be necessary to engage a group of suitable teachers to begin the feedback cycle so that a tool can be

developed to address specific education decision making needs. From a research perspective, there are a number of gaps that remain, which will need to be investigated to provide robust solution. In terms of information data mining, research will be required to refine methodologies for extracting the patterns necessary to display in the dashboard tool. This includes developing appropriate algorithms and heuristics as well as methods for aggregating data sets. A dashboard solution has to be able to draw on “large data” sets in order to achieve a level of certainty that teachers can trust. Visualizations will also need to be refined to ensure that they are easily understood and can apply to the context of each teacher. This focuses on the need for a customizable interface which could be formatted to suite different requirements. Finally, the recommendation component will also need to be enhanced to create a mechanism for machine learning which could allow the system develop and grow over time based on the interactions and decisions teachers make. In essence the system should also be a student of the teachers using it.

Works Cited

- The Apache Software Foundation*. (2012, May 14). Retrieved from Welcome to the Apache Software Foundation: <http://www.apache.org/>
- Google Developers*. (2013, 04 3). Retrieved from Google Charts: <https://developers.google.com/chart/>
- Graph Visualization Library With jQuery - Arbor*. (2013). Retrieved February 9, 2013, from jQuery Script.net: <http://www.jqueryscript.net/chart-graph/Graph-Visualization-Library-With-jQuery.html>
- iGoogle*. (2013, 05 15). Retrieved from iGoogle: <http://www.google.ca/ig>
- Introduction to Using Chart Tools*. (2013). Retrieved February 9, 2013, from Google Developers: <https://developers.google.com/chart/interactive/docs/>
- Ahmed, S. F., Chang, M., & Kinshuk. (2012). Recommend computer studies courses for students taken based on supported mobile learning modes. *IEEE International Conference on Technology Enhanced Education (ICTEE)*, (pp. 1-5). Kerala.
- Ali, L., Hatala, M., Gašević, D., & Jovanovic, J. (2012). A qualitative evaluation of evolution of a learning analytics tool. *Computers & Education*, 470-489.
- Arnold, K. E., Tanes, Z., & King, A. S. (2010). Administrative perspectives of data-mining software Signals: Promoting student success and retention. *The Journal of Academic Administration in Higher Education*, 29-39.
- Bargel, B. A., Schrock, J., & Roller, W. (2012). Using Learning Maps for Visualization of Adaptive Learning Path Components. *International Journal of Computer Information Systems and Industrial Management Applications*, 228-235.
- Bresfelean, V., Ghisoiu, N., Lacurezeanu, R., & Sitar-Taut, D.-A. (2009). Towards the development of decision support in academic environments. *Proceedings of the ITI 2009 31st International Conference on Information Technology Interfaces*, (pp. 343-348). Cavtat / Dubrovnik.
- Brтка, E., Radosav, D., & Brтка, V. (2010). Application of data mining techniques to regulated learning system. *8th International Symposium on Intelligent Systems and Informatics (SISY)*, (pp. 145-148). Subotica.
- Brusilovsky, P., Sosnovsky, S., & Shcherbinina, O. (2005). User Modeling in a Distributed E-Learning Architecture. *10th International User Modeling Conference*, (pp. 24-29). Edinburgh.

- Chanchary, F., Haque, I., & Khalid, S. (2008). Web Usage Mining to Evaluate the Transfer of Learning in a Web-Based Learning Environment. *First International Workshop on Knowledge Discovery and Data Mining (WKDD)*, (pp. 249-253). Adelaide.
- Chou, C.-Y., Chan, T.-W., & Lin, C.-J. (2002, Nov/Dec). An approach of implementing general learning companions for problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 14(6), 1376-1386.
- Correa, C., Crnovrsanin, T., & Ma, K.-L. (2012). Visual Reasoning about Social Networks Using Centrality Sensitivity. *IEEE Transactions on Visualization and Computer Graphics*, 106-120.
- Dai, Q., Bao, H., Yu, Z., & Wang, M. (2009). Design of Real-Time Monitoring and Visual Management System Based on RFID for Engineering Training. *WRI World Congress on Computer Science and Information Engineering*, (pp. 578-581). Los Angeles.
- eksith. (2013, 01 12). *Building a jQuery UI Dashboard*. Retrieved from This Page Intentionally Left Ugly: <http://eksith.wordpress.com/2012/01/02/jquery-ui-dashboard/>
- Friedler, S., Tan, Y., Peer, N., & Shneiderman, B. (2008). Enabling teachers to explore grade patterns to identify individual needs and promote fairer student assessment. *Computers & Education*, 1467-1485.
- Fry, B. (2008). *Visualizing Data*. O'Reilly Media.
- García, E., Romero, C., Ventura, S., & de Castro, C. (2011). A collaborative educational association rule mining tool. *The Internet and Higher Education*, 77-88.
- Gomez-Aguilar, D., Conde-Gonzalez, M., Theron, R., & Garcia-Penalvo, F. (2011). Reveling the Evolution of Semantic Content through Visual Analysis. *11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 450-454). Athens.
- Graf, S., & Kinshuk. (2006). An Approach for Detecting Learning Styles in Learning Management Systems. *Sixth International Conference on Advanced Learning Technologies*, (pp. 161-163). Kerkrade.
- Graf, S., & Kinshuk. (2009). Advanced Adaptivity in Learning Management Systems by Considering Learning Styles. *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT)*, (pp. 235-238). Milano.

- Graf, S., Kinshuk, & Liu, T.-C. (2008). Identifying Learning Styles in Learning Management Systems by Using Indications from Students' Behaviour. *Eighth IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 482-486). Santander.
- Hatzilygeroudis, I., Giannoulis, C., & Koutsojannis, C. (2005). Combining expert systems and adaptive hypermedia technologies in a Web based educational system. *ICALT Fifth IEEE International Conference on Advanced Learning Technologies*, (pp. 249-253). Kaohsiung.
- Hu, W., Almansoori, A., Kannan, P., Azarm, S., & Wang, Z. (2012). Corporate dashboards for integrated business and engineering decisions in oil refineries: An agent-based approach. *Decision Support Systems*, 729-741.
- Hwang, G.-J., Chen, C.-Y., Tsai, P.-S., & Tsai, C.-C. (2011). An expert system for improving web-based problem-solving ability of students. *Expert Systems with Applications*, 38(7), 8664-8672.
- Kent, M., Carr, B., Husted, R., & Pop, R. (2011). Learning web analytics: A tool for strategic communication. *Public Relations Review*, 536-543.
- Kopanakis, I., & Theodoulidis, B. (2003). Visual data mining modeling techniques for the visualization of mining outcomes. *Journal of Visual Languages & Computing*, 543-589.
- Latham, A., Crockett, K., McLean, D., Edmonds, B., & O'Shea, K. (2010). Oscar: An intelligent conversational agent tutor to estimate learning styles. *IEEE International Conference on Fuzzy Systems (FUZZ)*, (pp. 1-8). Barcelona.
- Lavrac, N., Bohanec, M., Pur, A., Cestnik, B., Debeljak, M., & Kobler, A. (2007). Data mining and visualization for decision support and modeling of public health-care resources. *Journal of Biomedical Informatics*, 438-447.
- Li, Y. (2009). A Remodeling Method of Automatic Learning Process Based on LMS in E-Learning. *International Conference on Web Information Systems and Mining (WISM)*, (pp. 565-569). Shanghai.
- Limoanco, T., & Sison, R. (2002). Use of learner agents as student modeling system and learning companion. *Proceeding of the International Conference on Computers in Education*, (pp. 1331-1332). Auckland.
- Macfadyen, L., & Dawson, S. (2010). Mining LMS data to develop an “early warning system” for educators: A proof of concept. *Computers & Education*, 588-599.

- Mahendrawathi, E., Pranantha, D., & Utomo, J. (2010). Development of dashboard for hospital logistics management. *IEEE Conference on Open Systems (ICOS)*, (pp. 86-90). Kuala Lumpur.
- Martin, E., & Di Bernardo, V. (2008). Enterprise Dashboard Tools for Management of Share-Use University Laboratory. *17th Biennial University/Government/Industry Micro/Nano Symposium (UGIM)*, (pp. 24-27). Louisville.
- Mitrovic, A. (2006). Large-scale deployment of three intelligent web-based database tutors. *28th International Conference on Information Technology Interfaces*, (pp. 135-140). Zagreb.
- Mitrovic, A., Martin, B., & Suraweera, P. (2007). Intelligent Tutors for All: The Constraint-Based Approach. *IEEE Intelligent Systems*, 22(4), 38-45.
- Nelaturu, S., Kambham, R., Karna, N., Parupalli, R., & Mandula, K. (2010). Building intelligent campus environment utilizing ubiquitous learning. *2010 International Conference on Technology for Education (T4E)*, (pp. 230-231). Mumbai.
- Ninomiya, T., Nakayama, K., Shimizu, M., Anma, F., & Okamoto, T. (2007). The Design of e-Learning Environment Oriented for Personalized Adaptability. *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 728-729). Niigata.
- Padolsey, J. (2013, 01 13). *How to Mimic the iGoogle Interface*. Retrieved from net tuts: <http://net.tutsplus.com/tutorials/javascript-ajax/inettuts/>
- Puntheeranurak, S., & Tsuji, H. (2005). Mining Web logs for a personalized recommender system. *3rd International Conference on Research and Education (ITRE)*, (pp. 445-448). Hsinchu.
- Rodriguez-Cerezo, D., Gomez-Albarran, M., & Sierra, J.-L. (2011). Supporting Self-Regulate Learning in Technical Domains with Repositories of Learning Objects and Recommender Systems. *11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 613-614). Athens.
- Romero, C., Ventura, S., Zafra, A., & de Bra, P. (2009). Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. *Computers & Education*, 828-840.
- Sachin, R., & Vijay, M. (2012). A Survey and Future Vision of Data Mining in Educational Field. *Second International Conference on Advanced Computing & Communication Technologies (ACCT)*, (pp. 96-100). Rohtak.

- Tang, N., Chen, Z., Lin, T., & Cui, J. (2012). Research and design of applying association rule in Course Management System. *2nd International Conference on Communications and Networks (CECNet)* (pp. 3394-3398). Yichang: IEEE.
- Tchoua, R., Klasky, S., Podhorszki, N., Grimm, B., Khan, A., Santos, E., . . . Vouk, M. (2010). Collaborative monitoring and analysis for simulation scientists. *International Symposium on Collaborative Technologies and Systems (CTS)*, (pp. 235-244). Chicago.
- Uthayakumar, C., & Sarukesi, K. (2011). An adaptive e-Learning system using knowledge management. *3rd International Conference on Trendz in Information Sciences and Computing (TISC)*, (pp. 70-74). Chennai.
- Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., & Duval, E. (2007). Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *IEEE Transactions on Learning Technologies*, 6(1), 1-20.
- Virvou, M., & Chrysafiadi, K. (2006). A Web-based Educational Application for Teaching of Programming: Student Modeling via Stereotypes. *Sixth International Conference on Advanced Learning Technologies*, (pp. 117-119). Washington DC.
- Wu, Q., Miao, C., & Shen, Z. (2012). A curious learning companion in Virtual Learning Environment. *2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, (pp. 1-8). Brisbane.
- Wu, S., Chang, A., Chang, M., Liu, T.-C., & Heh, J.-S. (2008). Identifying Personalized Context-Aware Knowledge Structure for Individual User in Ubiquitous Learning Environment. *Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education*, (pp. 95-99). Beijing.
- Yang, G., Kinshuk, & Graf, S. (2010). A Practical Student Model for a Location-Aware and Context-Sensitive Personalized Adaptive Learning System. *Proceedings of the IEEE Technology for Education Conference* (pp. 130-133). Bombay: IEEE Computer Society.
- Yigitbasioglu, O., & Velcu, O. (2012). A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 41-59.
- Zhang, L., Kaschek, R., & Kinshuk. (2005). Developing a knowledge management support system for teaching database normalization. *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 344-358). Kaohsiung.

Zhou, M., & Xu, Y. (2010). Injecting Pedagogical Constraints into Sequential Learning Pattern Mining. *10th International Conference on Advanced Learning Technologies (ICALT)*, (pp. 380-381). Sousse.

Zorrilla, M., & García-Saiz, D. (2012). A service oriented architecture to provide data mining services for non-expert data miners. *Decision Support Systems*.

Index I – PALD

/*

Database: PALD

Schema: <All Schemas>

*/

```
CREATE TABLE adaptiveevents (  
    eventid      int(11) AUTO_INCREMENT NOT NULL,  
    event        varchar(255) NULL,  
    eventtype    varchar(15) NULL,  
    eventdate    date NULL,  
    eventtime    time NULL,  
    componentid int(11) NULL,  
    userid       int(11) NULL,  
    PRIMARY KEY(eventid)  
);  
  
CREATE TABLE address (  
    id            int(11) AUTO_INCREMENT NOT NULL,  
    profile_id    int(11) NOT NULL,  
    location_id   int(11) NULL,  
    primary_addr  varchar(5) NULL DEFAULT 'yes',  
    address       varchar(70) NULL,  
    city          varchar(50) NULL,  
    province     varchar(50) NULL,  
    country       varchar(50) NULL,  
    postcode     varchar(10) NULL,  
    latitude      decimal(10,6) NULL,  
    longitude     decimal(10,6) NULL,  
    altitude      decimal(10,6) NULL,  
    PRIMARY KEY(id)  
);  
  
CREATE TABLE cell_network (  
    id            int(11) AUTO_INCREMENT NOT NULL,  
    user_id       int(11) NOT NULL,  
    phone_no      varchar(20) NULL,  
    cell_id       varchar(50) NULL,  
    network_code  varchar(50) NULL,  
    lac           varchar(50) NULL,  
    sig           int(11) NULL,  
    imei          varchar(100) NULL,  
    device_type_id int(11) NULL,  
    update_date   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    PRIMARY KEY(id)  
);
```

```

CREATE TABLE cell_network_history (
    id            int(11) AUTO_INCREMENT NOT NULL,
    user_id       int(11) NOT NULL,
    cell_network_id int(11) NULL,
    phone_no      varchar(20) NULL,
    cell_id        varchar(50) NULL,
    network_code   varchar(50) NULL,
    lac           varchar(50) NULL,
    sig           int(11) NULL,
    imei          varchar(100) NULL,
    device_type_id int(11) NULL,
    update_date   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE component (
    id            int(11) AUTO_INCREMENT NOT NULL,
    name          varchar(100) NULL DEFAULT 'Component',
    synccount     int(11) NULL DEFAULT '0',
    PRIMARY KEY(id)
);
CREATE TABLE component_usage (
    id            int(11) AUTO_INCREMENT NOT NULL,
    user_id       int(11) NOT NULL,
    component_id  int(11) NOT NULL,
    course_id     int(11) NULL,
    time_start    timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    time_end      timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
    PRIMARY KEY(id)
);
CREATE TABLE courses (
    id            int(11) AUTO_INCREMENT NOT NULL,
    code          varchar(10) NULL,
    name          varchar(10) NULL,
    description    varchar(255) NULL,
    active        varchar(10) NULL,
    atriskcapacity int(11) NULL,
    stallhcap     int(11) NULL,
    stallmcap     int(11) NULL,
    learnissuecap int(11) NULL,
    atriskuse     int(11) NULL,
    stallhuse     int(11) NULL,
    stallmuse     int(11) NULL,
    learnissueuse int(11) NULL,
    activedate    date NULL,
    enddate       date NULL,
    updatedate    date NULL,

```

```

        PRIMARY KEY(id)
    );
CREATE TABLE decision_ref (
    id          int(11) AUTO_INCREMENT NOT NULL,
    name        varchar(20) NULL,
    modified_by varchar(20) NOT NULL DEFAULT 'admin',
    modified_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE device_type (
    id          int(11) AUTO_INCREMENT NOT NULL,
    type        varchar(20) NULL,
    name        varchar(50) NULL,
    manufacturer varchar(100) NULL,
    os          varchar(100) NULL,
    level       varchar(20) NULL,
    PRIMARY KEY(id)
);
CREATE TABLE domainknowledge (
    domainid    int(11) AUTO_INCREMENT NOT NULL,
    domainname  varchar(255) NULL,
    keyword     varchar(255) NULL,
    createddate date NULL,
    createdtime time NULL,
    PRIMARY KEY(domainid)
);
/* New table added for dashboard*/
CREATE TABLE groups (
    id          int(11) AUTO_INCREMENT NOT NULL,
    student     int(11) NOT NULL,
    groupid     int(11) NOT NULL,
    groupname   varchar(25) NULL,
    course      varchar(25) NULL,
    active      varchar(5) NULL,
    description varchar(100) NULL,
    type        varchar(10) NULL,
    update_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    create_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
/* New table added for dashboard*/
CREATE TABLE help_requests (
    id          int(11) AUTO_INCREMENT NOT NULL,
    student     int(11) NOT NULL,

```

```

        active      varchar(5) NULL,
        description varchar(45) NULL,
        type        varchar(10) NULL,
        update_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
        create_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
        PRIMARY KEY(id)
);
CREATE TABLE interest (
    id            int(11) AUTO_INCREMENT NOT NULL,
    student       int(11) NOT NULL,
    concept       int(11) NULL,
    degree        int(11) NULL,
    create_time   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE knowledge (
    knowledgeid   int(11) AUTO_INCREMENT NOT NULL,
    knowledgetitle varchar(255) NULL,
    domainid      int(11) NULL,
    keyword       varchar(255) NULL,
    tag           varchar(255) NULL,
    createddate   date NULL,
    createdtime   time NULL,
    parentknowledge int(11) NULL,
    PRIMARY KEY(knowledgeid)
);
CREATE TABLE knowledge_level (
    id            int(11) AUTO_INCREMENT NOT NULL,
    student       int(11) NOT NULL,
    concept       int(11) NULL,
    degree        int(11) NULL,
    create_time   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    knowledgeid   int(11) NULL,
    satisfaction   int(11) NULL,
    PRIMARY KEY(id)
);
CREATE TABLE knowledgeleveltype (
    kleveltypeid int(11) AUTO_INCREMENT NOT NULL,
    klevelname   varchar(255) NULL,
    granularity   int(4) NULL,
    description   varchar(255) NULL,
    PRIMARY KEY(kleveltypeid)
);
CREATE TABLE lcmobile_version (

```

```

        id            int(11) AUTO_INCREMENT NOT NULL,
        version       varchar(50) NOT NULL,
        updatedate    timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
        description   varchar(255) NULL,
        PRIMARY KEY(id)
    );
CREATE TABLE learning_styles (
    id            int(11) AUTO_INCREMENT NOT NULL,
    student       int(11) NOT NULL,
    actref        int(11) NULL,
    learning_style_type  int(11) NOT NULL,
    senint        int(11) NULL,
    visver        int(11) NULL,
    seqglo        int(11) NULL,
    create_time   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE learning_styles_history (
    id            int(11) AUTO_INCREMENT NOT NULL,
    student       int(11) NOT NULL,
    no_updated    int(11) NOT NULL DEFAULT '0',
    actref        int(11) NULL,
    senint        int(11) NULL,
    visver        int(11) NULL,
    seqglo        int(11) NULL,
    update_time   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE learning_styles_ils (
    id            int(11) AUTO_INCREMENT NOT NULL,
    student       int(11) NOT NULL,
    actref        int(11) NULL,
    senint        int(11) NULL,
    visver        int(11) NULL,
    seqglo        int(11) NULL,
    q1            int(11) NOT NULL DEFAULT '0',
    q2            int(11) NOT NULL DEFAULT '0',
    q3            int(11) NOT NULL DEFAULT '0',
    q4            int(11) NOT NULL DEFAULT '0',
    q5            int(11) NOT NULL DEFAULT '0',
    q6            int(11) NOT NULL DEFAULT '0',
    q7            int(11) NOT NULL DEFAULT '0',
    q8            int(11) NOT NULL DEFAULT '0',
    q9            int(11) NOT NULL DEFAULT '0',
    q10           int(11) NOT NULL DEFAULT '0',
    q11           int(11) NOT NULL DEFAULT '0',

```

```

q12          int(11) NOT NULL DEFAULT '0',
q13          int(11) NOT NULL DEFAULT '0',
q14          int(11) NOT NULL DEFAULT '0',
q15          int(11) NOT NULL DEFAULT '0',
q16          int(11) NOT NULL DEFAULT '0',
q17          int(11) NOT NULL DEFAULT '0',
q18          int(11) NOT NULL DEFAULT '0',
q19          int(11) NOT NULL DEFAULT '0',
q20          int(11) NOT NULL DEFAULT '0',
q21          int(11) NOT NULL DEFAULT '0',
q22          int(11) NOT NULL DEFAULT '0',
q23          int(11) NOT NULL DEFAULT '0',
q24          int(11) NOT NULL DEFAULT '0',
q25          int(11) NOT NULL DEFAULT '0',
q26          int(11) NOT NULL DEFAULT '0',
q27          int(11) NOT NULL DEFAULT '0',
q28          int(11) NOT NULL DEFAULT '0',
q29          int(11) NOT NULL DEFAULT '0',
q30          int(11) NOT NULL DEFAULT '0',
q31          int(11) NOT NULL DEFAULT '0',
q32          int(11) NOT NULL DEFAULT '0',
q33          int(11) NOT NULL DEFAULT '0',
q34          int(11) NOT NULL DEFAULT '0',
q35          int(11) NOT NULL DEFAULT '0',
q36          int(11) NOT NULL DEFAULT '0',
q37          int(11) NOT NULL DEFAULT '0',
q38          int(11) NOT NULL DEFAULT '0',
q39          int(11) NOT NULL DEFAULT '0',
q40          int(11) NOT NULL DEFAULT '0',
q41          int(11) NOT NULL DEFAULT '0',
q42          int(11) NOT NULL DEFAULT '0',
q43          int(11) NOT NULL DEFAULT '0',
q44          int(11) NOT NULL DEFAULT '0',
create_time  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(id)
);
/* New table added for dashboard - could be removed*/
CREATE TABLE learning_style_types (
  id          int(11) AUTO_INCREMENT NOT NULL,
  Type       varchar(25) NOT NULL,
  Description varchar(45) NULL,
  create_time timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY(id)
);
/* New table added for dashboard*/

```

```

CREATE TABLE learningobjectpreference (
    loprid          int(11) AUTO_INCREMENT NOT NULL,
    courseid       varchar(10) NULL,
    studentid      int(11) NULL,
    reflection      int(11) NULL,
    self           int(11) NULL,
    discussion      int(11) NULL,
    reading         int(11) NULL,
    animations      int(11) NULL,
    exercise        int(11) NULL,
    example         int(11) NULL,
    reallife        int(11) NULL,
    createdate      date NULL,
    updatedate      date NULL,
    PRIMARY KEY(loprid)
);
CREATE TABLE location (
    id              int(11) AUTO_INCREMENT NOT NULL,
    user_id         int(11) NOT NULL,
    name           varchar(60) NULL,
    postcode       varchar(100) NULL,
    latitude        decimal(10,6) NULL,
    longitude       decimal(10,6) NULL,
    altitude        decimal(10,6) NULL,
    context_id     int(11) NULL,
    cell_network_id int(11) NULL,
    mobile_id       int(11) NULL,
    sessionID      varchar(25) NULL,
    speed           int(11) NULL,
    direction       int(11) NULL,
    distance        int(11) NULL,
    gpsTime         timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
    locationMethod  varchar(100) NULL,
    accuracy        int(11) NULL,
    isLocationValid varchar(5) NULL,
    extraInfo       varchar(255) NULL,
    update_date    timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE location_history (
    id              int(11) AUTO_INCREMENT NOT NULL,
    location_id     int(11) NULL,
    user_id         int(11) NOT NULL,
    name           varchar(60) NULL,
    postcode       varchar(100) NULL,
    latitude        decimal(10,6) NULL,

```



```

longitude    decimal(10,6) NULL,
altitude     decimal(10,6) NULL,
context_id   int(11) NULL,
cell_network_id int(11) NULL,
mobile_id    int(11) NULL,
sessionID    varchar(25) NULL,
speed        int(11) NULL,
direction    int(11) NULL,
distance     int(11) NULL,
gpsTime      timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
locationMethod varchar(100) NULL,
accuracy     int(11) NULL,
isLocationValid varchar(5) NULL,
extraInfo    varchar(255) NULL,
update_date  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(id)
);
CREATE TABLE login (
  id          int(11) AUTO_INCREMENT NOT NULL,
  user_id     int(11) NOT NULL,
  active      char(5) NULL DEFAULT 'N',
  logout_time date NULL,
  login_time  timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  hostip      varchar(255) NULL,
  useragent   varchar(255) NULL,
  PRIMARY KEY(id)
);
CREATE TABLE mimecontent (
  mimecontentid int(11) AUTO_INCREMENT NOT NULL,
  mimeid        int(11) NULL,
  bcontent      longblob NULL,
  tcontent      longtext NULL,
  filepath      varchar(255) NULL,
  tag           varchar(255) NULL,
  PRIMARY KEY(mimecontentid)
);
CREATE TABLE mimedata (
  mimeid       int(11) AUTO_INCREMENT NOT NULL,
  mimetype     varchar(255) NULL,
  createddate  date NULL,
  createdtime  time NULL,
  PRIMARY KEY(mimeid)
);
CREATE TABLE mobile_device (
  id          int(11) AUTO_INCREMENT NOT NULL,

```

```

        registration_id int(11) NOT NULL,
        phone_no        varchar(20) NULL,
        imei             varchar(100) NULL,
        device_type_id   int(11) NULL,
        update_date      timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY(id)
    );
CREATE TABLE progress (
    id                int(11) AUTO_INCREMENT NOT NULL,
    student           int(11) NOT NULL,
    lola_id           int(11) NULL,
    type_id           int(11) NULL,
    component_id      int(11) NULL,
    concept_id        int(11) NULL,
    update_time       timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    questionid        int(11) NULL,
    PRIMARY KEY(id)
);
/* New table added for dashboard*/
CREATE TABLE recommendations (
    id                int(11) AUTO_INCREMENT NOT NULL,
    user_id           int(11) NULL,
    action            varchar(100) NULL,
    accepted          varchar(5) NULL,
    create_date       timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    update_date       timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    outcome           varchar(100) NULL,
    groupid           int(11) NULL,
    PRIMARY KEY(id)
);
/* Table modified for dashboard*/
CREATE TABLE registration (
    id                int(11) AUTO_INCREMENT NOT NULL,
    user_id           int(11) NOT NULL,
    student_id        int(11) NOT NULL,
    course            varchar(10) NULL,
    instructor_id     int(11) NOT NULL,
    tutor_id          int(11) NULL,
    update_date       timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    start_date        timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    duration          int(11) not null,
    PRIMARY KEY(id)
);
CREATE TABLE role (
    id                int(11) AUTO_INCREMENT NOT NULL,
    name              varchar(100) NULL DEFAULT 'Guest',

```

```

        shortname    varchar(32) NULL DEFAULT 'guest',
        description  varchar(255) NULL,
        sortorder   int(11) NULL,
        PRIMARY KEY(id)
    );
CREATE TABLE sc_know (
    id              int(11) AUTO_INCREMENT NOT NULL,
    student_one    int(11) NOT NULL,
    student_two    int(11) NOT NULL,
    time_stamp     timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE sc_learnt (
    id              int(11) AUTO_INCREMENT NOT NULL,
    student_one    int(11) NOT NULL,
    student_two    int(11) NOT NULL,
    time_stamp     timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
CREATE TABLE sc_willingness (
    id              int(11) AUTO_INCREMENT NOT NULL,
    student        int(11) NOT NULL,
    peer           int(11) NOT NULL,
    group_id       int(11) NULL,
    decision_id    int(11) NOT NULL,
    request_time   timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY(id)
);
/* New table added for dashboard*/
CREATE TABLE studentartifact (
    id              int(11) AUTO_INCREMENT NOT NULL,
    type           varchar(100) NULL,
    description     varchar(100) NULL,
    link           varchar(100) NULL,
    studentid      int(11) NULL,
    PRIMARY KEY(id)
);
/* New table added for dashboard*/
CREATE TABLE student_risk (
    id              int(11) AUTO_INCREMENT NOT NULL,
    student        int(11) NOT NULL,
    active         varchar(5) NOT NULL,
    Type           varchar(25) NOT NULL,
    Description     varchar(45) NULL,
    update_time    timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

```

```

        create_time    timestamp DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
        PRIMARY KEY(id)
);
CREATE TABLE studentinfo (
    studentid    int(11) AUTO_INCREMENT NOT NULL,
    userid       int(11) NULL,
    firstname    varchar(100) NULL,
    lastname     varchar(100) NULL,
    email        varchar(100) NULL,
    homephone    varchar(20) NULL,
    workphone    varchar(20) NULL,
    language     varchar(30) NULL,
    gender       varchar(6) NULL,
    createdtime  time NULL,
    createddate  date NULL,
    modifiedtime time NULL,
    modifieddate date NULL,
    modifiedby   int(11) NULL,
    PRIMARY KEY(studentid)
);
CREATE TABLE user (
    id           int(11) AUTO_INCREMENT NOT NULL,
    loginname    varchar(100) NULL DEFAULT 'lcmobile',
    password     varchar(32) NULL DEFAULT 'lcmobile',
    deleted      int(11) NULL,
    mobilephone  varchar(20) NULL,
    imei         varchar(100) NULL,
    created_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    description  varchar(255) NULL,
    roleid       int(11) NULL,
    PRIMARY KEY(id)
);
CREATE TABLE user_profile (
    id           int(11) AUTO_INCREMENT NOT NULL,
    user_id      int(11) NOT NULL,
    firstname    varchar(100) NOT NULL,
    lastname     varchar(100) NOT NULL,
    studentid    varchar(100) NULL,
    email        varchar(100) NULL,
    home_phone   varchar(20) NULL,
    work_phone   varchar(20) NULL,
    language     varchar(30) NULL,
    major        varchar(100) NULL,
    graduate     varchar(10) NULL,
    au_student_id varchar(20) NULL,

```

```

        created_date          timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
        modified_date         timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
        modified_by           varchar(20) NULL,
        gender                 varchar(6) NULL,
        current_study_year     varchar(20) NULL,
        PRIMARY KEY(id)
);

CREATE INDEX profile_id
    ON address(profile_id);
CREATE INDEX cell_history_user_id
    ON cell_network_history(user_id);
CREATE INDEX device_type_id
    ON cell_network_history(device_type_id);
CREATE INDEX cell_user_id
    ON cell_network(user_id);
CREATE INDEX device_type_id
    ON cell_network(device_type_id);
CREATE INDEX component_id
    ON component_usage(component_id);
CREATE INDEX user_id
    ON component_usage(user_id);
CREATE INDEX student
    ON interest(student);
CREATE INDEX knowledgeid
    ON knowledge_level(knowledgeid);
CREATE INDEX student
    ON knowledge_level(student);
CREATE INDEX domainid
    ON knowledge(domainid);
CREATE INDEX parentknowledge
    ON knowledge(parentknowledge);
CREATE INDEX student
    ON learning_styles_history(student);
CREATE INDEX student
    ON learning_styles_ils(student);
CREATE INDEX student
    ON learning_styles(student);
CREATE INDEX location_history_user_id
    ON location_history(user_id);
CREATE INDEX location_user_id
    ON location(user_id);
CREATE INDEX user_id
    ON login(user_id);
CREATE INDEX mimeid

```

```

        ON mimecontent(mimeid);
CREATE INDEX mobile_reg_id
        ON mobile_device(registration_id);
CREATE INDEX component_id
        ON progress(component_id);
CREATE INDEX student
        ON progress(student);
CREATE INDEX reg_user_id
        ON registration(user_id);
CREATE INDEX student_one
        ON sc_know(student_one);
CREATE INDEX student_two
        ON sc_know(student_two);
CREATE INDEX student_one
        ON sc_learnt(student_one);
CREATE INDEX student_two
        ON sc_learnt(student_two);
CREATE INDEX decision_id
        ON sc_willingness(decision_id);
CREATE INDEX user_id
        ON user_profile(user_id);
CREATE INDEX roleid
        ON user(roleid);
ALTER TABLE address
        ADD CONSTRAINT address_ibfk_1
        FOREIGN KEY(profile_id)
        REFERENCES user_profile(id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT;
ALTER TABLE cell_network_history
        ADD CONSTRAINT cell_network_history_ibfk_2
        FOREIGN KEY(device_type_id)
        REFERENCES device_type(id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT;
ALTER TABLE cell_network_history
        ADD CONSTRAINT cell_network_history_ibfk_1
        FOREIGN KEY(user_id)
        REFERENCES user(id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT;
ALTER TABLE cell_network
        ADD CONSTRAINT cell_network_ibfk_2
        FOREIGN KEY(device_type_id)
        REFERENCES device_type(id)
        ON DELETE RESTRICT

```

```

        ON UPDATE RESTRICT;
ALTER TABLE cell_network
    ADD CONSTRAINT cell_network_ibfk_1
    FOREIGN KEY(user_id)
    REFERENCES user(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE component_usage
    ADD CONSTRAINT component_usage_ibfk_2
    FOREIGN KEY(component_id)
    REFERENCES component(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE component_usage
    ADD CONSTRAINT component_usage_ibfk_1
    FOREIGN KEY(user_id)
    REFERENCES user(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE interest
    ADD CONSTRAINT interest_ibfk_1
    FOREIGN KEY(student)
    REFERENCES user(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT
;
ALTER TABLE knowledge_level
    ADD CONSTRAINT knowledge_level_ibfk_2
    FOREIGN KEY(knowledgeid)
    REFERENCES knowledge_level(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE knowledge_level
    ADD CONSTRAINT knowledge_level_ibfk_1
    FOREIGN KEY(student)
    REFERENCES user(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE learning_styles_history
    ADD CONSTRAINT learning_styles_history_ibfk_2
    FOREIGN KEY(student)
    REFERENCES user(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE learning_styles_history
    ADD CONSTRAINT learning_styles_history_ibfk_1

```

```

FOREIGN KEY(student)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE learning_styles_ils
ADD CONSTRAINT learning_styles_ils_ibfk_1
FOREIGN KEY(student)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE learning_styles
ADD CONSTRAINT learning_styles_ibfk_1
FOREIGN KEY(student)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE location_history
ADD CONSTRAINT location_history_ibfk_1
FOREIGN KEY(user_id)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE location
ADD CONSTRAINT location_ibfk_1
FOREIGN KEY(user_id)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE login
ADD CONSTRAINT login_ibfk_1
FOREIGN KEY(user_id)
REFERENCES user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE mobile_device
ADD CONSTRAINT mobile_device_ibfk_1
FOREIGN KEY(registration_id)
REFERENCES registration(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
ALTER TABLE progress
ADD CONSTRAINT progress_ibfk_2
FOREIGN KEY(component_id)
REFERENCES component(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;

```



```

ALTER TABLE progress
  ADD CONSTRAINT progress_ibfk_1
  FOREIGN KEY(student)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE registration
  ADD CONSTRAINT registration_ibfk_1
  FOREIGN KEY(user_id)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE sc_know
  ADD CONSTRAINT sc_know_ibfk_2
  FOREIGN KEY(student_two)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE sc_know
  ADD CONSTRAINT sc_know_ibfk_1
  FOREIGN KEY(student_one)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE sc_learnt
  ADD CONSTRAINT sc_learnt_ibfk_2
  FOREIGN KEY(student_two)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE sc_learnt
  ADD CONSTRAINT sc_learnt_ibfk_1
  FOREIGN KEY(student_one)
  REFERENCES user(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE sc_willingness
  ADD CONSTRAINT sc_willingness_ibfk_1
  FOREIGN KEY(decision_id)
  REFERENCES decision_ref(id)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT;
ALTER TABLE user_profile
  ADD CONSTRAINT user_profile_ibfk_1
  FOREIGN KEY(user_id)
  REFERENCES user(id)

```

```
        ON DELETE RESTRICT
        ON UPDATE RESTRICT;
ALTER TABLE user
    ADD CONSTRAINT user_ibfk_3
    FOREIGN KEY(roleid)
    REFERENCES role(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE user
    ADD CONSTRAINT user_ibfk_2
    FOREIGN KEY(roleid)
    REFERENCES role(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
ALTER TABLE user
    ADD CONSTRAINT user_ibfk_1
    FOREIGN KEY(roleid)
    REFERENCES role(id)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT;
```