Migration of a Learning Object Repository – an Evaluation of the eNOSHA System and its Context Independency

Authors: 1) Peter Mozelius[1]
2) Amit Roy[2]
3) Enosha Hettiarachchi[3]
[4)] Jarkko Suhonen[2]

Affiliation:
1) Stockholm University, Sweden
2) University of Eastern Finland, Finland
3) University of Colombo, School of Computing, Sri Lanka

Abstract:
The idea of storing and sharing digital content for e-learning has been widely discussed. At the University of Colombo, School of Computing (UCSC) a lot of learning objects have been developed during the last decade. Even in a well structured university environment it is not possible to have a rational control of content for hundreds of different e-learning courses without an appropriate system that provides searching and a metadata markup of the learning objects. This paper will analyze the eNOSHA Learning Object Repository (LOR) that was designed and developed at UCSC in Colombo Sri Lanka during 2008 – 2009. Even if this was a software project for a LOR based on the UCSC needs the ambition has always been to built the eNOSHA system as context independent as possible. Local adaptation of the system should be done from within the system without any changes in the source code. In our analysis we will assess if this works when the system is migrated to a university in another part of the world with a different culture and language situation. We will also scrutinize the installation process and test its user-friendliness. What could be improved in the eNOSHA system and is there a need for an extended version to provide context independency? Data for the analysis has been gathered by semi-structured interviews and system testing at the University of Eastern Finland. Based on the findings we find that the currently used version 1.6 of the eNOSHA system needs to be updated. There are no big crucial problems found, but if the system should live up to the original objectives on flexibility and user-friendliness the software would need some amendment for multi-language support. Some security issues were found and the system must be easier to install in the future. We have also found a need for some kind of integrated start-up instructions. A general code revision and a security test of the

system would also improve the quality of the eNOSHA learning object repository. Though open source products often are created with a specific target audience in mind, it would be of interest to test this product with users outside the academic world.

Keywords: Learning object repository, Context independency, E-learning, Open source, Knowledge sharing, eNOSHA

1 Introduction

Distance education has definitely moved online in the 21st century and the modern technology enhanced learning is based on the use of digital learning objects. In the design and use of e-learning content, it is important to have a teaching and learning environment that connects all stakeholders in the process. (Hettiarachchi et al., 2010) A LOR has more and more become a required product for storing digital material in large scale virtual learning environments. Repositories facilitate the collaboration between the different stakeholders in a content development process and encourage reuse of digital assets. (Mozelius and Hettiarachchi, 2010)

1.1 Learning Object Repositories

At the same time as schools and universities have developed a lot of digital content for their online programmes the amount of free and reusable learning objects available on the Internet has constantly increased. With this accelerated growth of digital content the need for LORs to sift the information has increased as well (McGreal, 2008). A LOR could be defined as a system for storage of digital learning material with functionality for searching, reusing and sharing content. LORs can be categorized into:

1. **Content repositories**: All content stored on accessible servers
2. **Linking repositories**: Portals with links to content provided by others
3. **Hybrid repositories**: A combination of 1 and 2. (McGreal, 2008)

Type 2 and type 3 repositories will always be difficult to maintain since external links always will have a time consuming need for checking and updating. An external type 1 repository where all the storing is done from within the system is a more reliable way to provide quality and appropriate search functions. Important features for a LOR are:

- **Visibility –** all working roles need a common forum where learning object can be stored, found an updated

- **Searchability** – the system should be easy to search for relevant content with minimal effort

- **Version handling** – users should be able to handle different versions of objects

- **User-friendliness** - the system must be user-friendly, both when it comes to adding objects to the repository and in the searching existing learning objects

- **Collaborative environment** - the system should work as a collaborative environment where users should be able to work together to make the process more efficient

- **Reusability** - this functionality is indirectly achieved through the features for visibility, searchability and flexibility

(Hettiarachchi et al., 2010)

A university-based and institutional repository could be seen as a set of services that a university offers to the staff for the management and dissemination of digital content created by the institution and its community members. (Lynch, 2003)

1.2 Context dependency - independency

Context is the setting in which events occur. (Neovius et al., 2006) Open source solutions are created with a view of providing general utility through an application in similar contexts. Different organizations though they are similar in nature, still maintain their own uniqueness in addition to the common context. Freedom from these contextual differences i.e. context independence is an important element in determining the possible success of any application. For any application to be context-sensitive, these uniqueness features along with the context commonalities, have to be respected and the ability of the application to adapt to the local contexts provides an advantage to the users and managers of the application. Context-awareness introduces a variety of software engineering challenges. (Henricksen, and Indulska, 2006) In case of eNOSHA, it is a learning object repository and its possible contexts are learning organizations. A physical world simulation of eNOSHA is that of a library in which a

department or departments store their learning objects. The general features of storing these objects with some meta-information to make the retrieval process easier are common. When a software application is used for such process, the operational efficiencies are one of the chief aims of the organization.

1.3 Unicode

Computers deal with digital information much more efficiently than humans as they deal with numbers only. They store letters and other characters by assigning them a number. An interesting computer science problem faced in mid-1980s was that computer based systems aimed to store and present information in formats understandable to human beings and human beings have several different languages. The multi-byte encoding schemes available at that time were inadequate and at times, conflicted with each other. The risk of data corruption was always there. (Unicode, 2011)

Single language or bi-lingual character sets were suitable to fulfill the requirements of the pre-Internet age. In a world where World Wide Web connects several international users to a system, these character-sets may pose a hindrance to global acceptance of the application. Unicode is unique, unified, universal encoding (Becker, 1988) and helps solving these problems related not only to support new languages but also leaves scope to accommodate future cultural changes for use of new symbols like the Indian rupee sign (₹). One of the critical aspects of a global application today is the ability to support multi-byte characters.

1.4 Aim

The aim of this study is to test, evaluate and discuss the context independency of the eNOSHA learning object repository and suggest how the system could be updated and made more user-friendly for a global use in different kind of organizations.

2 Extended Background

2.1 The BIT and eBIT Programmes

Towards the end of the last century, there was an increasing demand for IT graduates in the Sri Lankan industry and the traditional university system could not educate enough students. To address this need, a new tailor made Bachelor of Information Technology (BIT) was designed and launched by the University of Colombo, School of Computing

(UCSC). (Wikramanayake, G. N. et al. , 2007). The BIT programme was constructed as an external programme without any lectures or teaching sessions at the University in Colombo. Course syllabi were a UCSC design and they were also responsible for the curriculum and the digital content. But for lectures and teaching sessions the students had to visit private facilitating places and teaching institutes (Mozelius et al., 2011).

All information about the programme such as registration, curriculum, examination, etc. has been published on http://www.bit.lk. Initially it was an information website but later it was reconstructed as a student portal, where students could register for the program as well as retrieve information. The BIT programme, which also includes some foreign students, has now, updated to the eBIT version, become one of the most popular IT degrees on the Sri Lankan island. (Mozelius et al., 2011)

eBIT – The Online Version of BIT

To develop online courses with a user-centric collaborative learning pedagogy the BIT curriculum was revised. A new Virtual Learning Environment (VLE) based Moodle LMS was installed and customized. This localized e-Learning PLATFORM was introduced as *Vidupiyasa* (Sinhala term meaning: *Interface for Learning and Knowledge*). Interactive learning objects were developed according to the international e-Learning standard SCORM for all the 28 courses in the 6 semester BIT programme.

(Should we mention the pass rates in numerals before and after online services?) The new eBIT programme was created to increase the BIT pass rate by providing online interactive learning materials and assessment. After one year, the pass rate of first year (Diploma of IT) semesters was thrice as high as the previous rate and at the same time many students obtained good grades due to the digital online content. (Mozelius et al., 2011)

2.2 The ViSCoS Programme (should we interchange 2.2 and 2.3 for the continuation of eNOSHA first and then introducing ViSCoS?)

ViSCoS (Virtual Studies of Computer Science) is an e-learing programmefor first year University level Computer Science studies, called Basic studies in Computer Science. The programme is run by School of Computing, University of Eastern Finland (UEF). Originally the programme was targeted to high school students in sparsely populated

Eastern Finland, but later it has become available to anyone interested in studying university level Computer Science studies via the Continuning Education Center at UEF. The ViSCoS programme consists of seven online courses, which provide students knowledge and skills in three main areas:

(1) Preliminaries of information and communication technology (ICT)
(2) Basics of programming with Python and Java
(3) Introduction to computer science and software development

In addition, the programme also offers individual courses belonging to the Intermediate studies in Computer Science. The Basic studies have been implemented both in Finnish and English. The ViScoS programme have been running for over ten years. The structure and the study content of the programme has been evolving during this time, which have also increased the amount of online learning materials. In summary, it was estimated that in year 2011 there will be course materials for about 15 courses. In addition, one course can have several versions of the course materials. Therefore, there was a clear need to install and use a content repository system in order to store the course materials for the ViSCoS studies. The arrangement was settled by Dr Jarkko Suhonen (*ViSCoS group*) at UEF and Peter Mozelius (eNOSHA development team) during staff exchange meetings in Joensuu.

2.3 The eNOSHA Story

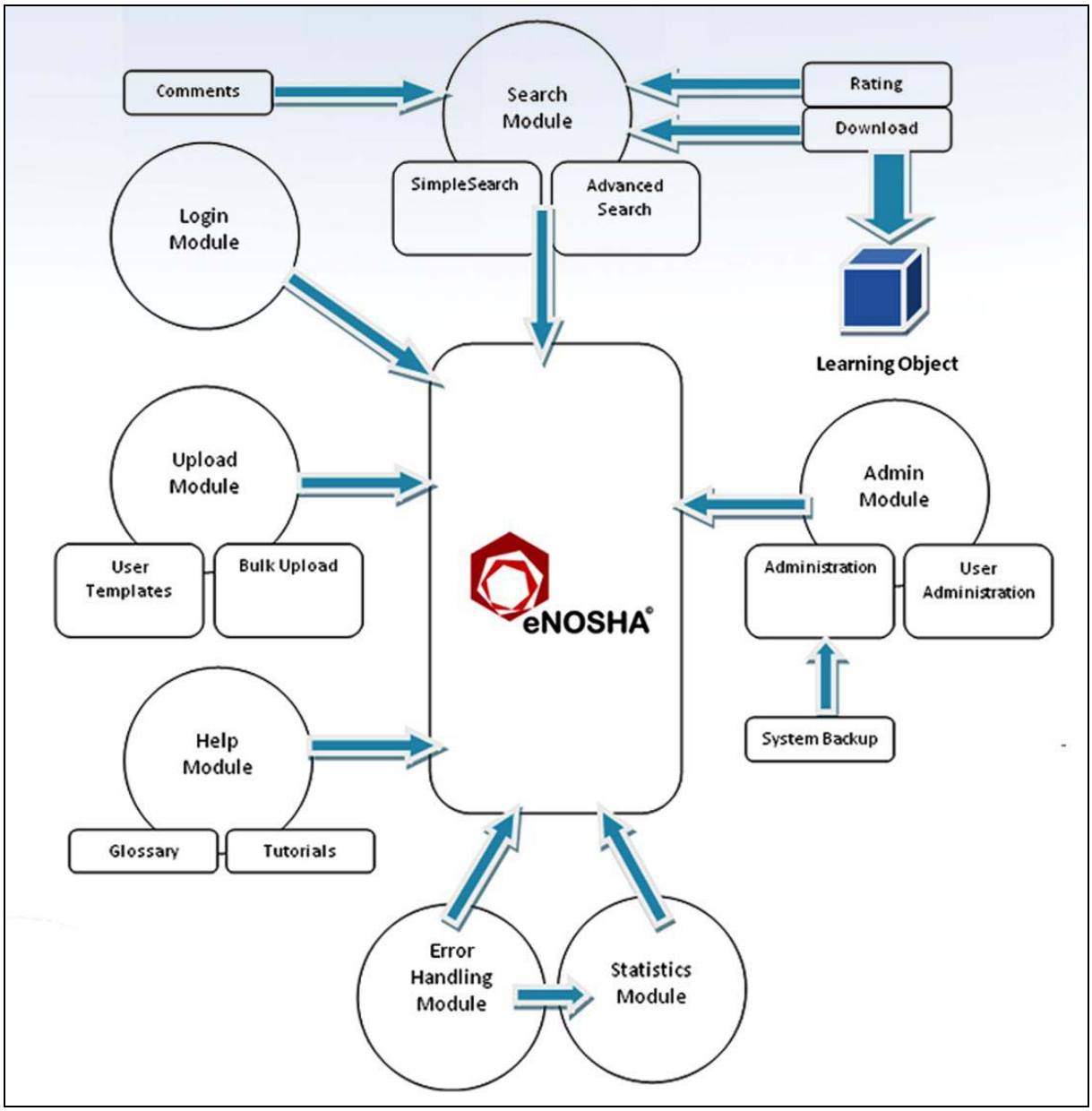
The e-learning Center at the UCSC has always created and published a large amount of learning material for its internal and external degree programs. In addition to this the staff of the e-Learning Center has developed a huge volume of reusable content. All these learning objects have been stored with backups on UCSC servers, but what have been missing during a long period are features for mark-up and searching of eLearning content. In this context, a method for these learning materials as well as the developed content to be shared among the staff itself, or certain material to be made publicly available was needed.

In the very beginning, the basic design idea was to reuse or develop a LOR that should be open, flexible and enough user-friendly to serve all stakeholders involved in the process of content development at the UCSC eLearning Centre. The most important

groups to consider at the UCSC eLearning Centre are Subject Matter Experts (SMEs), Instructional Designers (IDs) and Content Developers (CDs). In the UCSC *Conveyor Belt Model* for digital content development (Mozelius and Hatakka, 2009), it is important to have a flexible software system that will serve the different needs for the universities different working roles. Therefore as a result it was decided to develop a Free and Open Source Learning Object Repository (LOR) named eNOSHA (Neutral Object Storage with a Holistic Approach) which can be easily adapted to any organization which is in need of a LOR with minimal or no modifications.

The design of the eNOSHA system was started in December 2008. The first stage in the development was the design of a metadata set (Hatakka and Mozelius, 2009) for the system which has been revised after user evaluations. At the beginning, functionalities of the system were tested using prototypes. In addition to that Usability tests were done by using scenarios and the systems were modified according to the feedbacks obtained. After completion of the development, unit-tests were conducted in both Sri Lanka and Sweden to test the technical functionality of the system.

The eNOSHA system consists of several modules which also includes the most important parts of a LOR such as search and upload modules. In addition to that system consists of modules such as user management, administration, help, error handling and statistics. The Search module is divided into two parts, Search and Advanced Search and the Upload module consist of functionalities to upload content based on four aggregation levels - atoms, collection of atoms, course modules and full courses (Hettiarachchi et al., 2010). A sketch of the eNOSHA architecture can be like the image below:



Through these modules eNOSHA supports several functionalities such as simple and advanced search, administration functionalities, commenting content, user templates, bulk uploading, tag clouds, ranking of modules, editing of metadata, versioning, help facilities for novice users, error handling module for administrators and statistics (Hettiarachchi et al., 2010). In the version 1.6 of the eNOSHA system the learning object repository is integrated with the Moodle 1.9 virtual learning platform. The system has a Moodle module for importing learning objects directly from existing eNOSHA repositories. Learning objects stored in eNOSHA systems can be found from a search feature in the Moodle system. In May 2011 the eNOSHA system was installed at the UEF by Antii Rantaeskola and in the future the repository will be used on the UEF:s ViSCoS Programme.

3 Methodology

Testing of the installation process and the basic functionality was done by Antii Rantaeskola at the UEF in May 2011. Amit Roy has during 2011 constructed and executed test cases on context dependency of the eNOSHA software system. Uploading and searching tests with content in the Chinese language has been done by Amit Roy in collaboration with Chinese colleague, Qinpei Zhao, at the UEF. Semi-structured interviews have been conducted by Peter Mozelius and Amit Roy. Another evaluation of the software and its functionality has been done at the Department of Computer and Systems Sciences at Stockholm University, Sweden during the autumn of 2011. The evaluation Sweden with minor adaptations for a Swedish context has mainly been conducted by Peter Mozelius and Isuru Balasooriya.

4 Findings and Discussions

Every migration has its own problems and we have probably not found all of them yet. Two minor errors in the installation script were very time consuming in the installation process at UEF in Joensuu. We suspect that this might have been the main obstacle that has scared away several potential eNOSHA users earlier. Persons that do not have knowledge about PHP, SQL and the UNIX operating system might just close down and quit the use of the system very early without even making a full installation. The two discovered errors are:

1. A missing comma in a SQL query line 166 in the file **enosha.php**
2. Backslashes (\) instead of slashes (/) in the file **config.php**

As discovered early in the history of software development, the smaller the syntax errors the harder they are to find. The illustrative example of *Mariner 1*, where NASAs first mission to Venus failed by the substitution of a period for a comma in the programming code, must be classified as a *software engineering folklore* (Roberts, 1987). But the actual problem remains even in the case of the Mariner 1 crash, since the reason for the spacecraft accident was the substitution of the mathematical character *overbar* for a hyphen in the input to Mariner 1's software system. (Mariner 1, 2011)(Do we really need the highlighted section?) Like in high-tech spacecraft programs in the 1960's it is a single character that can make drastic semantic difference today in a programming language like PHP.

The problem with the backslashes in the config.php file is an example of context dependency that should not exist in a system that has been designed for platform independency. To have backslashes in the file paths is the syntax for the Windows operating system (which is the

environment where the system has been developed). But for the UNIX operating system, that is a very common on university servers, file paths must instead be delimited by slashes.

As in rocket science, security issues are important and the eNOSHA login procedure should be much more secure if the login was done using a *https connection* instead of the current *http connection*. Another, even more serious, security issue is that all files and directories are given reading and executing permissions in the installation process. University servers are multi-user systems and at the University of Eastern Finland all users given an account on the server could read the configuration files for the system. As an example, the password to the eNOSHA database can be retrieved with the current default settings. User settings could of course be modified later according to the security regulations at the actual university or organization, but it could be a good idea to restrict the privileges in Finland. The reason probably was lack of rights to the database.

After the installation was completed, it was easy to add new users but there were problems in removing users from the system. The current *help video* is long (duration) and could possibly work better divided into several shorter videos aligned to the relevant help sections. We got the same feedback from the testing in Sweden where test users from outside the developing team asked for a short introduction video and an extension of the built-in help glossary .

Despite continental differences, there are similarities between UEF/ViSCoS and UCSC/eBIT. Both Sri Lanka and Finland feel the shortage of Computer Science (CS) students and both the departments are directly related to Computer Science and both had started online courses in specializations related to CS and for their IT specific courses only. So both the contexts have precisely matching requirements. However, both the organizations have their own unique characters as well. While there are three roles (Subject Matter Experts, Instructional Designers and Content Developer) at UCSC, at UEF, all the required tasks are generally completed by one single person only. Therefore, it is not required in case of UEF to have all the three roles. At UEF, different roles with different privileges are not required at all.

eNOSHA passed the tests for language independency. Files with Chinese content with Chinese file names could be uploaded, searched and downloaded. The only file type that caused any problems was plain text-files created and saved with *Windows Notepad*. In this case characters outside the 256 bit ASCII/ANSI tables were not correctly handled. A problem

that probably may get repeated by software tools without Unicode-support. All the menus, buttons, information etc in the graphical user interface are in English and they are, in the current version, not possible to translate to any other language.

5 Conclusions

Based on the findings, we think that the currently used version 1.6 of the eNOSHA system needs to be updated. Two small errors in the files config.php and enosha.php must be seen as irritating obstacles in the installation process and maybe the reason why not more of the, earlier interested organizations, actually use the system. A detailed security test of the system combined with a general code revision would also improve the quality of the learning object repository.

Considering context independency the system performs well and maybe that the differences between a Computer Science programme in Sri Lanka and one in Finland are not too many. At a university English is often used but if the system should be used in other educational organizations, where English is not the main language, it would be an improvement to integrate functionality to present items in the graphical user interface in local languages. Even if open source products often are created with a specific target audience in mind, it would be of interest to test this product with users outside the academic world that are without former knowledge about open source products in general and learning object repositories in specific .

The eNOSHA system can be installed, integrated and used in an organization where the teaching is based on Moodle ver 1.9. But there is a need for further testing and updating if the LOR should be integrated with newer versions of Moodle or with other types of virtual learning environments. Finally we find the discovered technical problems and the lack of user-friendliness for some system features to be the most serious issues that need attention and immediate updating.

6 Future Work

Everything in this study is done at two universities where the employees have good skills in computer science as well as in the English language. It would be interesting to test this system at an educational organization in a rural area where English is not the working language. The work with an updated and more globally adapted version 1.7 of the system has started with

funding from the Department of Computer and Systems Sciences at the Stockholm University. For a global audience we also find it important to integrate the system with Moodle 2.0 even if UCSC, UEF and Stockholm University still uses Moodle 1.9.

7 Acknowledgement

We like to thank Antii Rantaeskola for his patience and careful testing of the system and to *NordForsk* for the support of Nordic-Asian research collaboration.

References:

Becker, J. (1988) "Unicode 88" http://www.unicode.org/history/unicode88.pdf (Retrieved 24/07/2011)

Hatakka, M., and Mozelius, P. (2009) "A model for a Learning Object Repository Metadata Set – A Case Study at UCSC, Sri Lanka", eASIA, Colombo, Sri Lanka 2009

Henricksen, K. and Indulska, J., (2006), "Developing Context-Aware Pervasive Computing Applications: Models and Approach", Pervasive and Mobile Computing February 2006: (Volume 2, Issue 1) p. 37-64

Hettiarachchi E., Mozelius P., Hewagamage K. P., Hatakka M., Balasooriya I., Karunarathne D. (2010) " eNOSHA - The Design and Development of a Learning Object Repository", ICTer journal Vol 3, 2010

Lynch, C. (2003) "Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age." ARL, 226, February 2003, 1-7. http://www.arl.org/

Mariner 1 – Wikipedia. (2011)) "Mariner 1" http://en.wikipedia.org/wiki/Mariner_1 (Retrieved 02/06/2011)

McGreal, R. (2008) *"A Typology of Learning Object Repositories"* , Springer Berlin Heidelberg, 2008

Mozelius, P., and Hatakka, M. (2009*) "Conveyor Belt Production of Course Material – a CaseStudy in Sri Lanka"* , ECEL conference, Bari, Italy, 2009.

Mozelius, P., and Hettiarachchi, E. (2010*) "eNOSHA, a Free, Open and Flexible Learning Object Repository"*, eIndia, Hyderabad, India 2010.

Mozelius P., Hewagamage K.P., Hansson H. (2011*) "Towards e-learning for all in Sri Lanka – progress and problems in some selected Sri Lankan 21st century initiatives"* , ODEL Conference, Cambridge, England 2011

Nagamori, M. and Sugimoto, S. (2006) *"A Metadata Schema Registry as a Tool to Enhance Metadata Interoperability"* TCDL Bulletin, 3. 2006

Neovius, M., Sere, K., Yan, L. and Satpathy, M., (2006) A Formal Model of Context-Awareness and Context-Dependency, Fourth IEEE International Conference on Software Engineering and Formal Methods, (SEFM 2006)

Roberts, E. (1987) *"Mariner 1"*, The Risk digest, Volume 5: Issue 66, 1987 http://catless.ncl.ac.uk/Risks/5.66.html#subj1 (Retrieved 02/06/2011)

Suhonen J., and Sutinen E. (2007*) "Learning Computer Science over the Web: The VisSCoS Odyssey"* Idea Group Inc, 2007

Unicode, *"What is Unicode?"* http://www.unicode.org/standard/WhatIsUnicode.html (Retrieved 24/07/2011)

Wikramanayake, G. N., Hewagamage, K. P., Gamage, G. I. & Weerasinghe, A. R. (2007). *"Asia ebit @ucsc: Implementing the paradigm shift from teaching to learning through e-learning framework."*, 25th National Information Technology Conference. Colombo, Sri Lanka.