

EKS: OPEN-SOURCE WEB-BASED DISTRIBUTED LEARNING TOOL

Wasana NGAOGATE

University Needs

As common requirements of a computer assisted learning system, we want a tool that is easy to use –for the teacher to organize the teaching material, and for students to learn individually or in a group. Not only for the current teaching material but also for further developments. LMS (Learning Management System) aims to develop a system that presents the teaching material in the correct form and sequence, helps students to get the proper content that meets their needs, and suggests for further steps in the learning processes.¹

In a university, there are some faculties that provide courses serving students from different faculties. For example, a mathematics department provides mathematics courses for engineering students, computer science students, or physics students. Students from different faculties have been registered as users in the department, whereas their personal information has already been recorded in their original department.

Repeated information still occurs as long as there is no cooperation between faculties, or there is no central department such as a computer centre that takes the responsibility of maintaining students' information. Moreover, students may want to access some teaching material across the servers in different faculties. For example, engineering students want to process MatLab software on the science faculty server, computer science students want to study in-depth in electronics at physics department.

Therefore, providing the teaching material, which works on a single computer system, is not flexible in practice. A university needs infrastructure that allows servers to communicate with each other in order to share, or exchange students' information and teaching material.

The Pilot Prototype: EKS Tool

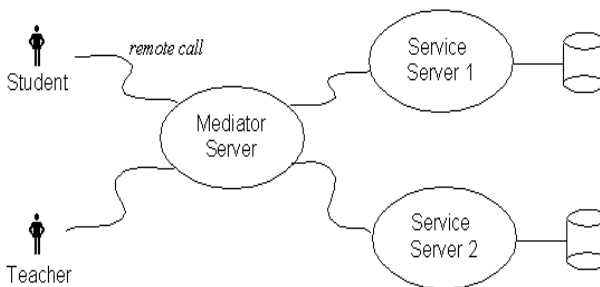
As the teaching material can be kept on heterogeneous servers, each topic in the teaching material can refer to different servers in the system and has clearly self-contained objective. A single server might be easier to manage but it involves higher risks in terms of data security. Therefore, the pilot prototype is designed to be a distributed system.

Each server in the system is independent and has either the same or different function from others. Servers are classified into two types:

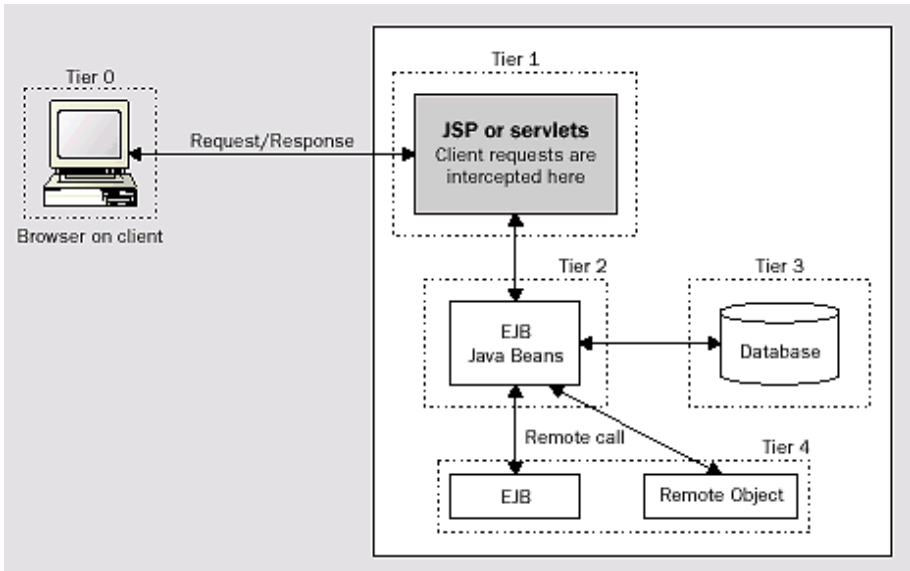
1. *Mediator server* that acts as a portal for all users and passes a service function number and parameters to service servers;
2. *Service servers* that serve as the service functions specified; they are further classified into two types:
 - 2.1 *Content servers* that contain teaching material and also maintain the changing of the content and keep records of events in learning process;
 - 2.2 *Query server* that processes records of events and produces useful reports for the teacher.

Because of the transport layer of the Java RMI architecture, each server can communicate to others by calling the service name that has been registered as a remote object. Once the remote object is called, it means it is invoked to serve its client. In other words, all servers in the system communicate to each other through the middle mediator or a middleware.

System Architecture



Initially, all servers have registered their services to the distributed system. Then the user interface for the teacher and students calls the mediator server remotely. Each service server has its own local database. It is a 3-tier architecture implemented by JSP, Java Beans, and MySQL as shown on the figure below.²

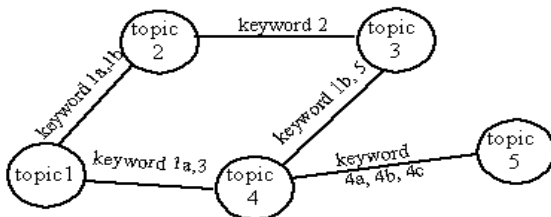


The Teaching Material

The EKS tool aims to provide a web-based tool that encourages and supports the teacher to improve and enlarge the teaching material. Therefore, we start our examination with the way teaching material is organized.

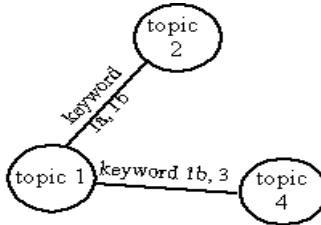
How to organize the material?

In order to organize the teaching material, if one topic is considered as the smallest unit of teaching material, one lecture might compose of one topic or several topics and each topic has clearly self-contained objective. There might be one keyword or more that link two topics together. A keyword is part of the topic.



Each topic semantically links to one or many topics by keyword. This structure allows students to select the topic in two ways:

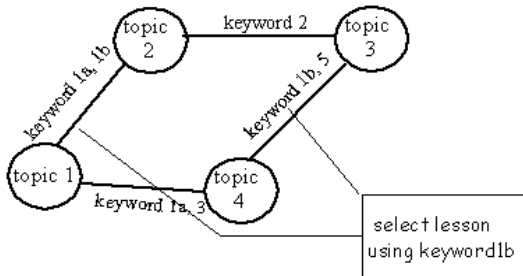
- 1) Selecting the topic using topic name



Topic 1 is related to topic 2 and topic 4.

If the student selects topic 1, the system will also suggest other topics which are related to topic 1; these are topic 2 and topic 4.

- 2) Selecting the topic using keyword



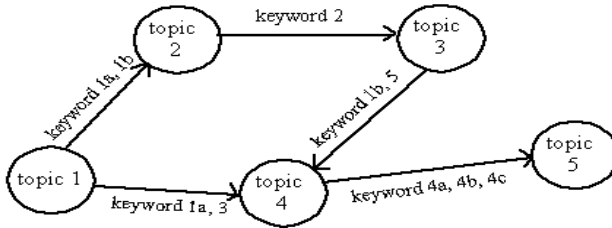
If the student selects lesson using keyword1b, as shown on the diagram, he or she will get choices: topic 1, topic 2, topic 3, and topic 4 because all of them are related to keyword1b.

The diagram can be organized according to the relationship of topics and can be reorganized again and again after the teacher has made a decision using reports, generated by the system.

Learning Dependency

Even though students are allowed to pick up any topic independently, there are some topics that have to be learned before moving to others, because they will be applied in the later topics.

A learning flow is defined by a transition diagram as in the example below.






Therefore, students are suggested what topic(s) they should study before studying the topic they want. Also, the teacher is warned if the student tries to create a dependency loop.

Screen Shots

Web-based Interface

EKS contains teaching material of module CS123 Functional Programming taught by Dr. Steve Matthews, Department of Computer Science, The University of Warwick, in 2002. A module is a course that consists of several subjects. A subject consists of several topics, which are self-contained learning units. At the main page, there are menus for the teacher and the students.

EKS

-  [Teacher Main Menu](#)
-  [Student Main Menu](#)
-  [About EKS Project](#)

Teacher Menus

There are three main menus for the teacher.

- [Module Management](#)
- [Keyword Management](#)
- [View Reports](#)

- [show list of subjects](#)
- [show list of topics](#)
- [show list of prerequisites](#)
- [show list of exercises](#)
- [create objective](#)
- [delete topic](#)
- [create new subject](#)
- [create new topic](#)
- [define prerequisite](#)
- [create exercises](#)
- [update topic](#)

Module Management. The teacher starts to prepare the teaching material from creating the subjects and topics.

How many subjects would you like to create?

(max 20 subjects)

No. Subject Name

0

1

2

3

For each subject, the teacher creates topics.

Please choose subject which you would like to create topics

- Basic concepts
- Higher Order Fuctions
- Recursion
- Types

How many topics would you like to create?:

Success, You have insert these records

Module: CS123

Subjects:

Basic concepts

Recursion

Types

Higher Order Fuctions

Please create topic(s) for the subject(s) [here](#)

The teacher inputs the URL of the topic web page.

No.	Topic Name	Topic URL (do not put http://)
0	lecture1	www.dcs.warwick.ac.uk/~sgm/cs123/lec/1.pdf
1	lecture2	www.dcs.warwick.ac.uk/~sgm/cs123/lec/2.pdf
2	lecture3	www.dcs.warwick.ac.uk/~sgm/cs123/lec/3.pdf
3	lecture4	www.dcs.warwick.ac.uk/~sgm/cs123/lec/4.pdf

Success, You have insert these records

Module: CS123

Subject: Basic concepts

Topics:

lecture1

lecture2

lecture3

lecture4

Then the teacher defines prerequisite topic(s). There is one or more prerequisite topic(s) of one particular topic.

Please choose a topic name

- [CS123-Basic concepts-lecture1](#)
- [CS123-Basic concepts-lecture2](#)
- [CS123-Basic concepts-lecture3](#)
- [CS123-Basic concepts-lecture4](#)
- [CS123-Recursion-lecture10](#)
- [CS123-Recursion-lecture5](#)
- [CS123-Recursion-lecture6](#)
- [CS123-Recursion-lecture7](#)
- [CS123-Recursion-lecture8](#)
- [CS123-Recursion-lecture9](#)
- [CS123-Types-lecture11](#)
- [CS123-Types-lecture12](#)
- [CS123-Types-lecture13](#)
- [CS123-Types-lecture14](#)
- [CS123-Types-lecture15](#)
- [CS123-Types-lecture16](#)

submit

=>

Please choose prerequisites for CS123-Basic concepts-lecture1
One topic could have one or many prerequisites (default = none).

- CS123-Basic concepts-lecture1
- CS123-Basic concepts-lecture2
- CS123-Basic concepts-lecture3
- CS123-Basic concepts-lecture4
- CS123-Recursion-lecture10
- CS123-Recursion-lecture5
- CS123-Recursion-lecture6
- CS123-Recursion-lecture7
- CS123-Recursion-lecture8
- CS123-Recursion-lecture9
- CS123-Types-lecture11
- CS123-Types-lecture12
- CS123-Types-lecture13
- CS123-Types-lecture14
- CS123-Types-lecture15
- CS123-Types-lecture16

submit

If the teacher tries to make a loop of dependency, the system will give the message.

Error Catch22, please check the prerequisite of topics again

Keyword Management. The teacher creates a keyword and defines topic(s) that the keyword is embedded in.

Please Input Keyword (max 255 chars)

expressions

Please select topic(s) the keyword will be embedded in

- [CS123-Basic concepts-lecture1](#)
- [CS123-Basic concepts-lecture2](#)
- [CS123-Basic concepts-lecture3](#)
- [CS123-Basic concepts-lecture4](#)
- [CS123-Recursion-lecture10](#)
- [CS123-Recursion-lecture5](#)
- [CS123-Recursion-lecture6](#)
- [CS123-Recursion-lecture7](#)

Keyword, Topic

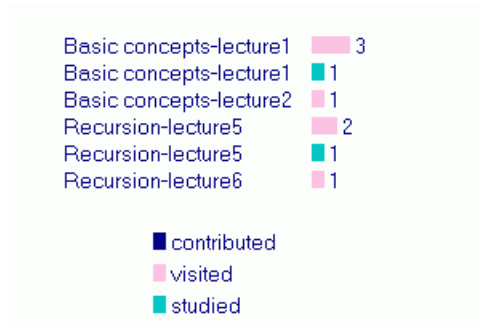
expressions, CS123-Basic concepts-lecture1
expressions, CS123-Basic concepts-lecture2
expressions, CS123-Basic concepts-lecture3
expressions, CS123-Basic concepts-lecture4
partitioning, CS123-Recursion-lecture10
partitioning, CS123-Recursion-lecture5
partitioning, CS123-Recursion-lecture6
partitioning, CS123-Recursion-lecture7
partitioning, CS123-Recursion-lecture8
partitioning, CS123-Recursion-lecture9

View Reports. In the pilot prototype, the system provides reports of access frequency of each topic, students' contribution in each topic and from each student.

[EKS Home](#) => [Teacher Main Menu](#) => [View Report: CS123](#)

- [📌 Access Frequency in Each Topic](#)
- [📌 Students' Contribution in Each Topic](#)
- [📌 Students' Contribution from Each Student](#)

After managing teaching material and allowing students to study online, the teacher can view reports provided by the system.



The teacher gives points to students' contribution.

usrId	date/time	contribution text	
student	2003-11-10 13:26:56	what is relationship between expression and type?	give points
student	2003-11-10 13:33:47	I do not understand the meaning of recursion	give points
wasana	2003-11-10 13:58:36	What is types?	give points

usrId: student

contribution: I do not understand the meaning of recursion

Please give point:

Student Menus

Students have to login to the system so that the system can keep track on what students have done and also that the system knows what machine students are using.

Log in

user id:


password:

[>> Register as a new user <<](#)


Student Main Menu

wasana

[Study](#)
[My Points](#)



Question of the Day
question in focus posted by the teacher



Annoucement
annoucement from module manager

Student Study Main Page. Students can visit the page, contribute, or do exercise. If they have no idea which topic should be picked up, they can input a keyword and search for topic(s) related to the keyword.

	CS123-Basic concepts-lecture1	contribute	do exercise	3 visited	1 studied	1 contributed
	CS123-Basic concepts-lecture2	contribute	do exercise	1 visited	0 studied	0 contributed
	CS123-Basic concepts-lecture3	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Basic concepts-lecture4	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Recursion-lecture10	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Recursion-lecture5	contribute	do exercise	2 visited	1 studied	0 contributed
	CS123-Recursion-lecture6	contribute	do exercise	1 visited	0 studied	0 contributed
	CS123-Recursion-lecture7	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Recursion-lecture8	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Recursion-lecture9	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture11	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture12	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture13	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture14	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture15	contribute	do exercise	0 visited	0 studied	0 contributed
	CS123-Types-lecture16	contribute	do exercise	0 visited	0 studied	0 contributed

Search by keyword

[Show List of Keywords](#)

Let's collect your points !!! [prize](#)
By Contributing to the topics OR Doing the exercises

Students look at the topic page which is composed of objective, prerequisite, and keyword(s) embedded in the topic. If they are interested to study this topic, they will click the 'study' button.

[Student Main Menu](#) => [Study](#) => Topic: CS123-Basic concepts-lecture2
wasana

- Objective
no objective defined
- You need to study this topic(s) before study CS123-Basic concepts-lecture2
CS123-Basic concepts-lecture1
- List of keyword(s) embedded in topic CS123-Basic concepts-lecture2
expressions

At the topic page, students' activity was recorded as “visited.” After clicking the 'study' button, the activity is “studied.” All activities are recorded by the system.

Students' Contribution. If students contribute to the system, they can check their points later after the teacher has checked their contribution and has given them points.

Please type in your contribution

what is relationship between expression and type?

• Thank you very much for your contribution.
You may check your points later from the main page

[Student Main Menu](#) => [My Points & Coins](#)
wasana

Let's collect more !!! [prize](#)

- Points: 0
- Silver Coins: 0
- Golden Coins: 0

The prize might be a bar of chocolate.

Conclusions and Further Work

EKS is a pilot prototype implemented by JSP, Java Beans, and MySQL in a distributed system that aims to demonstrate how teacher and students work together in order to improve teaching material. EKS is based on a distributed computing technology that makes the system flexible in the sense of adding more functions to the system by other developers because the EKS architecture allows that, for example, adding more service servers or even mediator servers.

The system allows students to study independently on their own pace and provides useful reports to the teacher so that he or she knows what has happened in the learning process and makes decision on how to adjust the teaching material according to students' needs.

The system needs to be further improved by adding more functions for both the teacher and students. Also, security should be further considered.

Notes:

- ¹ For basic definitions the reader may refer to the ADL section of the *International Relations and Security Network* website, <<http://www.isn.ethz.ch/elearning/adl/>> (7 May 2004).
- ² "The 'Dispatcher' Approach," Architecture Part 5, in Karl Avedal, et al., *Professional JSP: Using JavaServer Pages, Servlets, EJB, JNDI, JDBC, XML, XSLT, and WML* (Visualsoft UK, 2000), <<http://tutorials.beginners.co.uk/read/id/239>> (7 May 2004).

WASANA NGAOGATE is currently finishing her PhD study in Computer Science at the University of Warwick, UK. Ms Ngaogate started her career as a programmer at Bangkok Bank, Thailand. After two years she undertook studies for M.Sc. in computer science. After finishing the study, she worked as a lecturer at Ubon Ratchathani University, Thailand, for six years. Her research interests are in the areas of Computer Assisted Learning and Applications of Distributed Systems. *Mail address:* Wasana Ngaogate, Department of Computer Science, The University of Warwick, Coventry CV4 7AL, UK. *E-mail:* wngaogate@dcs.warwick.ac.uk.