Université de Montréal

Web2.0, Knowledge Sharing and Privacy in E-learning

par

Hicham Hage

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences en vue de l'obtention du grade de Doctorat en Informatique

Juillet, 2010

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée

Web2.0, Knowledge Sharing and Privacy in E-learning

présenté par:
Hicham Hage

a été évalué par un jury compose des personnes suivantes:

Louis Salvail, président-rapporteur
Esma Aïmeur, directrice de recherche
Claude Frasson, membre du jury
Julita Vassileva, examinatrice externe
Serge Dubuc, représentant du doyen

Resumé

Quand le E-learning a émergé il ya 20 ans, cela consistait simplement en un texte affiché sur un écran d'ordinateur, comme un livre. Avec les changements et les progrès dans la technologie, le E-learning a parcouru un long chemin, maintenant offrant un matériel éducatif personnalisé, interactif et riche en contenu. Aujourd'hui, le E-learning se transforme de nouveau. En effet, avec la prolifération des systèmes d'apprentissage électronique et des outils d'édition de contenu éducatif, ainsi que les normes établies, c'est devenu plus facile de partager et de réutiliser le contenu d'apprentissage. En outre, avec le passage à des méthodes d'enseignement centrées sur l'apprenant, en plus de l'effet des techniques et technologies Web2.0, les apprenants ne sont plus seulement les récipiendaires du contenu d'apprentissage, mais peuvent jouer un rôle plus actif dans l'enrichissement de ce contenu. Par ailleurs, avec la quantité d'informations que les systèmes E-learning peuvent accumuler sur les apprenants, et l'impact que cela peut avoir sur leur vie privée, des préoccupations sont soulevées afin de protéger la vie privée des apprenants. Au meilleur de nos connaissances, il n'existe pas de solutions existantes qui prennent en charge les différents problèmes soulevés par ces changements. Dans ce travail, nous abordons ces questions en présentant Cadmus, SHAREK, et le E-learning préservant la vie privée. Plus précisément, Cadmus est une plateforme web, conforme au standard IMS QTI, offrant un cadre et des outils adéquats pour permettre à des tuteurs de créer et partager des questions de tests et des examens. Plus précisément, Cadmus fournit des modules telles que *EQRS* (Exam Question Recommender System) pour aider les tuteurs à localiser des questions appropriées pour leur examens, *ICE* (Identification of Conflits in Exams) pour aider à résoudre les conflits entre les questions contenu dans un même examen, et le *Topic Tree*, conçu pour aider les tuteurs à mieux organiser leurs questions d'examen et à assurer facilement la couverture des différent sujets contenus dans les examens. D'autre part, *SHAREK* (Sharing REsources and Knowledge) fournit un cadre pour pouvoir profiter du meilleur des deux mondes : la solidité des systèmes E-learning et la flexibilité de PLE (Personal Learning Environment) tout en permettant aux apprenants d'enrichir le contenu d'apprentissage, et les aider à localiser nouvelles ressources d'apprentissage. Plus précisément, SHAREK combine un système recommandation multicritères, ainsi que des techniques et des technologies Web2.0, tels que le RSS et le web social, pour promouvoir de nouvelles ressources d'apprentissage et aider les apprenants à localiser du contenu adapté. Finalement, afin de répondre aux divers besoins de la vie privée dans le E-learning, nous proposons un cadre avec quatre niveaux de vie privée, ainsi que quatre niveaux de traçabilité. De plus, nous présentons *ACES* (Anonymous Credentials for E-learning Systems), un ensemble de protocoles, basés sur des techniques cryptographiques bien établies, afin d'aider les apprenants à atteindre leur niveau de vie privée désiré.

Mots-clés : E-learning, Web2.0, Vie Privée, Partage des connaissances

Abstract

E-learning emerged over 20 years ago, and was merely book like text displayed on a computer screen. With the changes and advances in technology, E-learning has come a long way, providing personal and interactive rich content. Today, E-learning is again going through major changes. Indeed, with the proliferation of E-learning systems and content authoring tools, as well as established standards, it has become easier to share and reuse learning content. Moreover, with the shift to learner centered education and the effect of Web2.0 techniques and technologies, learners are no longer just recipients of the learning content, but can play an active role into enriching such content. Additionally, with the amount of information E-learning systems can gather about learners, and the impact this has on their privacy, concerns are being raised in order to protect learners' privacy. Nonetheless, to the best of our knowledge, there is no existing work that supports the various challenges raised by these changes. In this work, we address these issues by presenting Cadmus, SHAREK, and privacy preserving E-learning. Specifically, Cadmus is an IMS QTI compliant web based assessment authoring tool, offering the proper framework and tools to enable tutors author and share questions and exams. In detail, Cadmus provides functionalities such as the EQRS (Exam Questions Recommender System) to help tutors locate suitable questions, ICE (Identification of Conflicts in Exams) to help resolve conflicts between questions within the same exam, and the topic tree, designed to help tutors better organize their exam questions and easily ensure the content coverage of their exams. On the other hand, SHAREK (Sharing REsources and Knowledge) provides the framework to take advantage of both the rigidity of E-learning systems and the flexibility of PLEs (Personal Learning Environment) while enabling learners to enrich the learning content, and helping them locate new learning resources. Specifically, SHAREK utilizes a multi-criteria content based recommender system, and combines Web2.0 technologies and techniques such as RSS and social web to promote new learning resources and help learners locate suitable content. Lastly, in order to address the various needs for privacy in E-learning, we propose a framework with four levels of privacy, and four levels of tracking, and we detail ACES (Anonymous Credentials for E-learning Systems), a set of protocols, based on well established cryptographic techniques, to help learners achieve their desired level of privacy.

Keywords : E-learning, Web2.0, Privacy, Knowledge sharing

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would first like to express my appreciation to my supervisor, Prof. Esma Aïmeur, for her guidance and support. Prof. Aïmeur, besides your intellectual and wise counsel, I cherish the most your personal qualities, your understanding and your kindness, especially during the difficult times I went through. I will hold in memory that I had an excellent research director.

A special thought goes to my wife, Mariette. Without your unwavering support and dedication, I would not have made it this far. I promise I will make up for all the nights and week-ends I spent away working late.

To my boys, Carl and Marc I dedicate this work. No Carl, you cannot draw on these pages, and no Marc, you cannot put these pages in your mouth.

I would like to extend my gratitude to the members of the jury for your time, and all the constructive comments to help me improve this work.

To everyone in the Heron lab, thank you for your help and feedback. Specifically, I would like to extend a special acknowledgment to Flavien Serge Mani Onana and Pierre Chalfoun. Flavien Serge, thank you for your help and guidance. I really appreciate the times when you selfishly took time, away from your work and family, to help me. Pierre, I will miss our coffee break together, and in particular, our discussions and the way you always challenged my ideas, pushing me to excel.

Chapter 1 : Introduction

When looking for the definition of E-Learning, one would come across different interpretations (LiNe-Zine, 2000). We summarize the definition of E-learning with the following statement: E-learning is the delivery and support of educational and training material using computers.

E-learning is an aspect of distant learning, where teaching materials are accessed through electronic media and where teachers and students can communicate electronically (email, chat rooms, forums, skype, ...).

E-learning emerged over 20 years ago, and consisted solely of text, like a book on a screen, and was ineffective and unpopular with learners. Today, E-learning has become richer with multimedia content and more interactive. With E-learning, education is shifting from being Tutor Centered, where the tutor is the center and has access to the resources, and becoming more Learner Centered, where the student is the center and has access to a multitude of resources (Figure 1).

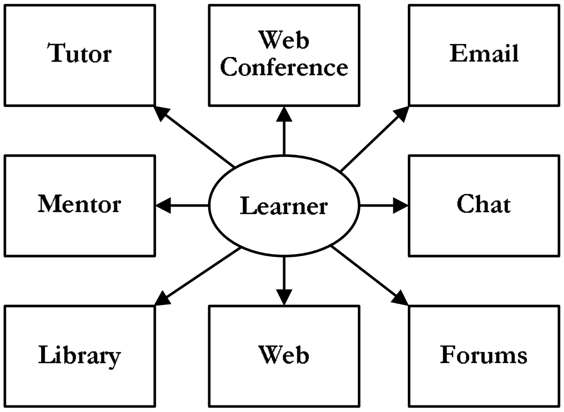


Figure 1: Learner Centered education

E-learning is very convenient and portable; the student is not bound by physical space since study material is accessible remotely. Furthermore, the student is not bound by time, since he can study at his own pace, and can have access to assistance through email or message boards. E-learning is also very flexible; students can skip material or chapters they already know. Students can choose instructor-led or self-study courses and can choose from a variety of learning tools that best fit their style. In addition, E-learning involves a great collaboration and interaction between students and tutors or specialist. Such collaboration is made easier by the online environment. For example, a student in Canada can have access to a specialist in Europe or Asia through email or assist to his lecture through a web conference. Despite the advancement in E-learning, many aspects still require further development; such aspects include *Knowledge Sharing* and *Learner Privacy*.

Knowledge sharing, from the tutors' perspective, is important since it reduces the cost in time and resources of redeveloping learning material that already exists. Moreover, in many cases, re-using learning material with well established statistics is imperative, specifically in E-testing. E-testing offers several advantages over traditional paper tests, such as sound and multimedia content, adaptive testing techniques, quick scoring and feedback. Despite these advantages over traditional test, E-testing suffers from the various limitations. One such limitation is *Question Type support*: E-testing systems only support a limited set of question types such as True/False and Multiple choice questions, while disregarding other question types, such as Image Hot Spot, which are important and supported by E-learning standards. Another limitation in E-testing is due to the lack of standardization and the fact that various systems store learning material using an internal format, and only the author has access to his material. Thus, in order to *share* learning material, authors must resort to import/export facilities that are not always available, not to mention the process of locating that material. Finally, Content Coverage refers to the verification that questions in tests and exams cover properly the subject matter. This issue is accentuated further by the fact that exam questions are usually selected at random within the E-testing system.

Knowledge sharing from the learners' perspective, or *harnessing the learner's knowledge*, is another aspect that is starting to get more and more attention in E-learning. Specifically, learners regularly access learning content and tools from outside the scope of the content defined by the tutor, and the tools provided by the E-learning system. This is further accentuated with the fact that learning is becoming more learner centered, where the learner has an access to a multitude of resources. This knowledge is very important since it can enrich the learning content, hence benefitting both the tutor and learners. Indeed, the learner is no longer just a recipient of learning content, but rather he can be an active participant in the creation of such content. Nonetheless, there are no existing mechanisms within E-learning systems to efficiently harness and share this knowledge.

Finally, another aspect of E-learning that requires attention is *privacy*. One of the main advantages of E-learning is its adaptability to the learner's specific needs and preferences. But in order to do so, the E-learning systems must collect large amounts of information about the learner (Arroyo & Woolf, 2005), information that could be misused, and therefore violating his *privacy*, which is the claim of individuals to determine what information about themselves is known to others, as well as when and how it is used (Westin, 1967). The security aspects of E-learning systems do offer some privacy protection; nonetheless it remains unsatisfactory on several levels. Other than the case of *Head-in-the-sand privacy* (by which the learner wants to keep secret his ignorance even from himself), learners might need to keep private different parts of their profile for different reasons. For example, a learner who is following a professional training course, for competitive reasons, would rather keep his identity hidden; yet, he wouldn't mind leaving a trace of his activities in the E-learning system. On the other hand, a secret agent would rather take the training course for a top-secret mission without revealing his identity and without even leaving a trace that someone took this training.

Consequently, we propose *Cadmus*, to address knowledge sharing from the tutor's perspective, *SHAREK* (**SHA**ring **RE**sources and **K**nowledge) to address knowledge sharing from the learner's perspective and finally we present *Privacy Preserving E-learning* to provide a learner's privacy preserving environment for E-learning. Although the three approaches are introduced separately in this work, they could be all integrated within the same E-learning system.

The name Cadmus comes from Phoenician and Greek mythologies: Cadmus was a Phoenician prince who went looking for his sister Europa, abducted by Zeus. During his journey, Cadmus introduced the Phoenician alphabet to the Greeks, who then adapted it into the Greek alphabet.

Cadmus, our system, is an IMS QTI (Question and Test Interoperability) (IMS, 2006) compliant E-testing system. Cadmus is presented to address the limitations of knowledge sharing from a tutor's perspective, in the context of an E-testing system. Hence, since

IMS-QTI supports most question types, and in order to increase *Question Type support*, Cadmus stores questions and tests following the IMS-QTI format. Moreover, Cadmus stores questions and tests in a shared repository, thus allowing authors to *implicitly* share their knowledge and E-testing material. Nonetheless, authors still have the option of sharing or keeping private the authored material. For such a question-and-exam repository to be beneficial it must contain extensive information on questions and exams. The bigger and more useful the repository becomes, the more dreadful is the task to search for and retrieve necessary information and material. Although there are tools to help teachers locate learning material (Tang & McCalla, 2003; Walker, Recker, Lawless, & Wiley, 2004), to our knowledge there aren't personalized tools to help the teacher select exam material from a shared data bank. Consequently, Cadmus incorporates an Exam Questions Recommender System (Hage & Aïmeur, 2005) to help teachers find and select questions for exams. The recommender uses a hybrid, feature-augmentation recommendation approach (Burke, 2002, 2004). Additionally, the recommender system gathers implicit and explicit feedback (Cabena, Hadjinian, Stadler, Verhees, & Znasi, 1997) from the user in order to improve future recommendations. However, selecting questions depending merely on the teacher's preference cannot guarantee a flawless exam with no conflicts. Indeed, we define that a conflict exists in an exam if two or more questions are redundant in content, and/or if a certain question reveals the answer of another question within the same exam. Such conflicts might be frequent typically when a teacher is using shared questions (authored by others), and especially in the automation of the exam creation process. As such, we introduce ICE (Identification of Conflicts in Exams), a module within Cadmus that uses IR (Information Retrieval) techniques to identify conflicts within an exam. ICE (Hage & Aïmeur, 2006a) is based on the vector space model using the cosine function and TF-IDF weighing technique (Singhal, 2001). Furthermore, ICE combines the EQRS techniques in order to recommend replacements for conflicting questions. Finally, Cadmus incorporates the *Topic Tree* (Hage & Aïmeur, 2007), a hierarchy of different topics, of which the author can select those that are covered in the exam, such as the question selection is performed from each to ensure *content coverage*.

On the other hand, SHAREK (**SHA**ring **RE**sources and **K**nowledge) is intended to help learners to share their knowledge and resources. Inspired by the Web2.0 (Schauer, 2005) approach at harnessing collective intelligence, SHAREK (Hage & Aïmeur, 2008b) allows learners to augment the learning material proposed by the tutors by adding or attaching, to a course or lecture, learning resources that they have found (or created). As such, learners can be active contributors into enhancing and enriching the course's learning resources, while maintaining the integrity of the learning material developed by the tutor. In order to achieve this, SHAREK combines Web2.0 technologies and techniques (such as tagging, comments, rating and RSS) with recommendation techniques, notably a multi-criteria collaborative filtering recommender system to help learner manage, share and easily discover learning resources.

Finally, in order to satisfy various privacy needs, we adapt the levels of Privacy and the levels of Tracking introduced in (Aïmeur, Brassard, Fernandez, & Mani Onana, 2007a) to the context of E-learning. In particular, learners are able to receive anonymous transcripts and anonymous degrees such as to prove their achievements to third entities (employers, other E-learning systems, etc.) without compromising their private data. Moreover, in order for the learner to prove that he is the rightful owner of the anonymous transcript or degree, we introduce the concept of *Blind Digital Certificates*, a digital certificate that does not reveal the learner's identity. Finally, we propose *Anonymous Credentials for E-learning Systems* (ACES), a set of protocols that preserves the learners' privacy(Aïmeur, Hage, & Mani Onana, 2008). We are aware that not everybody will embrace our wish for privacy. Nevertheless, as many would agree, we consider privacy to be a fundamental human right: it is not negotiable! This is why we introduce Privacy-Preserving E-learning as an alternative to standard E-learning. Of course, the final choice belongs to each learner.

This document is organized as follows: Chapter 2 offers background information. Chapter 3 details the three issues raised with regards current E-learning systems and offers an overview of related work. Chapters 4, 5 and 6 respectively introduce our proposed solutions to the issues raised in Chapter 3. Specifically, Chapter 4 details Cadmus and the various components, Chapter 5 highlights SHAREK, and Chapter 6 details privacy preserving E-learning. Chapter 7 concludes this work.

Chapter 2 : Background

This section offers background information. Specifically, it introduces E-learning by detailing the components of an E-learning system, as well as by providing an overview of current standards and specification, and a listing of major E-learing systems and their components. The Section also introduces Web2.0 and defines its three pilars. Moreover it offers an over view of how Web2.0 is affecting E-learning.

2.1 E-learning

2.1.1 Components of an E-learning System: LMS and LCMS

There are four parts in the life cycle of E-learning (Figure 2): *Skill Analysis*, *Material Development*, *Learning Activity* and *Evaluation*.

- Skill Analysis: analysis and evaluation of the learner's current skill, and the skill level expected after the training.
- Material Development: develop and decide on the course material needed to attain the course objective.
- Learning Activity: the learner goes through the prepared material.
- Evaluation: the new acquired skill or knowledge is tested and assessed.

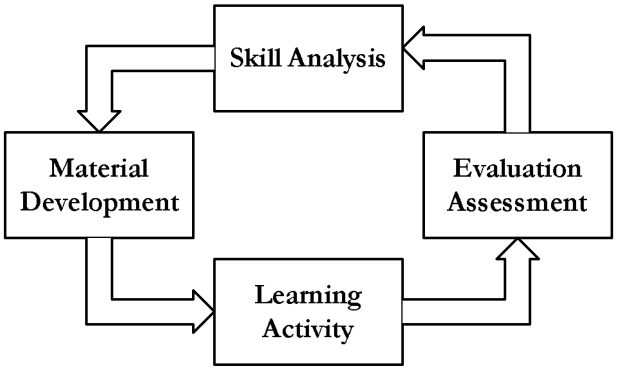


Figure 2: E-learning Lifecycle

Throughout the E-learning lifecycle, there are many components collaborating in order to analyze the learner's skills, develop and deliver proper training material, and evaluate the learning process. Figure 3 illustrates a simplified example of an E-learning system.

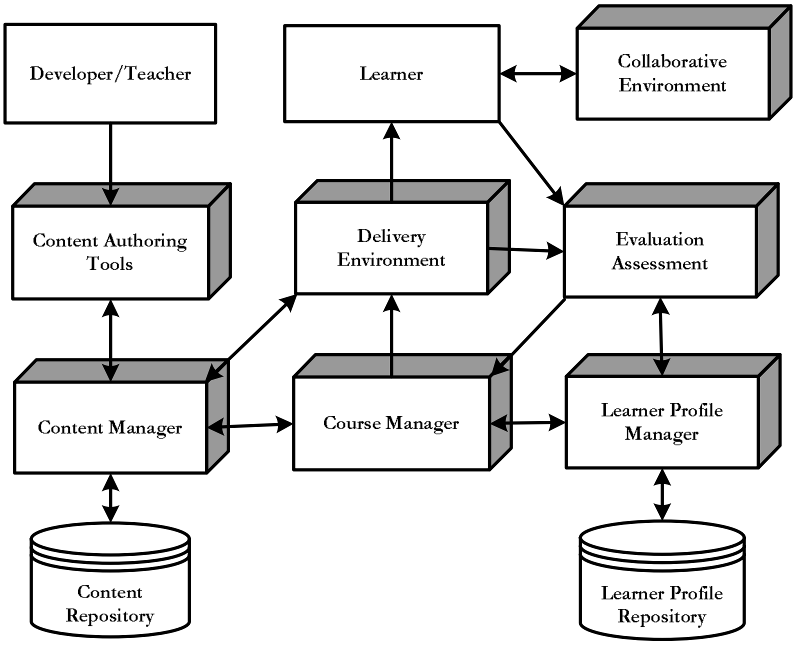


Figure 3: General architecture of an E-learning system

Here is a brief explanation of each component, and then a small example to illustrate the interaction between these different components.

- The Learner is the person who will be going through the learning/training process. He can be a student, an employee in a company, or anyone who wants to increase their skill and knowledge through distant learning.
- The Developer or Teacher is the person developing the training material. He can be a specialist developing training material, a teacher adding or changing some data, or another E-learning system from where training material is imported.
- Content Authoring Tools are available to help the developer create or edit learning material, classify the content, or add a few notes on a subject.
- The Content Manager is used to structure the content for search and retrieval, to add/remove/update content, and for version control.
- The Content Repository is used to store learning material in self contained reusable packages.
- The Course Manager is for managing and deciding the content to be delivered to the learner.

- The Delivery Environment is how and where the training content is delivered. The Delivery Environment depends on the media type of the training material.
- The Evaluation or Assessment tool evaluates the learner's current skill, the skill level expected after the training, and evaluates what the learner gained from the training.
- The Learner Profile Manager is used to keep track of the learner's goals and learning history, and to manage his preferences and learning environment.
- The Learner Profile Repository is where all the information about the learner is stored and accessed by the Learner Profile Manager.
- The Collaborative Environment is a set of tools available for the learner to help him in the learning process; tools such as email, internet, forums, library, etc.

The following example illustrates the interaction between the various components of an E-learning system. Consider a person (the learner) who wants to learn how to cook an omelet. For this matter, assume a cooking E-learning system. To simplify the example, assume that: preparing an omelet has only two parts: preparing the ingredients (eggs and vegetables), and cooking the omelet.

The training content is already developed (by a Chef for example) and stored in the Content Repository. To start the learning process, the Course Manager will check with the Learner Profile Manager and get information about the learner, such as: the preferred Delivery Environment (reading, illustration with pictures, or a demonstration), and what are the skill level and previous training of the learner related to preparing an omelet (does the learner know how to break the eggs, how to prepare the vegetables, …). Now that the Course Manager knows what training the learner needs and in which context, it will query the content manager to look up the training material. In this example, if the learner already knows how to prepare the vegetables, but not the eggs, and prefers to read the instructions with illustrations, then the Course Manager will query the Content Manager only for training material on how to prepare the eggs, in the format of text and illustration (web page, a manual from the library …). At this point, the Course Manager will relay the gathered information to the Delivery Environment and the learner will go through the learning process, relying on the Collaborative Environment for more help if needed. After the learning process is over, an Evaluation, or an Assessment, of the new acquired skill is done (a cooking test and a tasting of the omelet), and the information is reported to the Learner Profile Manager that will record the new training done by the learner, and the newly acquired skills.

The components of an E-learning system are usually divided into two distinct systems: an LMS (Learning Management System) and an LCMS (Learning Content Management System). Although both parts share a similar name and acronym, they have two completely different functions. The LMS primary objective is to manage the learner, and monitor his performance and progress. On the other hand, the LCMS manages the content and the learning material. If we refer back to Figure 3, and we want to order the variable components of the E-learning system under the LMS or the LCMS, we get the result illustrated in Figure 4.

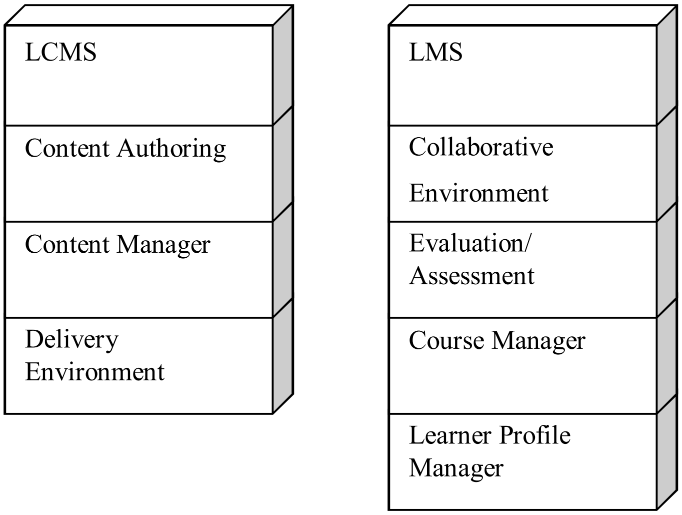


Figure 4: LMS and LCMS

It is important to note that functionalities may sometimes overlap between the LMS and the LCMS, i.e. an LMS system might include some LCMS functionality or vice versa. Table 1 highlights some of the main differences between an LMS and an LCMS.

Table 1: LMS vs. LCMS

	LMS	LCMS
Management of:	Learner	Learning Content
Classroom management	Yes (not always)	No
Learner profile data	Yes	No
Event scheduling	Yes	No
Skill analysis	Yes	Yes (in some cases)
Content creation	No	Yes
Content organization	No	Yes
Manage the content development process	No	Yes
Delivery of content	No	Yes

Most existing E-learning systems actually combine both of the LMS and the LCMS functionalities. Such a system is usually referred to, in the literature, as an LMS (Learning Management System).

2.1.2 E-learning Standards and Specifications

Up until recently, creation and implementation of E-learning material was restricted to a private internal use only, in organizations such as schools, universities, and training departments of companies. In order to deliver the training material, various organizations chose different delivery media, different LMS, different platforms/operating systems and different authoring tools. If an organization upgraded or changed any of its information technologies, the course might not operate, and a change to one component of the course would affect the entire course. Standards and specifications help simplify the development, the use and reuse of E-learning material. As stated in the ADL (Advanced Distributed Learning Initiative) goals, standards and specifications ensure that E-learning material is:

Reusable: modified easily and can be used on different development tools.

Accessible: available as needed by learners or course developers.

Interoperable: functional across different hardware or software platforms.

Durable: easy to modify and updated for new software versions.

Currently, there are many organizations developing different standards for E-learning, each promoting its own standards. Some of the leading organizations with the most widely accepted standards are:

IEEE Learning Technology Standards Committee

The LTSC (Learning Technology Standards Committee) is a committee of the IEEE (Institute of Electrical and Electronic Engineering) to develop standards for learning technology. (ltsc.ieee.org/)

ADL (Advanced Distributed Learning Initiative)

ADL was started, in 1997, by the U.S. Department of Defense and the White House Office of Science and Technology Policy (OSTP). The purpose of the ADL is to standardize E-learning while collaborating with the government, industry, and academia. (www.adlnet.org)

IMS (Instructional Management System Project)

The IMS is an open consortium of industry, government and education members. It was started in 1997 by the National Learning Infrastructure Initiative of EDUCAUSE. The IMS provides a neutral forum for the development of standards and specifications for interoperability and reuse. (www.imsproject.org)

Some of the most widely accepted standards and specification for E-learning are: SCORM (Sharable Content Object Reference Model), LOM (Learning Object Metadata), QTI (Question and Test Interoperability) and LIP (Learner Information Packaging).

2.1.2.1 Sharable Content Object Reference Model

The ADL combined E-learning specifications from international standard groups (IEEE, IMS, AICC and ARIADNE) into a single specification, SCORM. To define SCORM, it is simply a set of standards for developing, packaging and delivering training materials.

The SCORM standard has three components: Content Packaging, Runtime Communication, and Course Metadata.

Content Packaging is the packaging of all the material and components necessary for the delivery of a course. In order for a course to be properly shared, the architecture and the learning resources should be included in the course. SCORM requires each course to include an XML file containing this description. This file is called *imsmanifest.xml*. The imsmanifest file has four sections:

- The preamble section: contains XML pointers to schemas for validating the file.
- The metadata section: contains general course information, such as the title.
- The organization section: contains the sequence and the order of the course material.
- The resources section: contains a list of all the files and resources used in this course.

Many courses require adapting to the learner's actions during the course. Runtime Communication occurs when such a course communicates with the LMS. SCORM runtime communication requires two elements: runtime commands to communicate the learner's information to and from the LMS, and the learner's metadata to store the data on individual learners. Presently, in SCORM there are 8 runtime commands, and 49 student metadata elements.

The SCORM Course Metadata is information packaged within a course when it is archived. The course metadata allows students, teachers or course developers to search through the archives, and easily identify the content of the course. Course metadata can contain the course title, description, etc. SCROM contains a dictionary of metadata terms that can be used for the course description. SCROM uses the IMS Learning Resources Metadata specification, which is based on the IEEE Learning Technology Standards Committee and ARIADNE (Alliance of Remote Instructional Authoring and Distributions Networks for Europe).

2.1.2.2 Learning Object Metadata

LOM (Learning Object Metadata) is a standard for defining the attributes of different LO (Learning Objects[1]), it is an index information used to search and reuse LOs. LOM is a hierarchy of data elements. Nine categories exist at the top level of the hierarchy: general, life cycle, meta-metadata, educational, technical, rights, relation, annotation and

[1] A Learning Object is defined as an entity, whether digital or not, used for learning or training purpose

classification categories. What follows is the exact definition of the categories from the LOM specifications (ltsc.ieee.org/):

a) "The General category groups the general information that describes the learning object as a whole."
b) "The Lifecycle category groups the features related to the history and current state of this learning object and those who have affected this learning object during its evolution."
c) "The Meta-Metadata category groups information about the metadata instance itself (rather than the learning object that the metadata instance describes)."
d) "The Technical category groups the technical requirements and technical characteristics of the learning object."
e) "The Educational category groups the educational and pedagogic characteristics of the learning object."
f) "The Rights category groups the intellectual property rights and conditions of use for the learning object."
g) "The Relation category groups features that define the relationship between the learning object and other related learning objects."
h) "The Annotation category provides comments on the educational use of the learning object and provides information on when and by whom the comments were created."
i) "The Classification category describes this learning object in relation to a particular classification system."

The following are examples of the data elements hierarchy, where the *dot* separates each level in the hierarchy:
General.Title
LifeCycle.Contribute.Date
Technical.InstallationRemarks

For each data elements the LOM defines:
a *name*, to refer to the data element,

an *explanation*, to define the data element,
a *size*, that is the number of values allowed,
an *order*, to specify the order of the values,
an *example*, an illustrative example,
a *value space*, to define the set of allowed values of the data element,
a *datatype* that indicates the type of the value of the data element.

2.1.2.3 Learner Information Packaging

LIP (Learner Information Packaging) sets a list of specifications used in order to exchange learner information, such as name, address and other information. LIP presents the learner information in a defined package. LIP allows profile servers to store learner information in their own format, and provides means to import and export that data between different profile servers. Profile servers use a broad range of information on learners. LIP provides an extensive set of elements to cover most of the data used by profile servers. Most of the LIP elements are optional, thus implementers can only include the specific elements of their actual application. LIP defines eleven segments to group the learner's data:

- Identification: contains basic information about the learner such as name, address, phone number and email
- Goal: contains information about the learner's personal goal and aspirations. A nested structure provides the ability to identify sub-goals.
- QCL: specifies the Qualifications, Certifications, and Licenses. Reflects the achieved accomplishments and indicates the source of the QCL and the level attained.
- Accessibility: contains the learner preferences such as language, disability or accessibility, and technical or physical preferences.
- Activity: contains educational and training actions. This area goes beyond the recording of the activity and result; it provides a space to include a digital representation related to the activity, e.g. a digital representation of a work of art.
- Competency: contains skills the learner has acquired though formal/informal

training or work history. These skills can be related to information in the Activity and/or QCL sections.

- Interest: contains information on the learner's hobbies and recreational activities. These items can be related to information in the QCL section, and can also contain digital representations.
- Transcript: contains a summary of academic achievement. This section introduces the concept of an *exrefrecord* which is a structure that allows the referencing of external data formats, e.g. a document can be stored in an EDI or in a PDF format.
- Affiliation: contains information and description of organizations affiliated with the learner, such as clubs and professional associations.
- Security Key: contains the learners' security information such as passwords, and security keys.
- Relationship: contains descriptions of the relationships between data contained in the other segments.

2.1.2.4 Question and Test Interoperability

IMS QTI (Question and Test Interoperability) (IMS-QTI, 2006; Sclater & Low, 2002) sets a list of specifications used in order to exchange assessment information, such as questions, tests, and results. QTI allows assessment systems to store their data in their own format, and provides a mean to import and export that data, in the QTI format, between various assessment systems. QTI is composed of two parts: the ASI (*Assessment Section and Item*) relates to the test contents, and the Results Reporting relates to the test results. To avoid any ambiguity, QTI uses its own terminology. QTI refers to tests as *assessments*. In order to properly deliver a question, there are other things to know, such as the score for getting it right, and the layout of the question. Therefore, QTI refers to questions and their respective information as *items*. QTI also refers to a group of items in an assessment as a *section*. A section can contain items and/or sections (Figure 5).

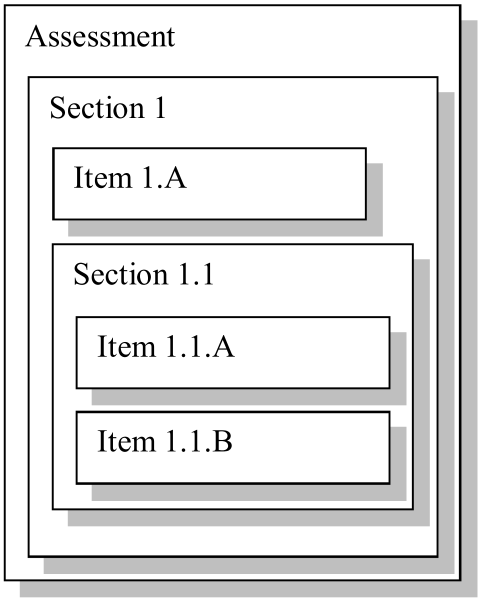


Figure 5: Example of an Assessment

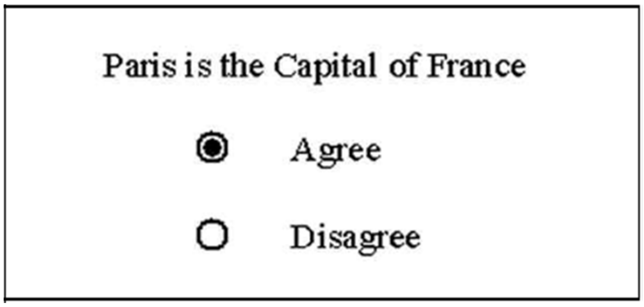
QTI makes available a number of the most used question types: multiple choice, true/false, multiple response, image hot spot, fill in the blank, select text, slide, drag object drag target, order objects, match items, and connect points.

Results in QTI relate to a single learner. However, since a learner can do more than one test, results can contain several results for several assessments. The result reporting data model is made of four constructs:

- Summary: contains data on the test, such as the highest score, and the number of attempts made.
- Assessment: as in the ASI model
- Section: as in the ASI model
- Item: as in the ASI modem

As an alternative result reporting mechanism, one can use the IMS LIP (Learner Information Packaging).

Figure 6 represents a simple True and False example. Figure 7 illustrates the corresponding XML to represent the question of Figure 6 following the IMS QTI standard. The example is taken form the ASI Best Practice & Implementation Guide. For purpose of practicality, the response processing was left out, and only the representational part of the question illustrated in Figure 7. Due to concerns that the IMS QTI is becoming increasingly complex, IMS developed a cut down version of the QTI specification, QTI Lite. QTI Lite deals only with Items, Sections and Assessments are not included. Furthermore, the only item implemented is multiple choice single response. There is support for text and images, but not for video and audio, and QTI Lite maintains a simplified response processing.



Paris is the Capital of France

● Agree

○ Disagree

Figure 6: IMS QTI Example

```
<questestinterop>
 <qticomment>
  This is a simple True/False multiple-choice example using V1.2. The rendering is a
  standard radio button style.
  Response processing is incorporated.
 </qticomment>
 <item ident="IMS_V01_I_BasicExample001">
  <presentation label="BasicExample001">
   <flow>
    <material>
     <mattext>Paris is the Capital of France</mattext>
    </material>
    <response_lid ident="TF01" rcardinality="Single" rtiming="No">
    <render_choice>
    <flow_label>
     <response_label ident="T">
      <material><mattext>Agree</mattext></material>
     </response_label>
     <response_label ident="F">
      <material><mattext>Disagree</mattext></material>
     </response_label>
    </flow_label>
    </render_choice>
    </response_lid>
   </flow>
  </presentation>
 </item>
</questestinterop>
```

Figure 7: IMS QTI Example XML

2.1.3 E-learning Platforms

E-learning has come a long way from being just a book-like content. As illustrated earlier, E-learning systems have become very complex and involve many utilities and tools. This section highlights some of the E-learning systems available in the market, and offers a comparison of the various utilities available in each system. In addition, it lists some of the available virtual universities.

There exist many E-learning platforms in the market today. The following table introduces some of the most known E-learning platform, their manufacturer, and the platform's website. For detailed information on these platforms, please refer to EduTools (EduTools, 2006); which offers a complete and independent analysis of the various E-learning platforms.

Table 2: E-learning Platforms

Platform	Company
ANGEL 6	Adv. Research & Technology Institute
Anlon 4.1	Anlon Systems, Inc.
Atutor 1.3	University of Toronto (ATRC)
Bazaar 7	University of Athabaska
BlackBoard 6	Blackboard
Bodington	University of Leeds
BSCW 4.0.6	OrbiTeam Software
Click2learn Aspen 2	Click2learn
Colloquia 1.3.2	University of Wales
COSE 2.051	Cambridge Software Publishing
CourseWork	Stanford University
Educator	Ucompass
Eledge 3.1	Chuck Wright
Embanet hosting ANGEL	Embanet Corporation
Embanet hosting BlackBoard	Embanet Corporation
FirstClass 7	Centrinity Inc.
Groove Workspace 2.5	Groove Networks, Inc.
IntraLearn SME 3.1.2	IntraLearn
Janison Toolbox 6.2	Janison
KEWL 1.2	University of Western Cape
KnowEdge eLearning Suite	Inter Netion
Learnwise	Granada Learning
Manhattan Virtual Classroom 2.1	Western New England College
Teknical Virtual Campus	Teknical Ltd
The Learning Manager 3.2	The Learning Management Corporation

Virtual-U 2.5	Virtual Learning Environments Inc.
WebCT Campus Edition 4.1	WebCT
WebCT Vista 2.1	WebCT
Whiteboard 1.0.2	Todd Templeton

These different E-learning platforms offer different functionalities, which are divided into 5 categories: Communication Tools, Productivity Tools, Student Involvement Tools, Course Delivery Tools, and Curriculum Design Tools.

2.1.3.1 Communication Tools

Communication Tools are available for electronic communication between teacher and students, and among students themselves. Communication tools are grouped into:

- Discussion Forums: online tools to capture and exchange messages over a period of time (days, weeks). Threaded discussion forums are forums where the messages are divided into categories.
- File Exchange: tools to allow learners to upload and share files with teachers and other students.
- Internal Email: email tools used from inside the course.
- Online Journal/Notes: tools to allow learners to keep journal entries or personal notes.
- Real-time Chat: tools to exchange messages in real time.
- Video Services: tools to allow the teacher to either stream video within the system, or enable video conferencing.
- Whiteboard: offers an electronic dry-erase board that can be used in a virtual classroom environment, and other synchronous functionalities such as application sharing.

2.1.3.2 Productivity Tools

Productivity tools are available to help the learner be more productive. Productivity Tools are grouped into:

- Bookmarks: tools that allow the student to mark important pages, from within or outside the course, and then easily come back to them at a later time.

- Calendar/Progress Review: tools that allow the students to keep track of course progress and assignments plan.
- Orientation/Help: tools to help the student use the different tools in the course management system.
- Searching Within Course: tools that allows student to search and find course material using keywords.
- Work Offline/Synchronize: tools to allow student to complete their work offline, and then synchronize the completed work into the course the next time they login.

2.1.3.3 Student Involvement Tools

Student involvement tools are available to help the students to interact with one another. Student involvement tools are grouped into:

- Group work: tools that provide the capacity to split the class into groups, and provide groups work space, which enables the teacher to assign specific assignments and projects for different groups.
- Self-assessment: tools to allow student to take and review assessment tests. These assessments usually do not count in the final grade.
- Student Community Building: tools that offer a space to create clubs, study groups, or collaborative teams.
- Student Portfolios: tools that offer a space for student to display their work, along with personal information.

2.1.3.4 Course Delivery Tools

Course Delivery tools are made available for teachers in order to help them with the delivery of the course material. Course Delivery tools are grouped into:

- *Automated Testing and Scoring*: tools that allow the teacher to create, distribute, and score the tests.
- Course Management: tools that allow the teacher to control the progress of the class through the course material.
- Instructor Helpdesk: tools to help and guide the teacher how to use the course

management.

- Online Grading Tools: tools to help the teacher grade and provide a feed back to the student, and manage a grade book.
- Student Tracking: tools to allow the teacher track the usage of the course material by the students, and to allow the teacher to perform analysis and reporting on a class or on an individual level.

2.1.3.5 Curriculum Design Tools

Curriculum Design tools are intended to help the teacher create and design the course content. Curriculum Design tools are grouped into:

Accessibility Compliance: is meeting the standards that allow access for people with disabilities to learning material.

Course Templates: tools that help the teacher create the initial structure/template for a course.

Curriculum Management: tools to provide students with customized programs according to previous testing or prerequisites.

Customized Look and Feel: tools to allow the customization of the look and feel of the course, including the ability to institutionally brand a course.

Instructional Design Tools: tools to help the teacher create the learning sequence, for example with a wizard or a course template.

2.1.4 E-testing

Referring back to E-learning life cycle (Figure 2), E-testing is the *evaluation* and *assessment* of the learner's knowledge after the learning activity. As with E-learning, E-testing relies mainly on computers to assess the learner's knowledge. E-testing offers many advantages, when compared to a traditional paper exam, such as sound and multimedia content, adaptive testing techniques, quick scoring and feedback. Moreover, learners tend to have better attitudes towards computer based testing. (Butler, 2003) performed an experiment with 908 student volunteers from 25 different classes at Ball State University. Some students were tested with computer based exams, other with traditional paper exams. Students were surveyed on their attitudes towards: grades and learning, anxiety and readiness, sense of control, cheating and discussing the exam. Students who took the computer based exams showed better attitude about the grade they received, and were more positive towards a higher number of exams. Moreover, they were less anxious about exams, although there was no effect on the feeling of readiness. In addition, students who took the computer based exams reported higher feelings of control, a higher likelihood to cheat, and a higher tendency to discuss the exam's questions.

The E-testing life cycle is composed of three stages (Figure 8) (Brusilovsky & Miller, 1999): Preparation, Delivery and Assessment.

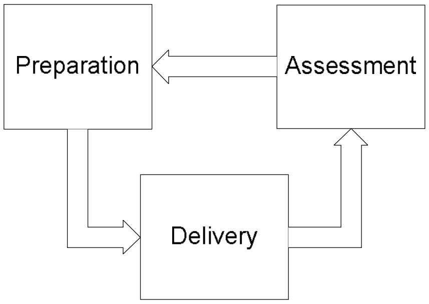


Figure 8: E-testing Life Cycle

The *Preparation* stage consists of preparing the E-testing material and is composed of three steps:

- Authoring: the actual creation of the questions using either a question markup language or GUI (Graphical User Interface) tools.
- Storage: authored questions may be stored in a static manner, such as an html quiz, or in Question Banks where they can be reused in different quizzes.
- Selection: this step consists of selecting and adding questions to exams. One way to do that is selecting the questions manually, or the E-testing system can select questions *randomly* from question pools.

During the *Delivery* stage, the learner takes the quiz or the exam prepared in the previous step. Depending on the delivery environment, questions can be presented to learners using some of the following: HTML, PHP, ASP, Java, and/or Macromedia Flash. Moreover, during the delivery of the quiz, the E-testing system could interact with the learner by offering personalized hints (Mabrouk, 2006), feedback on answers, and adapting the test to the student (such as Item Response Theory (Baker, 2001)).

After taking the quiz, the learner's performance is evaluated in the *Assessment* stage which consists of the following steps:

Evaluation: determine if the answer is correct, partially correct or false. Most question types can be computer scored, and Artificial Intelligence techniques are used to score more complex questions such as short essay (Jordan, Makatchev, & Vanlehn, 2003; Ventura *et al.*, 2004).

Grading: the grade is granted depending on whether the answer to the question is correct, false, or partially correct and the grade assigned to question

Feedback: provide the learner feedback about his answer: whether the answer is correct, incorrect, or partially correct, why it is incorrect, and where to find more information. Feedback can be at two stages (Mathan & Koedinger, 2003): after each question is answered and evaluated or after the whole test is evaluated. Moreover, feedback is sent to the author, and could be explicit (ask the learner for feedback/evaluation) or implicit (in the form of statistics: how many learners answered correctly/incorrectly …).

2.2 Web 2.0

Although the term Web2.0 suggests a new version of the World Wide Web, it does not refer to an update or any technical specifications, but rather to changes in the ways software developers and end-users perceive and use the web. Indeed, the term Web2.0 refers to a perceived second generation of web-based communities and hosted services (such as blogs, Wikis, etc.) which aim to facilitate creativity, and to promote collaboration and sharing between users. Table 3 (O'Reilly, 2005) formulates a sense of what is Web2.0 by example:

Table 3: Web1.0 vs. Web2.0 (O'Reilly, 2005)

Web1.0	Web2.0
Britannica Online	Wikipedia
Personal websites	Blogging
Publishing	Participation
Directories (taxonomy)	Tagging ("folksonomy")
Content Management Systems	Wikis

Specifically, Britannica (http://www.britannica.com/) is the online version of the renowned Encyclopedia Britannica. The content of an encyclopedia is usually authored and validated by well established scholars, and the access to that content costs a certain amount of money, whether by buying the book collection, or an online membership. On the other hand, Wikipedia is a free multilingual encyclopedia, authored by the public at large. Wikipedia has more than 15 million articles, in more than 270 languages, written collaboratively by volunteers around the world (Wikipedia, 2010). Almost all of its articles can be edited by anyone who can access the Wikipedia website.

Generally, Content Management Systems (CMS) enable one or several authorized users, usually the site owners, to easily update the content of websites, whereas Wikis are websites that generally allow visitors at large to modify their content. Additionally, the access to modify a wiki can be restricted to certain authorize users, providing a simple and effective environment for collaborative work. For instance, when relying on email for collaboration, each user must review and modify the document, then email the updated version to the others, who might have already performed some modification, resulting in many versions of the same document. In contrast, in a wiki environment all users access the same document in the wiki and apply their modifications there, where changes can be tracked to the users directly.

The purpose of a blog, or a web blog, is very similar to a personal website: the owner uses it to publish information on a certain subject, including the owner himself (as an online diary). The personal website is usually static: its content is updated, but not on a regular basis. Whereas the blog is usually dynamic: the author adds regular entries and commentary, descriptions of events and/or other multimedia. The blog entries are commonly displayed in reverse-chronological order, and visitors can leave comments on the entries.

Simply defined, taxonomy is the categorization, or classification of an item into a properly defined set of classes or categories. For instance: the classification of music into genres, where the possible classes (classical, rock, pop, etc.) are predefined by domain experts, and rarely change. In contrast, a Tag is freely entered and assigned by users. As such, in the case of music classification, users can freely Tag the song genera, and are not restricted to a predefined taxonomy, hence the name “folksonomy”.

The following point somewhat summarizes the difference between web1.0 and web2.0: publishing vs. participation. Specifically, in Web1.0 (publishing) the content is controlled by the publisher, and the internet users are just the recipient of the information. Whereas in Web2.0 (participation) the users are no longer passive recipients of information, but are active participants in the creation of such information (Figure 9), whether by building personal blogs, participating in Wikis, tagging, rating, sharing, and/or referring websites. A recently published report (Lenhart, Madden, Macgill, & Smith, 2007) indicates that 64% of online teenagers in the US, ages 12 to 17, engage in at least one type of content creation. Moreover, both YouTube and Wikipedia are listed among the top 10 most visited sites by Alexa (Alexa, 2008). Both sites rely heavily on user input.

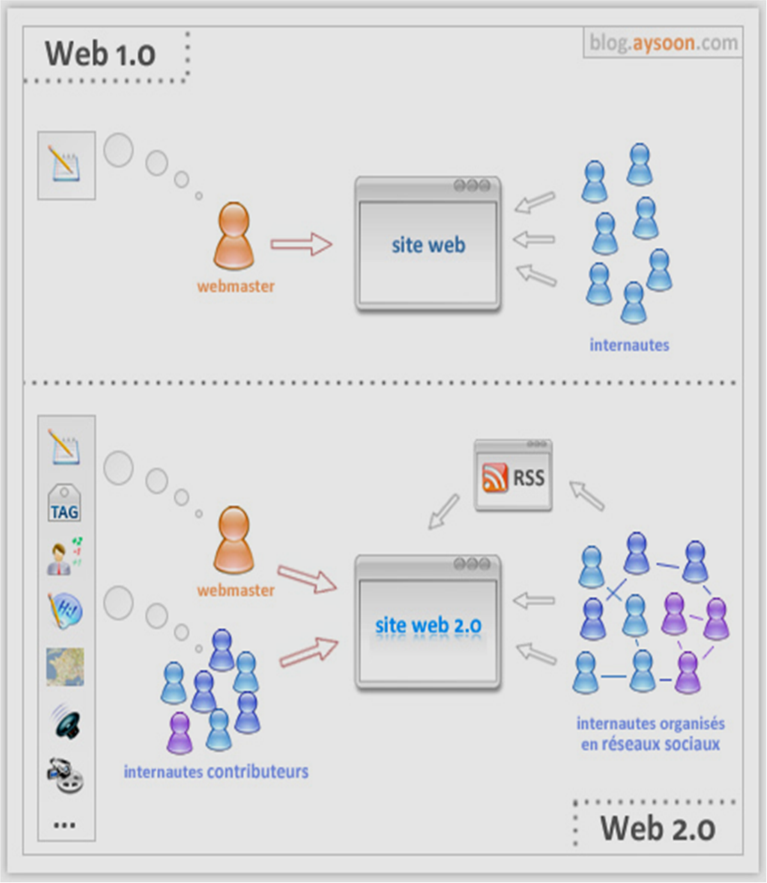


Figure 9: Web1.0 vs. Web2.0 illustrated (Webilus, 2008)

Web2.0 relies on three pillars: the Social Web, Service Oriented Architecture (SOA) and Rich Internet Application (RIA).

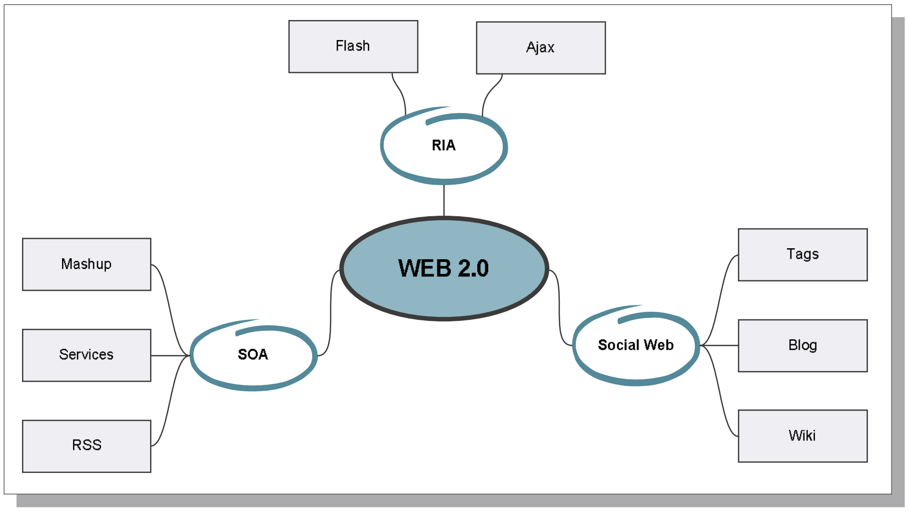


Figure 10: Web2.0 pillars (adapted from (Webilus, 2008))

2.2.1 Social Web

The Social Web refers to the "social interactions" between the users of the web, and the resulting virtual "social groups". It allows users to share their writings, videos, photos, and more with their friends, family, colleagues, or the public at large. For instance, the Social Web includes simple publishing through a blog or a wiki. As such, in the case of the blog the owner of the blog and his *faithful* readers can become a social circle where the readers can comment on the blog posts, or each other's comments. Similarly, with the Wiki, the users who regularly visit add or maintain the Wiki become a virtual social community centered on the Wiki.

Another aspect of the Social Web are the Social Networking Sites (SNS), where users can create a public, or semi-private profile in order to stay in contact with other users of the site, referred to as *Friends*. Specifically, SNS such as MySpace.com or Facebook.com enable users to create a profile page, add other users to their friends list, share photos, and other multimedia content. Other specialized SNS focus on certain aspects of social networking, such as LinkedIn.com which provides an environment for professional networking, and provides specific functionalities such as to recommend someone for a position or a job. Similarly, Academia.edu provides a social networking environment for academics, where researchers can connect and offers particular functionalities, such as to share research paper or to help locate other users with similar research interests.

The main drive behind the Social Web is collaboration and the harnessing of collective intelligence. Common features that exist in the Social Web, such as tagging, rating, comments and recommendation exploit and share the knowledge and experiences of the users. As an example, we will consider social bookmarking sites, such as delicious.com or StumbleUpon.com. Such sites enable users to bookmark their favourite web sites. Moreover, users can share their favourite web sites with other users, or a community of friends. Additionally, users can add tags, comments, even rate these websites (such as thumbs up or thumbs down in stumbleupon). Furthermore, users can search for web sites using the tags, or even receive recommendations based on their profile – stumbleUpon provides a collaborative based recommendation based on the ratings of the users.

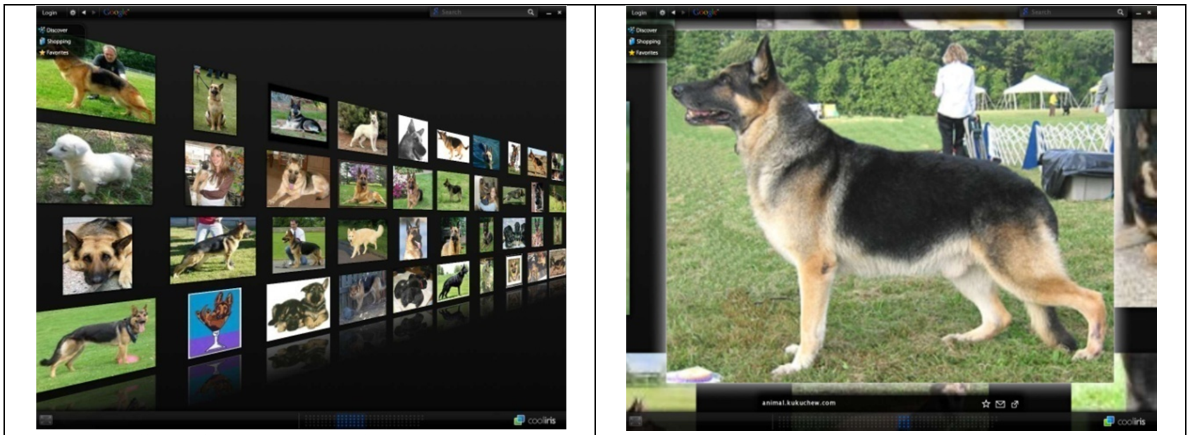
2.2.2 Rich Internet Application

Rich Internet applications (RIAs) are web applications that provide functionalities and interactions similar to desktop applications. Typically, RIAs are delivered through browser add-ons or directly through the webpage by using for instance an Ajax framework (such as Spry or Jquery) or Macromedia Flash.

Consider the following two applications to illustrate how RIA is delivered through a browser add-on: *Cooliris* (http://www.cooliris.com/) and *Coolpreview* (http://www.coolpreviews.com). Cooliris provides an interface to help users when searching for images. Specifically, Cooliris takes the result of a search engine (such as Google image search), and renders it in a animated, flexible interface. For instance, consider a user who is searching the web for images of dogs, particularly the German Shepherd. The user can go to an image search engine and perform the search. Through the search result page, the user is presented with a small preview of the images, and if he wants to view the original version, he must click on the thumbnail, which will take him to the web page hosting that particular image, where he must locate the image. At this stage, if the user is not satisfied, he must return back to the search result page and browse through the remaining thumbnails.

With Cooliris, the user has the option to perform the search within the Cooliris interface, or he can switch to Cooliris from the search result page. In the first case, the user must specify which image search engine to use, including Google, Yahoo, Flickr and

YouTube. All the thumbnails are organized in rows on a wall where the number of rows is user specified and ranges between 1 and 7 inclusively. A scroll bar allows the user to browse the wall (Figure 11 - a), and when the user clicks on the certain thumbnail, the image is automatically enlarged (Figure 11 - b) and the user has the option to go back to the wall (just by clicking outside the enlarged image) or he can either add the image to his favourites, email it to a friend, or go the original website where the image is hosted. Note that in order to add an image to the favourites or send it to a friend, the user must register for an account with Cooliris. Moreover, when the user goes to the website containing the specific image, Cooliris is minimized and a button is made available for the user to revert back to Cooliris. Note that in the case when searching for videos within Cooliris, when the user selects a certain thumbnail and it is enlarged, the user has the option to play the video directly from within Cooliris.

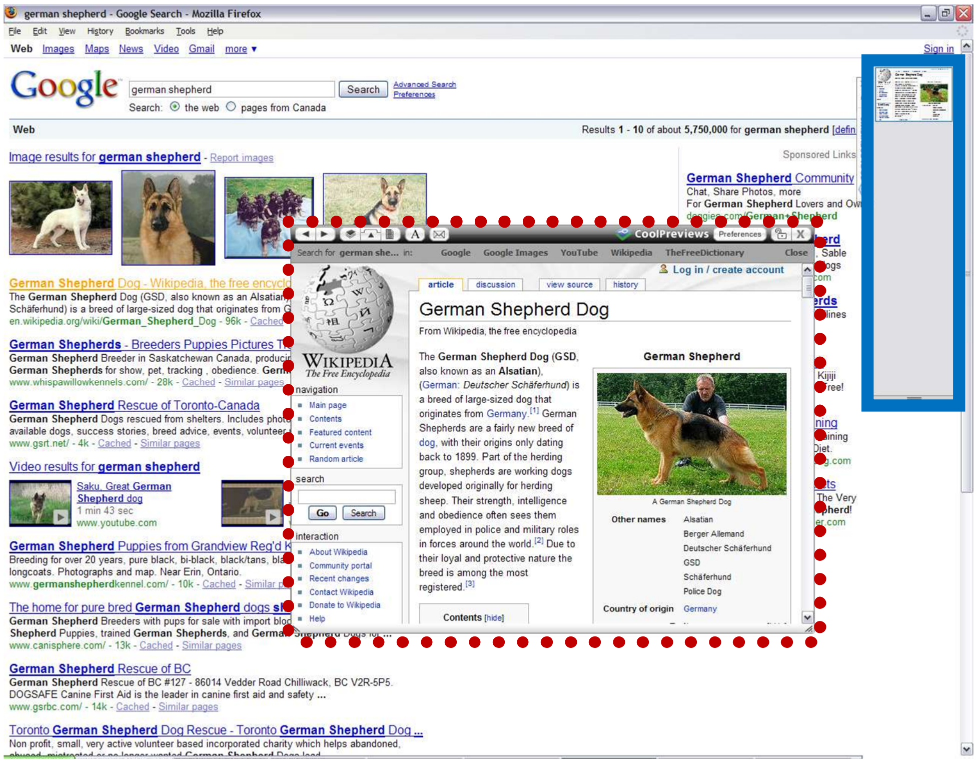


a) Scrolling and browsing the wall

b) Previewing a thumbnail image

Figure 11 : Cooliris preview

Similarly to Cooliris, Coolpreview is designed to help users while browsing the web. While the first is designed for images and videos, Coolpreview is designed for web pages. Specifically, Coolpreview offers the user the possibility to preview the page attached to with a link, without ever leaving the current page. For instance, consider again the case of the user interested in the German Shepherd dog, but now he is searching for websites instead of images. After performing the search on Google, usually the user has one of three options: first, following each link returned by Google, then using the browser's back button to return to the search result page; second, opening each link in a new browser window, and third opening each link in a new tab. On the other hand, Coolpreview offers the user with the possibility to open a virtual window that displays the target website (Figure 12 in the middle, highlighted in the red dotted square). Additionally, Coolpreview offers many functionalities, including the possibility to display the next link on the page (or in this case the next web site), add the page to a temporary bookmark stack (Figure 12 on the right hand side, highlighted in blue solid square), open the page in a new tab or even email the link.



Results 1 - 10 of about 5,750,000 for german shepherd [defin

Image results for german shepherd - Report images

German Shepherd Dog - Wikipedia, the free encycl
The German Shepherd Dog (GSD, also known as an Alsatian
Schäferhund) is a breed of large-sized dog that originates from G
en.wikipedia.org/wiki/German_Shepherd_Dog - 96k - Cached

German Shepherds - Breeders Puppies Pictures T
German Shepherd Breeder in Saskatchewan Canada, producin
German Shepherds for show, pet, tracking , obedience. Germ
www.whispawillowkennels.com/ - 28k - Cached - Similar pages

German Shepherd Rescue of Toronto-Canada
German Shepherd Dogs rescued from shelters. Includes phot
available dogs, success stories, breed advice, events, volunteer
www.gsrt.net/ - 4k - Cached - Similar pages

Video results for german shepherd
Saku, Great German Shepherd dog
1 min 43 sec
www.youtube.com

German Shepherd Puppies from Grandview Reg'd K
Breeding for over 20 years, pure black, bi-black, black/tans, bla
longcoats. Photographs and map. Near Erin, Ontario.
www.germanshepherdkennel.com/ - 10k - Cached - Similar p

The home for pure bred German Shepherd dogs sl
German Shepherd Breeders with pups for sale with import bloc
Shepherd Puppies, trained German Shepherds, and German Shepherd Dogs for ...
www.canisphere.com/ - 13k - Cached - Similar pages

German Shepherd Rescue of BC
German Shepherd Rescue of BC #127 - 86014 Vedder Road Chilliwack, BC V2R-5P5.
DOGSAFE Canine First Aid is the leader in canine first aid and safety ...
www.gsrbc.com/ - 14k - Cached - Similar pages

Toronto German Shepherd Dog Rescue - Toronto German Shepherd Dog ...
Non profit, small, very active volunteer based incorporated charity which helps abandoned,

German Shepherd Community
Chat, Share Photos, more
For German Shepherd Lovers and Ow
doggies.com/German+Shepherd

German Shepherd Dog
From Wikipedia, the free encyclopedia

The German Shepherd Dog (GSD, also known as an Alsatian), (German: Deutscher Schäferhund) is a breed of large-sized dog that originates from Germany.[1] German Shepherds are a fairly new breed of dog, with their origins only dating back to 1899. Part of the herding group, shepherds are working dogs developed originally for herding sheep. Their strength, intelligence and obedience often sees them employed in police and military roles in forces around the world.[2] Due to their loyal and protective nature the breed is among the most registered.[3]

German Shepherd
A German Shepherd Dog

Other names	Alsatian Berger Allemand Deutscher Schäferhund GSD Schäferhund Police Dog
Country of origin	Germany

Figure 12 : Coolpreview

On the other hand, there is a multitude of web pages that illustrate the use of RIA are the web-based virtual computers, such as G.ho.st (http://g.ho.st/). Such environments provide a virtual computer accessible online, which provides many functionalities and tools, including disk space (5 Gbytes in the case of G.oh.st), media player, even an office suite to create and store documents, spreadsheets and presentations.

2.2.3 Service Oriented Architecture

Service Oriented Architecture (SOA) (Booth *et al.*, 2004) is an architectural style where the main goal is to relax the dependencies between various components and to achieve loose coupling. Specifically, a *service* is a task performed by the *service provider* to achieve a desired end result for a *service consumer*. Consequently, a service-oriented architecture is a collection of services (service providers and consumers), where these services communicate with each other. Such communication could be just simple data passing or it could involve two or more services coordinating to perform a certain activity. Note that the service provider can also be a service consumer. SOA usually employs a *find-bind-execute* paradigm as illustrated in Figure 13. The service provider *registers* in the directory, providing a detailed description of the service provided. The service consumer queries the directory to *find* a suitable service provider. When found, the service consumer sends a request to the service provider, who in turn sends the response to the service consumer.

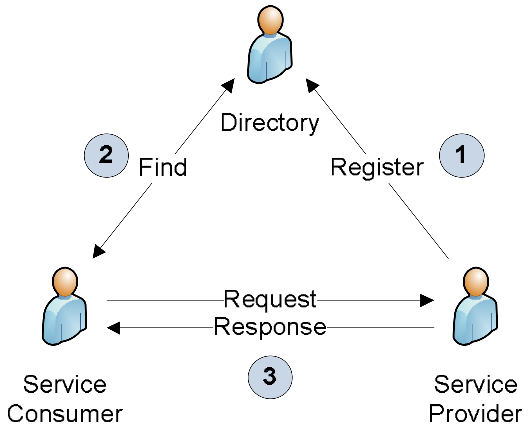


Figure 13 : Simple SOA - adapted from (Booth, *et al.*, 2004)

Some of the main advantages of using SOA are:

Reusability: In an SOA, a requesting application only needs to know the public interface of a desired service. Hence, the functions of an application are generally easier to access as a service in an SOA than in some other architecture. Consequently, integrating applications and systems as well as reusing their different components can be much simpler.

Interoperability: the interaction between loosely-coupled services implies widespread interoperability. In other words, within a SOA, the desired objective is for service consumers and service providers to communicate and understand each regardless of the platform they are on. This objective can be met by having a standard way of communication between services, a way that is consistent across various platforms, systems, and languages.

Scalability: since services in a SOA are loosely coupled, the applications that exploit these services tend to scale easily, or at least easier than applications in a more tightly-coupled environment. This is due in large to the fact that there are few dependencies between the requesting application and the services it uses.

Flexibility: loosely-coupled services are typically more flexible than tightly-coupled applications. In a tightly-coupled architecture, the different components are usually strongly bound to each other, typically sharing semantics and libraries, and often sharing their state. This makes it difficult to evolve the application to keep up with changing requirements. The loosely-coupled and asynchronous nature of services in a SOA allows applications to be more flexible, and to easily evolve in order to adapt to changing requirements.

The most common approach used to achieve the interactions between various services are the web services-based SOAs. A web service is a service that communicates with clients using a set of XML-based standard protocols and technologies, such as WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery, and Integration). The universality of the aforementioned protocols and technologies has made web services the most predominant approach to implementing a SOA. In short, **WSDL** (Web Services Description Language) an XML format used for describing a Web Services interface forms the basis of web services.

SOAP (Simple Object Access Protocol) is an XML-based protocol that enables applications to exchange messages and information over the internet. A SOAP message (figure) consists first of an *Envelope* that identifies the XML document as a SOAP message. The SOAP envelope contains an optional *Header* which provides information on authentication, encoding of data, or how the recipient should process the message. Moreover, the envelope includes the *Body* which contains the actual message or information.

UDDI (Universal Description, Discovery, and Integration) is a directory storing information about web services. In short, UDDI provides the definition of a set of services and supports the description and discovery of Web Services providers, the Web Services they make available, and the technical interfaces used to access those services. The idea is to locate organizations and the services they offer, much like using a phone book.

Figure 14 highlights the SOA architecture presented in Figure 13 based on web services. Note that all the messages are exchanged using SOAP.

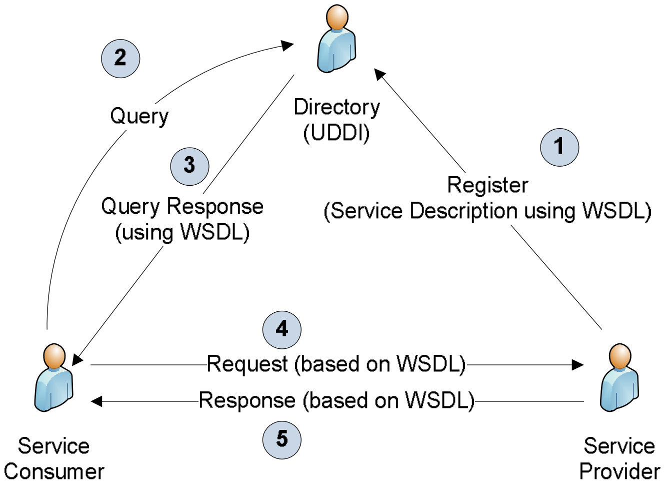


Figure 14 : Web service scenario

The flexibility and interoperability of SOA and web services has lead to a new type of web applications called Mashup. Specifically, a mashup describes a Web application that combines multiple services and/or data sources into one single application. For instance, an example of a mashup is Woozor (www.woozor.com): it combines Google maps with information form weather.com in order to provide weather forecast from around the world. Another example is Netzwelt (www.netzwelt.de), a German online magazine. It combines free and legal promo MP3s on the net with Upcoming.org tour dates, Amazon CD reviews, YouTube videos and Akuma MP3 download store.

2.2.4 E-learning 2.0

Now that Web2.0 is clearly defined, let us define what E-learning 2.0 is all about. Similarly to Web2.0, E-learning2.0 does not refer to a new class of LMS (Learning Management Systems) or a new educational technology. Rather it is a natural consequence of changes in how tutors and learners perceive learning. Indeed, in recent years, education has been shifting from being *tutor-centered*, to being *learner-centered*. In tutor-centered education (Figure 15), the tutor is the active participant in the educational process and learners are considered as passive receptacles of knowledge. Tutor-centered education is a *one size fits all* approach.

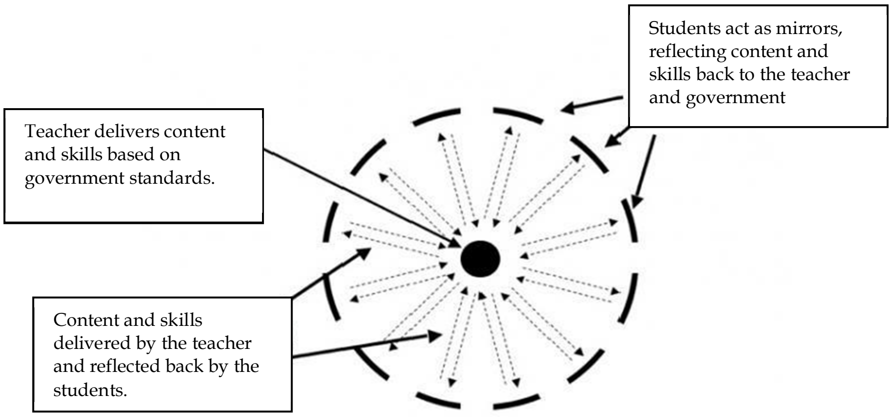

Students act as mirrors, reflecting content and skills back to the teacher and government
Teacher delivers content and skills based on government standards.
Content and skills delivered by the teacher and reflected back by the students.

Figure 15: Tutor-centered education (Webilus, 2008)

On the other hand, in learner-centered education (Figure 16), the learners have access to a variety of knowledge sources and the tutor places more emphasis on what learners can contribute to the educational encounter. It is important to note that E-learning2.0 is not a consequence of Web2.0. Indeed, both share the same basic concept where the user/learner is not only a spectator and a simple consumer of information, but rather an active participant in the creation of such information. As such, one can view Web2.0 tools and technologies as a natural recourse to achieve learner-centered education.

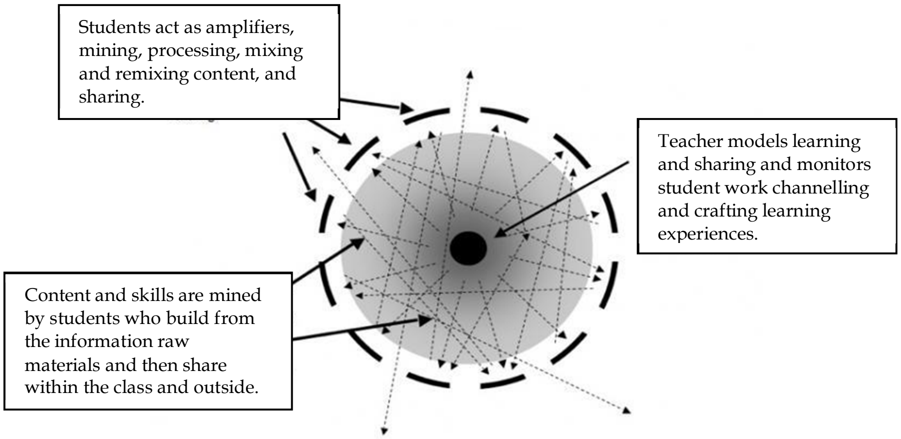

Students act as amplifiers, mining, processing, mixing and remixing content, and sharing.
Teacher models learning and sharing and monitors student work channelling and crafting learning experiences.
Content and skills are mined by students who build from the information raw materials and then share within the class and outside.

Figure 16: Learner-centered education (Webilus, 2008)

What follows are some examples of “Web2.0” tools and websites designed for and/or used in learning. For instance, a webcast consists of distributing media content over the using streaming media technology. A webcast may be distributed live or on demand. In essence, webcasting is “broadcasting” over the internet. A simple example of webcasting is a TV station that simultaneously streams over the internet the show being broadcasted on TV. On the other hand, a podcast is a series of media content made available via syndication, such as RSS. Dedicated software applications, known as podcatchers automatically identify and retrieve new available media files. In order to clearly understand the difference between a webcast and a podcast, consider the following example: you like to watch a TV show called “example show”. If the show is aired once a week at a certain time, than you need to be home, in front of your TV at that specific day and time to watch the show. The only way to see an episode more than once is on reruns. This illustrates a *live webcast*. On the other hand, if that same show is available on demand TV, then you can watch the available episodes at your convenience. Nonetheless, you still need to be home in front of your TV. This is similar to an on *demand webcast*. Now, imagine the TV station has a delivery boy called “podcatcher” who will faithfully deliver to your house, every time a new episode of your show is ready, a DVD with that episode on it. In this case, you can watch any released episode at your convenience, whether on your home TV, or on a portable DVD player on the train on your way to work. This case illustrates the *podcast*.

The utility of webcasts and podcast in E-learning is very clear: tutors can either webcast their lectures live to students, or the lectures are made available on demand or through a podcast. Note that a lecture can consist of various media, such as audio only, a slide presentation with audio, a recording of the tutor, etc.

Currently, webcasting and podcasting can enhance the learning experience(Lau *et al.*, 2010), and are being used in several universities worldwide (Shim, Shropshire, Park, Harris, & Campbell, 2007). It is important to note that webcasting and podcasting are not just used by virtual universities, but also as a complement to lectures in *traditional* classrooms, for instance, Berkely makes publicly available webcasts of several courses (available at http://webcast.berkeley.edu/), consisting of either an audio recording of the tutor's lecture, a video recording of the tutor giving his lecture, or a slide presentation of the lecture with the explanations of the tutor.

Alternatively, wikis are websites that generally allow visitors at large to modify their content. Nonetheless, wikis generally can support authentication, such that certain members can modify only certain pages. This feature is important since it enable the use of wikis in group work assignments. Wikis offer the possibility of central access for all the users or limited user groups, which makes it an ideal choice for running projects, drafting documentations and other group work. As such, wikis are used to promote team work and collaboration between students (Raitman, Augar, & Zhou, 2005a). Alternatively, wikis can also be employed by tutors to collaborate on creating learning content. For instance, wikiversity (http://wikiversity.org/) offers tutors the chance to collaborate and create freely available learning resources, where currently, on the English site of wikiversity, there are more than 10,000 pages available, covering various topics.

Similarly, SuTree[2] and eduSLIDE.net[3] offer both learners and tutors access to a variety of learning resources. Specifically, SuTree.com offers a variety of how-to videos, ranging from learning how to whistle, to following a complete course watching MIT lectures. eduSLIDE allows tutors to create lessons (presentations) and group them into courses, making these courses available for learners.

Additionally, many existing "web2.0" pages and tools can help learners during the learning process. For instance, Footnote.com allows students to access primary source documents and photos, and to easily create and post online history reports. Moreover, VoiceThread[4] can be used by both tutors (to create lessons) and learners (for homework purposes) to upload pictures and create an audio narrative to go along with them. VisualThesaurus[5] offers, as its name indicates, a visual thesaurus. Specifically the lookup word is presented in the center of the graph, and edges connect the lookup word with its synonyms. A color code is used on the edge connecting the word to its synonyms to

[22] Available at http://sutree.com/, last accessed on 01/2010

[3] Available at http://www.eduslide.net/, last access on 01/2010

[4] Available at http://voicethread.com/, last accessed on 01/2010

[5] Available at http://www.visualthesaurus.com/, last accessed on 01/2010

indicate whether the synonym is a noun, verb, adjective or an adverb. Moreover, the edge connecting the lookup word with its antonym is presented differently. Wayfaring.com uses Google maps to list podcasts and webcasts from about 68 universities worldwide. wePapers[6] allows users to share academic papers, ranging from research papers, tutorials, lectures, to tests and exams. Moreover, users can comment, and even ask questions to the community about these papers.

Another useful browser add-on is Diigo[7], which provides learners with the ability to highlight specific parts of webpages, add sticky notes and comments (private or public) to the highlighted sections or the whole page, and learners can share the highlights and notes with their Diigo social network.

2.2.5 Personal Learning Environments

The proliferation of tools and websites such as listed earlier has led to the concept of Personal Learning Environment (PLE) (Pusnik, Sumak, & Hericko, 2010). PLE is a combination of tools and processes, whether formal or informal, which learners use to gather information, reflect on it and work with it. The appeal of PLE for learners relies in the fact that they can choose the tools that best suit their preferences. An interesting representation I came once across compares a Learning Management System (LMS) and a Personal Learning Environment (PLE) using the following analogy: an LMS is similar to a *Swiss army knife* containing a set of tools, some of which you might never used. On the other hand, a PLE is like having a box containing the tools you use, but most importantly tools that you chose and prefer. Indeed, although it might be more practical to fit a large set of tools into your pocket (Swiss army knife analogy), having only the specialized tools that you are comfortable with does have it advantages.

2.3 Security of E-learning systems

Security is an important aspect of E-learning. Indeed, most (if not all) of the E-learning systems and Intelligent Tutoring Systems store information about the learner, and use an

[6] Available at http://www.wepapers.com/, last accessed on 01/2010

[7] Available at http://www.diigo.com/, last accessed on 01/2010

underlying layer of communication between the client computer (where the learner is working) and the server (where the application is actually running). In this section we first introduce some notions about security, and then we highlight some underlying threats that need to be considered, from a security point of view.

2.3.1 Pillars of security

Information security, (in this case the learners' information) is based on three pillars: *Confidentiality*, *Integrity*, and *Availability*. Maintaining the Confidentiality of the information involves protecting the information from unwarranted disclosure, and making sure that only the users with the proper privileges have access to that information. In other words, the user can only access the information he is permitted to. On one hand, the confidentiality of the information is considered during the transfer of the data between the client and the server. Indeed, with the availability of high bandwidth and the speed of the internet connection we tend to forget that in order to reach a certain website, the connection goes through several connection points. Indeed, running a simple "*tracert*" to the website we are trying to reach displays the detailed information about the route taken by any information exchanged between the user's PC and the web server hosting that certain web page. Figure 17 highlights such a route, where the circles in the cloud illustrate possible connection points.

Consequently, imagine a learner sending his login information, or even uploading his homework to the E-learning system: the data could be intercepted and used maliciously by another learner. Similarly, imagine the learner requesting his grade report for the E-learning system: that information could be viewed by an unauthorized person while being sent from the server to the learner. On the other hand, the confidentiality of the

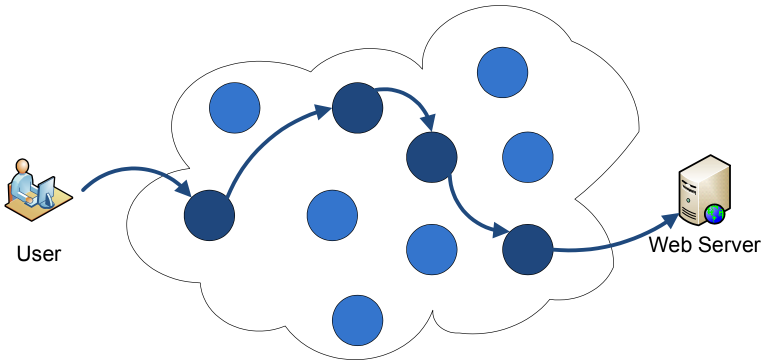


Figure 17: Information route from user's PC to web server

information is also considered while it is being stored within the system. Indeed, imagine that any person with access to the registrar's office of your academic institution can also access and view your academic record. Regardless whether you have a good or bad academic record, this is inacceptable. Similarly, the confidentiality of the information stored within the E-learning systems should be guarded, and only the persons with the proper access privileges might have access to that information.

The second pillar is Integrity, which enforces the validity and authenticity of the data. In other words, ensuring information integrity protects the data from any tampering or modifications from unauthorized users. To begin with, the integrity of the information is considered during the transfer of the data between the client and the server. Indeed, consider taking a learner taking an online quiz. The answers to the quiz's questions are sent through the same route described earlier. Without any integrity verification mechanisms, to insure that the data was not modified through the transmission, a malicious user can intercept the answers of the learner and modify them before they reach the E-learning system, successfully tampering with the learner's score. Additionally, the integrity of the information is also considered while it is being stored within the system. Indeed, again in this case, without the proper mechanisms to protect the data integrity, a malicious user with access to the E-learning system could tamper with the information (increasing or decreasing a test score for instance) unnoticed.

The third pillar is Availability, which relates to the availability of the E-learning system. Indeed, such systems must be available at all time, and provisions must be considered to implement and ensure this availability. One might be tempted to think: why is this crucial? Well, consider a learner without access to the system the day homework is due. Not only will the learner not have access to any necessary learning resources available through the E-learning system, but he will not be able to submit his homework. Moreover, consider a learner performing an online quiz. Even if the system was not available for only a few minutes, it is still precious time lost, not to mention the stress and the emotional pressure caused to the learner.

2.3.2 Security Threats

This section highlights *some* of the existing security threats. Actually, what we list here is the tip of the iceberg, and is intended to raise the awareness that when on the internet, we are not as safe and secure as we think we are.

2.3.2.1 SQL Injection

SQL Injection exploits security vulnerabilities at the database level of the system. Such vulnerabilities occur when the user input (data provided by the user) is not properly filtered, allowing the user input to contain executable SQL code. For instance, consider an authentication system that asks the user to provide a user name and a password, and uses the following query to validate the user's credentials:

```
SELECT * FROM user_table
WHERE        user_name = provided_user_name
AND    user_password = provided_password”.
```

If the user enters valid values for the variables *provided_user_name* and *provided_password*, the query will work just fine and as expected. Nonetheless, if a malicious user provides the following user name: *“abcd OR 1=1 --”*, the *WHERE* clause of the query becomes: *“WHERE user_name = abcd OR 1=1 -- AND user_password = provided_password”*. In this case, regardless of the password provided by the malicious user, since the *“--”* is a comment in SQL, the database system will ignore anything that comes after it. Consequently, the query will always return the entire users list from the *user_table* due to the *“OR 1=1”* in the query.

Although the example portrayed here is fairly simple, malicious users using the SQL Injection attack can formulate far more complex queries and do a large amount of damage. Indeed, just to cite a couple of recent events, in August of 2009, the BBC published a story about a US citizen allegedly stealing 130 million credit card numbers using an SQL injection attack (BBC, 2009b). More recently, in December of 2009, the New York Times reported on a hacker who accessed, using an SQL Injection attack, the RockYou (rockyou.com) database where he found unencrypted login information for more than 32 million user accounts (O'Dell, 2009).

2.3.2.2 Cross Site Request Forgery

Cross-Site Request Forgery (CSRF) is an injection type attack, where a malicious web site causes the user's browser to perform unwanted actions on a trusted site. Specifically, the malicious website would try to inject *malicious* requests to the trusted website. For example, consider a user that is logged in to his banking website to pay his bills, while at the same time browsing the malicious website. The malicious website could send a request to the banking site, asking for a money transfer to a specific account held by the attacker. Specifically, the malicious website could post an image that links to the website banking site instead, using the following link for example:

"http://mybank.com/transfer?from=account&amount=1000&to=malicious"

It is important to note that such attacks are difficult: the attacker must first gather different information about the targeted site, and the targeted user. Moreover, in order for this attack to happen, the user must simultaneously have a valid session opened on the targeted site, and be connected to the site of origin of the attack. Nonetheless, these vulnerabilities are real and could have a devastating effect. In this report (Zeller & Felten, 2008), a professor from Princeton and his graduate student report on successful CSRF attacks against several popular websites, including ING Direct (ingdirect.com), where they were able to transfer funds out of users' accounts.

2.3.2.3 Denial of Service

A *denial-of-service* attack (DoS attack) or *distributed denial-of-service* attack (DDoS attack) is an attempt to overload a computer's resources in order to render it unable to process legitimate users' requests. It is generally conducted against web servers, saturating them with *fake* requests, making them unable to process genuine users' requests. One common method of attack involves overwhelming the target machine by saturating it with *fake* communications requests, such that it cannot respond to legitimate request, or responds so slowly as to be rendered effectively unavailable. A distributed denial of service attack (DDoS) occurs when multiple systems collaborate to flood the resources of the targeted system. Often, DDoS attacks are conducted using *zombie* machines, computers that were compromised and are now being controlled by the attacker.

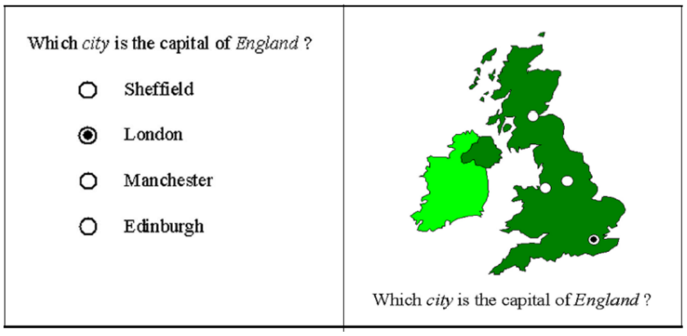
In July 2009, South Korea witnessed one of its the largest cyber attacks. DDoS attacks were used to crash the websites of dozens of government offices and banks among others (Lee, 2009). Additionally, in August of 2009, Twitter and Facebook were the victims of similar attacks. While Twitter was taken offline for a while by the attacks, Facebook's service was reduced (BBC, 2009a). Such attacks are quite common and usually used for extortion purposes (Messmer, 2010).

Chapter 3 : Issues to solve

Although E-learning has advanced considerably from being book-like to becoming rich in content and adaptive, there are still some of its aspects, such as E-testing, that are still limited due to the fact that they are underdeveloped or just simply ignored. In this section we raise these issues and limitations, then we highlight some of the existing systems, how they address these issues and the limitations in their approaches.

3.1 E-testing

The first limitation in current E-learning systems is *Question type support*. Most E-learning platforms support only the basic question types such as True/False, Multiple Choice, Multiple Selection, Fill in the Blanks and Short Answer (EduTools, 2006). Other question types, although supported by IMS QTI (IMS-QTI, 2006), are either completely ignored or partially supported by some systems, question types such as Image Hot Spot, Ordering Objects, Matching and Connect Points. One could argue that question type support affects the flexibility of the system, but not its efficiency: for example, Figure 18 illustrates how an Image Hot Spot question can be modified and presented as a Multiple Choice question (IMS-QTI, 2006).



Which *city* is the capital of *England* ?

○ Sheffield

● London

○ Manchester

○ Edinburgh

Which *city* is the capital of *England* ?

Figure 18: Image Hot Spot into a Multiple Choice

Nonetheless, this flexibility is important in E-testing to properly adapt the question to the learner's preferences, therefore affecting efficiency. For example, a learner with a visual learning style (Flaherty, 1992; Keefe, 1979) would be at disadvantage if the question illustrated in Figure 18 was presented to him under the Multiple Choice format instead of the Image Hot Spot. According to Howard (Howard, 1998), the *learning style* is a factor that influences a learner's educational performance. Therefore, it is important to determine the learning style of the student and most importantly, the E-learning system must adapt it.

Another issue is *implicit Knowledge Sharing*. E-learning and E-testing material is only accessible to their developers. Tutors must explicitly make available their material to others. This is a major drawback for several reasons: for one, due to the *lack of standardization* and the fact that E-learning platforms store E-learning and E-testing material in their own internal format, for two tutors to share their data they must explicitly use import/export tools (which are not always available (EduTools, 2006)) to perform the exchange. Knowledge sharing is a very important; it helps tutors share their experience and knowledge in order to deliver better education. Moreover, in the case of E-learning and E-testing, sharing helps reduce the time and cost of redeveloping learning material which was already developed. Furthermore, item statistics are important for personalization (such as in Item Response Theory), therefore it is advantageous to reuse an item with well established statistics instead of just recreating it.

One more issue is *No Test Personalization*. In this case, we consider the issue from a tutor and a learner point of view. From the tutor point of view, E-testing systems help the tutor create exams, by either recommending items, or creating a complete exam. Nonetheless, in both cases the item selection procedure is done pseudo-randomly (Blackboard, 2004; EduTools, 2006; WebCT, 2003). The E-testing system recommends items (or selects the items for the test) based on tutor specified restrictions with regards to the content. Nonetheless, the order in which the items are presented is random. In the second case the exam questions are selected randomly. In this context, personalization is important since it removes the *random* factor and increases the tutor's confidence. Indeed, taking into account the tutor's preferences to recommend test items or to create the tests for him increases his confidence in the system. On the other hand, test personalization is as important to the learner. Nonetheless, E-testing systems, that offer auto-evaluation tests to help the learner prepare for an exam, do not personalize the auto-evaluation process. Instead, the selection is performed randomly from a collection of items predefined by the tutor.

An additional issue is *Content Coverage*. In the context of an exam, content coverage refers to the sufficient coverage of various subjects included in the exam. This is particularly important since a tutor needs to test all the learners' knowledge on various aspects of a certain subject. This issue becomes important especially considering the fact that the questions are selected at random.

3.2 LMS and PLE

Many PLE advocates portray an LMS (Learning Management System) (Chapter 2.1.1) as being inflexible and used to control the learning and the learner, whereas a PLE is portrayed as easy to use, personalized, and liberated. In short, LMS is equivalent to controlling how you learn, whereas PLE corresponds to giving you control over how you learn. Although controlled and passive learning reduces self reliance and causes loss of curiosity and creativity, an uncontrolled education would create a shortage of certified labor and would introduce unqualified people into the labor pool. Ideally, a middle point between the flexibility of PLEs and the rigidity of LMSs would capitalize on the advantages of both worlds, while circumventing their weaknesses. Indeed, the driving concept behind learner centered education is to *promote freedom and flexibility* in learning, while *maintaining some control*. Specifically, this is where E-learning stands today (Figure 19). The Tutor delivers the learning content to the learner through the LMS. On the other hand, the learner has access to the controlled environment provided by the LMS as well as a PLE containing the set of his favorite tools and resources, which are external to the LMS. As such, the learner can *freely* perform the learning activity, relying on the content and tools provided through the LMS, and on external *uncontrolled* resources through the PLE. In addition, the learner has access to both his personal social network (outside the LMS), and a peer network through the LMS. Note that some peers can also be part of the learner's external social network. In such a scenario, the tutor *controls* the curriculum (which courses and topics the learner must complete), and he can *validate* the learner's knowledge through assessments. On the other hand, the learner has the freedom to choose *how* to complete the learning activities: whether by solely using the content and tools provided through the LMS, by relying completely on his PLE, or a combination of both. In the last case, the LMS can be actually viewed simply as another component of the PLE.

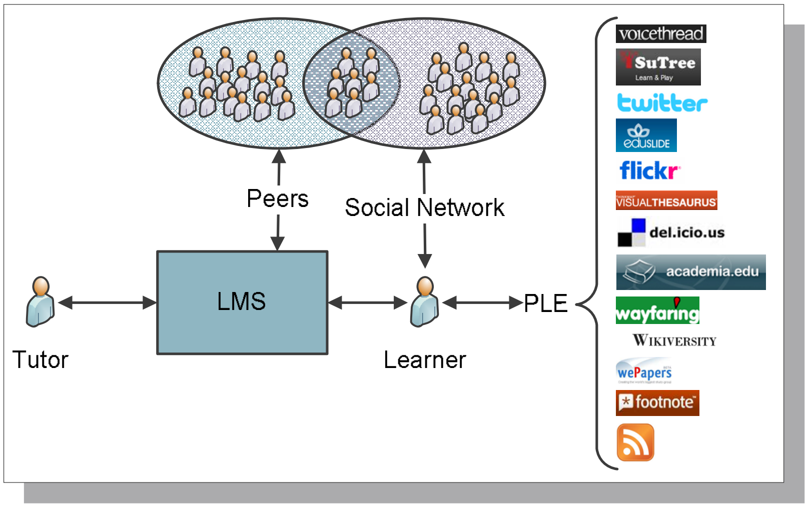


Figure 19: Using LMS and PLE for education

Although this approach does bring together the advantages of LMS and PLE, it still presents some drawbacks, specifically to the learner. First, since there is no formal way to know what external resources the learner is accessing, the tutor cannot validate the content accessed and used by the learner. Hence, the learner may unwillingly access invalid content, which will induce him in error. Consequently the learner will be penalized during the assessment of his knowledge, which may cause conflicts with the tutor. Second, due to the large variety of resources and tools that could be part of a PLE, the discovery of new tools, as well as determining the most suitable tool (depending on the current needs) becomes a daunting task.

3.3 Learner Privacy

And finally, *Privacy*. One of the main advantages of E-learning is its adaptability to the learner's specific needs and preferences (Mbendera, 2010). But in order to do so, the E-learning systems must collect large amounts of information about the learner (Arroyo & Woolf, 2005). Once this information is collected, it could be used for commercial profits, for purposes other than personalization, and could be shared with other E-learning systems or organization, thus violating learner *privacy*, the claim of individuals to determine what information about themselves is known to others, as well as when and how it is used (Westin, 1967). Privacy is nearly absent in current E-learning systems. Only primitive forms of privacy are offered in some platforms, for instance not allowing the tutor access to certain information such as auto-evaluations performed by the learners.

Nonetheless, the tutor has access to virtually all the remaining information including, but not limited to, who the students are, what parts of the course they referred to, how many times and for how long, as well as all the messages in the forums, and all the information about the quizzes and tests the learner took in his course. There are many reasons why a learner would like to keep his information private. We group these reasons under two main categories: *Competitive* and *Personal*. In the **Competitive** context, the learner requires his privacy due to competitive considerations. For example, consider a prominent politician taking a course to increase his knowledge in a certain domain of interest to the electors. Other than for protecting himself from any prejudice from the part of the tutor, he has the right and interest in keeping this fact hidden, and his performance results private, from public knowledge and scrutiny, especially from his opponents. As another example, consider a company that uses E-learning for employee training purposes. If competitors have knowledge of the training and the performance of the employees, it could seriously affect the competitiveness of the company and its reputation, especially if the employees performed poorly. On the other hand, in the **Personal** context, the learner requires his privacy due to personal considerations. For example, he may wish to protect himself from a biased tutor. The bias of the tutor might stem from prejudice or stereotyping, based on a previous encounter with the learner, or even from personal reasons. Another reason a learner would prefer to keep his privacy is the increased pressure and stress due to performance anxiety; a learner might feel more comfortable and relaxed knowing the tutor will not know how he performed in the test. Specifically, learners did reflect, in recent research, a clear preference to privacy in learning systems (Aïmeur, Hage, & Mani Onana, 2007b; Anwar & Greer, 2009; Hage & Aïmeur, 2009).

3.4 Related work

Some of the issues raised in the previous section were already addressed. E-learning Systems (Blackboard, 2004; EduTools, 2006; WebCT, 2003) and other independent E-testing solutions, such as WebQuiz[8], Hot Potatoes[9] and Test Maker[10], counter some of the

[8] Available at http://www.smartlite.it/en2/products/webquiz/index.asp, last accessed on 03/2008

E-testing support limitations. These alternative solutions simplify the exam authoring process, and offer, in some cases, better question type support. Moreover, to ensure content coverage the concept of *question pools* (Paskey, 2001; Rudner, 1998) is employed. A question pool is a collection of items pertaining to a certain subject such that a tutor can now specify a question pool for each subject covered within the exam. Therefore, to ensure content coverage, a certain number of questions are selected from each pool.

Still, these solutions do not address many of the limitations mentioned earlier. For one, they do not advocate implicit knowledge sharing; questions and exams are only available to their respective authors and may be shared explicitly using Import/Export facilities. Moreover, Learning Objects repositories (Porter, Curry, Muirhead, & Galan, 2002) are created and made available, such as Merlot[11] and Lornet[12], and research is performed on methods to help tutors find relevant material within these repositories (Aktas, Pierce, Fox, & Leake, 2004; Jovanovic, Gasevic, & Devedzic, 2006). Nonetheless, the tutor must still access these repositories and search for adequate material.

Moreover, there exist literature on personalizing the learning process in E-learning systems (Dolog, Henze, Nejdl, & Sintek., 2004; Gaudiosi & Boticario, 2003) and there exist methods to personalize tests to learner preferences (Baker, 2001; Desmarais & Pu, 2005), nonetheless, these tools apply the same E-learning platforms methodologies as far as Personalization, that is, questions remain selected at random (EduTools, 2006).

On the other hand, in current E-learning and ITS systems, there are provisions for collaborative learning (Baghaei & Mitrovic, 2006; Israel & Aiken, 2007; Soller, Martínez-Monés, Jermann, & Muehlenbrock, 2005; Takeuchi, Hayashi, Ikeda, & Mizoguchi, 2006). Systems such as SPRITS (Aïmeur, ManiOnana, & Saleman, 2006), Comtella (Vassileva, 2004) and iHelp (Brooks, Panesar, & Greer, 2007) and Papyres (Naak, Hage, & Aïmeur, 2008) promote knowledge and resource sharing. Specifically,

[9] Available at http://hotpot.uvic.ca/, last accessed on 03/2008

[10] Available at http://www.igneon.com/, last accessed on 03/2008

[11] Available at http://www.merlot.org, last accessed on 03/2010

[12] Available at http://lornet.org/, last accessed on 03/2010

system such as iHelp and SPRITS are designed for sharing knowledge, where learners can help each other to solve certain problems and locate suitable help. On the other hand *Comtella* and *Papyres* are used for sharing academic articles. Nonetheless, learners access a wider variety of resource types. Moreover, these approaches do not resolve the problem raised with regards to learners using their PLE in parallel with the LMS.

Finally, although privacy has been successfully incorporated in various domains such a E-commerce (Aïmeur*, et al.*, 2007a; Canny, 2002a, 2002b) it is barely addressed within E-learning systems. Nonetheless, there were concerns raised with regards to security. For example, the consequences of a successful SQL Injection attack (Section 2.3.2.1) on an E-learning system are numerous: for instance, the attacker could have access to the tutor's resources (upcoming exams or homework, grade books, etc.) or the learner's resources (homework, reports, learning resources, etc.). Alternatively, a CSRF attack (Section 2.3.2.2) can be used to manipulate the E-learning system into releasing, modifying or even deleting sensitive information. For instance, a learner could manipulate the E-learning system into modifying the grade book successfully increase his own grades.

There exists literature, such as (Lin, Korba, Yee, Shih, & Lin, 2004; Raitman, Ngo, Augar, & W.Zhou, 2005b), on how to achieve two key security requirements: *confidentiality* and *integrity*, which provide a certain level of privacy. **Integrity** guarantees that the data is not maliciously or accidentally tampered with or modified. **Confidentiality** assures that the data and information is kept secret and private and is disclosed only to the authorized person(s). The confidentiality of the information is considered at two different stages: while it is being transmitted to/from the E-learning system, and when it is stored within the E-learning system. In the first case, the data can be encrypted using Public Key Encryption such that only the appropriate receiver can read the data. In the second case, the use of access control mechanisms (Franz, Wahrig, Boettcher, & Borcea-Pfitzmann, 2006) can be employed to restrict access to the data. Access control cannot totally guarantee the privacy of the learner: first of all, it does not protect against a *super user* with full access privileges. Second, the learner has no control on which information about him is being gathered by the E-learning system. Although

Privacy Policies have been provided for this purpose (Yee & Korba, 2003), they cannot restrict unwanted access to the data.

Chapter 4 : Cadmus

In order to overcome the limitations of current E-testing systems, we offer the framework and the necessary tools to reach our goal. Moreover, to illustrate our approach, we introduce and implement Cadmus, a platform independent, IMS-QTI compliant E-testing environment. Cadmus offers an Authoring Environment where teachers and developers can author and maintain their questions. In order to support implicit knowledge sharing, the questions are stored in a shared Question Base following the IMS-QTI standard, and they may be kept private (accessible only to their author) or made public (accessible to the other authors). Moreover, authors can use their own, or shared questions to create IMS-QTI compliant exams stored in the exam repository. In addition, authors can track the performance of learners, and they can check the questions and exams statistics (how many times the question was answered correctly, what is the overall average of an exam…). On the other hand, Cadmus offers a Learning Environment where students and learners (hereafter called learners) can auto-evaluate their knowledge, take an exam, and track their own progress and development. Figure 20 offers a general overview of the architecture of Cadmus.

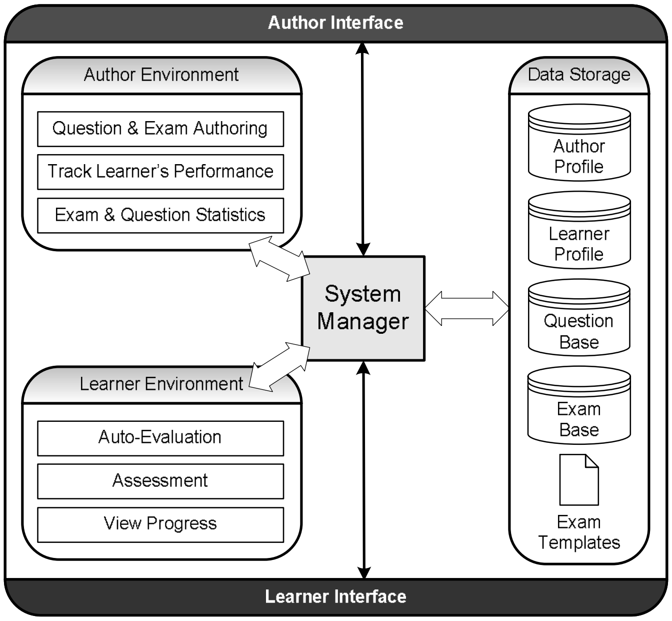


Figure 20: Cadmus Architecture

From the architecture it is clear that Cadmus has two types of components: data components to store the information, and tool components that compose the Author and

Learner environments. The System Manager controls all communication and access to the various components of Cadmus. We first define the data storage components; then we introduce the research axes grouped under the Tutor and Learner Environments.

Question Base

The Question Base stores all the questions created by the authors. The actual question is stored in an external XML file following the IMS-QTI specifications, and the database contains the following information about the question:

Ident: unique question identifier used to uniquely identify the question

Title: contains the title of the question

Language: corresponds to the language of the question

Topic: denotes the topic of the question

Type: denotes the type of the question, i.e.: multiple choice, true/false

Difficulty: specifies the difficulty level of the question, according to possible values: Very Easy, Easy, Intermediate, Difficult, and Very Difficult

Keywords: contains keywords relevant to the question's content

Objective: corresponds to the pedagogical objective of the question: Concept Definition, Concept Application, Concept Generalization, and Concept Mastery

Occurrence: a counter of the number of exams this question appears in

Author: the author of the question

Availability: designates whether the question is available only to the author, to other teachers as well, or even to learners

QTIQuestion: handle to the IMS QTI-compliant XML file where the question and all the relevant information are stored

Exam Base

The Exam Base stores all the exams created by the authors. The actual exam is stored in an external XML file following the IMS-QTI specifications, and the database contains the following information about the exams:

Ident: unique exam identifier used to uniquely identify the exam

Title: contains the title of the exam

Language: corresponds to the language of the exam, i.e. English, French

Topic: denotes the topic of the exam

Type: denotes the type of the exam, i.e.: pop-quiz, mid-term, final …

Difficulty: specifies the difficulty level of the exam, according to possible values: Very Easy, Easy, Intermediate, Difficult, and Very Difficult

Keywords: contains keywords relevant to the exam's content

Objective: corresponds to the pedagogical objective of the exam, detailed by the author in 2-3 lines

Comments: encloses general comments and remarks about the exam

Occurrence: a counter of the number of students who took the exam

Author: the author of the exam

Availability: designates whether the exam is available only to the author, to other teachers, or even to students

QTIExam: handle to the IMS QTI-compliant XML file where the exam and all of the relevant information are stored

Authors Profile

The Authors Profile stores information and data about the authors. This information and data stored at this stage might not be clear, but is necessary for the tools such as the EQRS (Exam Question Recommender System). The user profile contains the following:

Identity: contains information about the author such as his first name, last name and address.

Question Preferences: determines the author's preferred questions criteria when selecting exam questions

Exam Preferences: determines the author's preferred exams criteria

Learner Profile

Identity: contains information about the learner such as his first name, last name, address and student id number

Demographic Profile: refers to demographic characteristics of the learner, such as age, gender, weight, race, ethnic origin, language, etc.

Learning Profile: refers to information such as the learner's qualifications, his learning style, interests, goal and aspirations

Course History: lists the courses the learner has followed in the past, and their respective information such as the learner's activities within the course and his final grade

Current Courses (cc) lists the courses in which the learner is currently registered and those he is attending, as well as the courses' respective information such as learner's activities within the course

At this stage, the learner environment is not of interest, and will not be detailed. The learner profile will be detailed further in 0when dealing with learner privacy.

The authoring environment offers three major functionalities, of which two are straight forward: *tracking* the learner's performance and *viewing* the questions and exams statistics. In order to track the learner's performance, the author requires readily available information on the learner such as the grades on the quizzes and exams, is there any specific topic the learner is failing more than another, etc. Similarly, the questions and exams statistics (such as: what is the number of exams this question appears in, how many learners answered the question, what is the percentage of correct vs. incorrect answers, how many learners took a specific exam, and what is the average) are easily obtained. Thus, Cadmus' main focus is on the questions and exams authoring functionalities providing the tutor with tools to help in the question selection process, while taking into accounts for the tutor's preferences, as well as a module to insure that the exam is conflict free, and that the content coverage constraint is satisfied.

4.1 EQRS - Exam Questions Recommender System

Cadmus aims at offering authors an extensive question base. The more comprehensive the question base is, the harder it is to search for and select questions. The first suggestion that comes to mind is to filter questions according to their content and the needs of the author. A Content-Based filter will help, but might not be enough. For instance, there might be between 50 and 100 questions in the library that satisfy the content requirement, but not all will be rated the same by different authors with different preferences: an author might prefer “multiple choice” to “true and false”, or might prefer questions with a certain level of difficulty, thus the need for an Exam Questions Recommender System (EQRS) (Hage & Aïmeur, 2005). In particular, recommender systems offer the user with an automated recommendation from a large information space (Miller, Konstan, & Riedl, 2004). The popularity of recommender systems has increased over the past years. Today, recommender systems are used in various fields of application, such as restaurants, movies, music, and E-learning. There exist many recommendation techniques differentiated upon the basis of their knowledge sources used to make a recommendation. In (Burke, 2002) 5 recommendation techniques are identified:

Collaborative Recommendation: The recommender system accumulates user ratings of items, identifies users with common ratings, and offers recommendations based on inter-user comparison.

Demographic Recommendation: The recommender system groups users according to their demographic information (such as sex, age and nationality) and recommends accordingly.

Content-Based Recommendation: The recommender system uses the features of the items, and the user’s interest in these features to make a recommendation.

Utility-Based Recommendation: The recommender system uses the feature of an item and to compute its utility for the user and recommends accordingly.

Knowledge-Based Recommendation: The recommender system bases the recommendation of items on inferences about the user’s preferences and needs.

Each recommendation technique has its advantages and limitations, thus the use of hybrid systems that combines multiple techniques to produce the recommendation. There exist several techniques of hybridization:

Weighted: The recommender system groups the scores of various recommendation techniques to produce a single recommendation.

Switching: The recommender system switches between several techniques, depending on the situation, to produce the recommendation.

Mixed: The recommender system offers the recommendation of the several different techniques at the same time.

Feature Combination: The recommender system combines features from the data sources of different techniques and uses the combined features as an input to one single recommendation technique.

Cascade: The recommender system uses one technique to generate a recommendation, and uses a second technique to break any ties.

Feature Augmentation: The recommender systems uses one technique to generate an output, which in turn is used as an input to a second recommendation technique.

Meta-level: The recommender system uses one technique to generate a model, which in turn is used as an input to a second recommendation technique.

EQRS uses a feature-augmentation, hybrid-recommendation approach, where the first level is a Content-Based filter and the second level a Knowledge-Based filter. The Content-Based filter will reduce the search to questions with content pertinent to the author's needs, and the Knowledge-Based filter will sort these questions with regards to the author's preferences, such that the higher ranking questions are the most likely to be chosen by the teacher. Figure 21 illustrates the architecture of the recommender system. We can distinguish two different types of components: Storage components (Question Base and User Profile) and Process Components (Content-Based Filter, Knowledge-Based Filter and Feedback). The Question Base and the Author Profile were detailed earlier in this section.

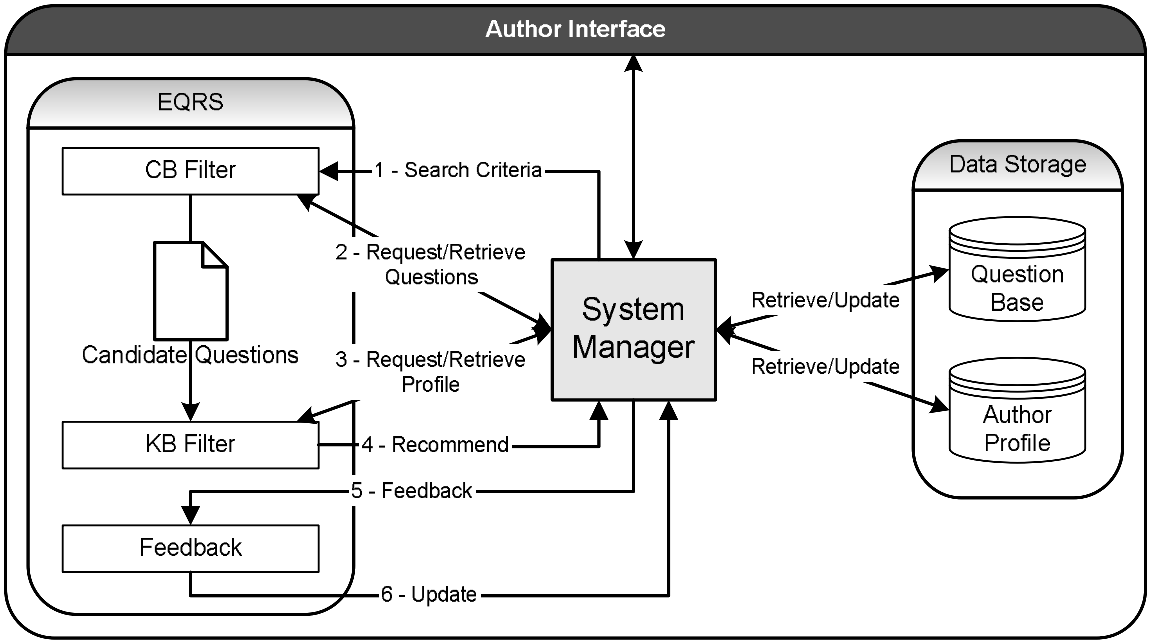


Figure 21: EQRS Architecture

It is important to note that the teacher-specified Type, Occurrence, Difficulty, and Author weights are set manually by the author. These weights represent his criteria preference, i.e. which of the four independent criteria is more important for him, or, in other words, which criteria is more relevant to the search he is performing. The author can select one out of five different weight values, where each is assigned a numerical value (Table 4: Weights Values) used in the utility function explained further in Section 4.1.2. The system-calculated weights infer the author's preferences of the various values each criteria might have. For example, the Type criteria might have one of three different values: True/False (TF), Multiple Choice (MC) or Multiple Selection (MS), thus the system will calculate three different weights: w_{TF}, w_{MC} and w_{MS}. The system keeps track of a counter for each individual weight (i.e. a counter for True/False, a counter for Multiple Selection …), and a counter for the total number of questions selected thus far by the teacher. Each time the teacher selects a new question, the counter for the total number of questions is incremented, and the corresponding individual weight is incremented accordingly, i.e. if the question is a True/False, then the True/False counter is incremented, and w_{TF} = Counter (True/False) / Total number of questions. The value of the individual weights is the percentage of usage, so that if the user selected 100 questions out of which 33 were TF, 59 were MC, and 8 were MS, then $w_{TF} = 0.33$, $w_{MC} = 0.59$, $w_{MS} = 0.08$, and $w_{TF} + w_{MC} + w_{MS} = 1$.

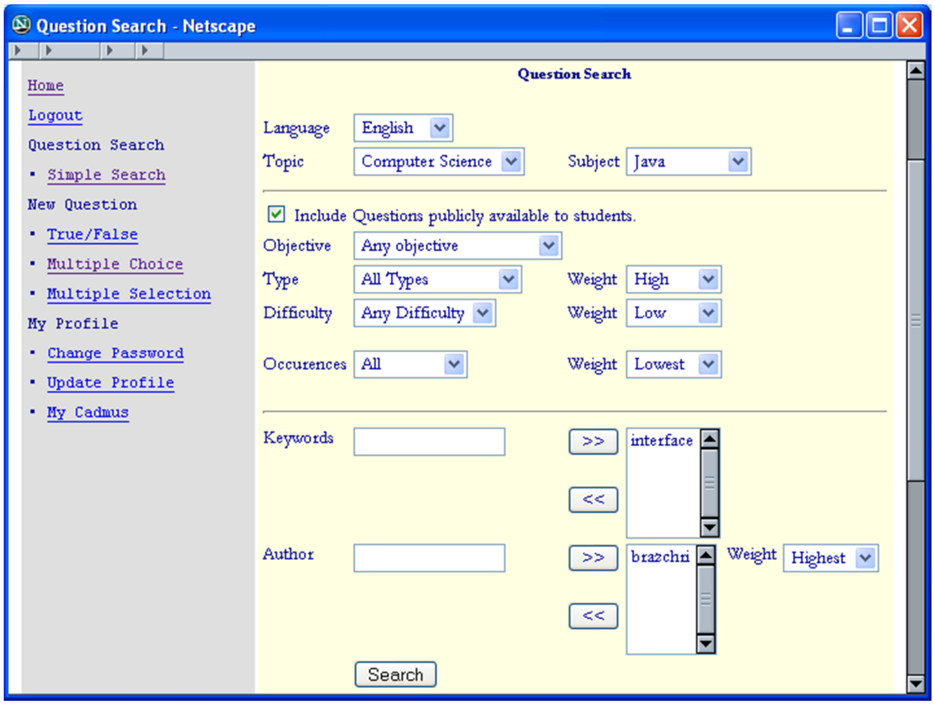
Table 4: Weights Values

Weight	Lowest	Low	Normal	High	Highest
Value	0.25	0.5	1	2	4

4.1.1 Content-Based Filter

When, for the purpose of creating a new exam, the teacher wants to search for questions, he must specify the search criteria for the questions (Figure 22). The search criteria are used by the Content-Based Filter and consist of the following: *Language*, *Topic*, *Subject*, the option of whether or not to include *questions* that are *publicly available to students*, *Objective*, *Type*, *Type Weight* (used by the teacher to specify how important this criteria is to him, compared with other criteria), *Difficulty*, *Difficulty Weight*, *Occurrence*, *Occurrence Weight*, *Keywords* (only the questions with one or more of the specified keywords are retrieved. If left blank, the question's keywords are ignored in the search), *Author* (only the questions of the specified author(s) are retrieved), and *Author Weight*. The teacher must first select the *language* and the *topic* for the question, and has the option to restrict the search to a specific *subject* within the selected topic. Since some questions may be *available to students*, the teacher has the *option to include or omit* these questions from the search. Furthermore, the teacher may restrict the search to a certain question *objective*, question *type*, question *occurrence*, and question *difficulty*.

Moreover, the teacher can narrow the search to questions from one or more *authors*, and can refine his search further by specifying one or more *keywords* that are relevant to the question's content. Finally, the teacher can specify the weight, or the importance of specific criteria (this weight is used by the Knowledge-Based filter). When the user initiates the search, the recommender system will start by collecting the search criteria and weights. Then the search criteria are constructed into an SQL query that is passed to the database. The result of the query is a collection of *candidate questions* whose content is relevant to the teacher's search. The candidate questions and the criteria weights are then used as the input to the Knowledge-Based filter Figure 23 illustrates the result of the search performed in Figure 22. The search result will display the *Title*, *Subject*, *Type*, *Author*, and the *Occurrence* of the question. The teacher has the option to select one or more questions using the checkbox. Furthermore, the teacher can preview the question by clicking on the view button.



Question Search

Language: English
Topic: Computer Science
Subject: Java

☑ Include Questions publicly available to students.

Objective: Any objective
Type: All Types — Weight: High
Difficulty: Any Difficulty — Weight: Low
Occurences: All — Weight: Lowest

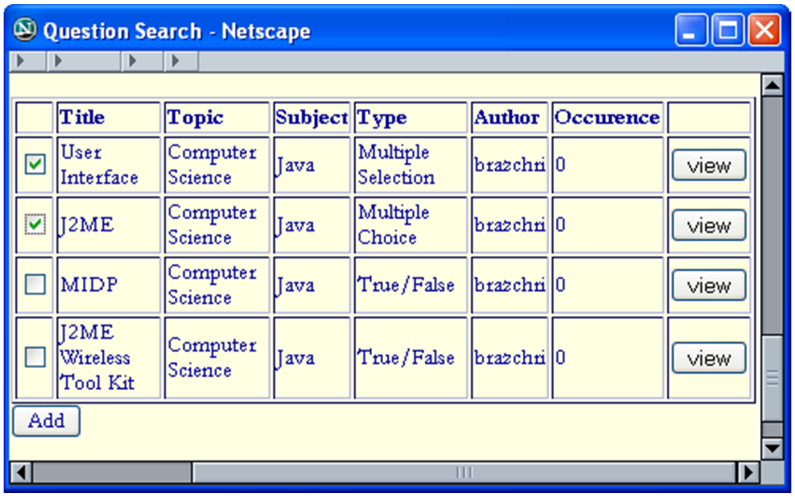
Keywords: interface
Author: brazchri — Weight: Highest

Figure 22: Question Search

4.1.2 Knowledge-Based Filter

The Knowledge-Based Filter takes as input the candidate questions and the criteria weights. The criteria weight is specified by the teacher, and represents the importance of this specific criteria to the user compared to other criteria. Table 4 presents the possible values of the criteria weight and the respective numerical values. The Knowledge-Based filter retrieves the teacher's profile from the Author Profile repository, and uses the utility function to calculate the utility of each candidate questions and the teacher's needs and preferences.

As such, in order to decide which question the teacher will prefer the most, we need to compare several criteria that are unrelated. For instance, how can someone compare the Type of a question with the number of times it appears in exams (the Occurrence)? Since we cannot correlate the different criteria, we left this decision to the teacher: he must select the criteria weight. This weight must either reinforce or undermine the value of the criteria.



	Title	Topic	Subject	Type	Author	Occurence	
☑	User Interface	Computer Science	Java	Multiple Selection	brazchri	0	view
☑	J2ME	Computer Science	Java	Multiple Choice	brazchri	0	view
□	MIDP	Computer Science	Java	True/False	brazchri	0	view
□	J2ME Wireless Tool Kit	Computer Science	Java	True/False	brazchri	0	view

Figure 23: Search Result

The Knowledge-Based recommender uses a heuristic Utility Function, equation (1), to calculate the utility of a question to the teacher's needs and preferences. Consider C={*Type, Difficulty, Occurrence, Author*}. For any $i \in C$, we define $T(i)$ as the set of values the criteria i can have. For instance, T(*Type*)={*Multiple Choice*, *True/False*, *Multiple Selection*}.

$$s(Q) = \sum_{\substack{i \in C \\ j \in T(i)}} \alpha_i \beta_j \qquad (1)$$

The utility value s for some item Q is the sum of the products $\alpha_i \beta_j$, such that α_i is the weight specified by the teacher for the criteria i and β_j is the weight calculated by the recommender system. The multiplication by α_i will either reinforce or undermine the weight of the criteria. Consider the following example to illustrate the utility function: in the search performed in Figure 22, the teacher set α_{Type}= High, $\alpha_{Difficulty}$= Low, $\alpha_{Occurence}$= Lowest and α_{Author}= Highest (numerical values illustrated in Table 4). Table 5 illustrates the values of two different questions $Q1$ and $Q2$, and Table 6 illustrates the individual weights retrieved from the tutor's profile. Note that Table 6 actually contains only a part of the profile, reflecting only the data pertinent to the example.

Table 5: Question Values

	Type	Difficulty	Occurrence	Author
Question1 (Q1)	True/False	Easy	High	Brazchri
Question2 (Q2)	Multiple Choice	Easy	Low	Brazchri

Table 6: Teacher's Profile Values

Criteria	Type		Difficulty	Occurrence		Author
Value	True/False	Multiple Choice	Easy	High	Low	Brazchri
Weight	0.33	0.11	0.5	0.06	0.54	0.15

Calculating the utility function for both questions will give:

(1)
$$s(Q1) = (\alpha_{Type} \times \beta_{True/False}) + (\alpha_{Difficulty} \times \beta_{Easy}) + (\alpha_{Occurence} \times \beta_{High}) + (\alpha_{Author} \times \beta_{\mathrm{Brazchri}})$$
$$s(Q2) = (\alpha_{Type} \times \beta_{MultipleChoice}) + (\alpha_{Difficulty} \times \beta_{Easy}) + (\alpha_{Occurence} \times \beta_{Low}) + (\alpha_{Author} \times \beta_{\mathrm{Brazchri}})$$

(2)
$$s(Q1) = (2 \times 0.33) + (0.5 \times 0.5) + (0.25 \times 0.06) + (4 \times 0.15) = 1.525$$
$$s(Q2) = (2 \times 0.11) + (0.5 \times 0.5) + (0.25 \times 0.54) + (4 \times 0.15) = 1.25$$

Although there exists a big difference between the Occurrences' weights in the favor of $Q2$, $Q1$ will rank higher because the teacher deemed the Type criteria as more important than the Occurrence criteria.

4.1.3 Feedback

The Exam Question Recommender System first retrieves candidate questions using the Content-Based filter, then ranks the candidate questions using the Knowledge-Based filter, and finally displays the questions for the teacher to select from. The author can then select and add the desired questions to the exam. At this stage the exam creation and its effect on the questions and teacher's profile is only simulated; no actual exam is created. The Exam Question Recommender System gathers the feedback from the teacher in two manners: *Explicit* and *Implicit*. Explicit feedback is gathered when the author manually changes the criteria weights, and his profile is updated with the new selected weight. Implicit feedback is gathered when the author selects and adds questions to the exam. Information such as the question type, difficulty, occurrence and author is gathered to update the *system-calculated* individual weights in the teacher's profile (as highlighted earlier).

4.1.4 Testing and Results

The purpose of the Recommender System for Exam Questions is to simplify the task of searching for and selecting questions for exams. The aim of the testing is to determine the performance of the recommendation in helping the teacher select questions. To test the recommender system, we used a database containing about 200 Data Structures and Java questions. The system has a total of 33 different authors/users (Professors and Ph.D. students). For each recommendation and selection, the system recorded the following: *Teacher's Name*, *Date*, *Search Number*, *Questions Recommended*, *Questions Selected*, and *Rank*. The date and the search number enable us to track the performance and quality of the recommendation as the user makes more choices and his profile is developing. The rank of the selected questions is an indication of the accuracy of the Knowledge-Based Filter, the higher the rank of the selected questions, the more accurate is the recommendation of the Knowledge-Based filter. The preliminary results are very encouraging and we are still undergoing further testing. There were 33 registered users testing the system for a total of 89 recommendations, and 366 questions selected and added to exams (some questions were selected more than once). On average 40 questions were recommended after each search. Figure 24 illustrates the Ranking Partition of the selected questions.

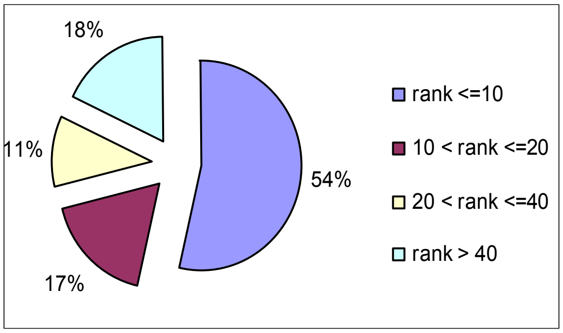


Figure 24: Ranking Partition

Almost 55% of the selected questions were among the top ten recommended questions. Figure 25 illustrates the rank partitioning of the questions selected among the top 10. We notice that the first ranking question is the most selected, while the top five ranked questions constitute about 75% of the selected questions within the top ten ranked by the recommender system. On an average of 40 questions proposed with each search, almost 55% of the selected questions were within the first ten questions recommended by the Exam Question Recommender System, and almost 75% were within the first 20 recommended questions. Thus far, we can conclude that in 75% of the cases, the teacher did not need to browse farther than 20 questions, thereby making it easier for the teacher to search for the required questions for his exam.

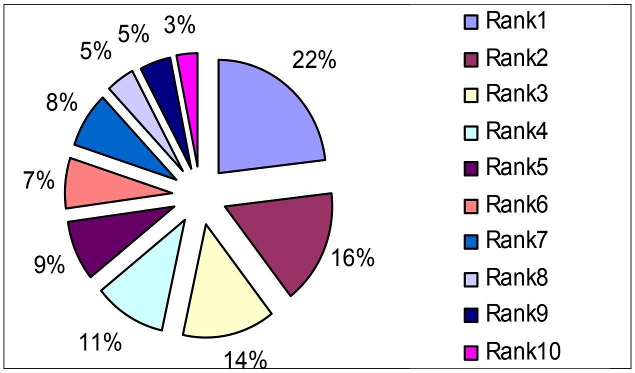


Figure 25: Top Ten ranking

4.2 ICE - Identification of Conflicts in Exams

After selecting the exam question, a normal step would be to check for conflicts. Such conflicts exist in an exam when two or more questions are redundant in content, and/or when a certain question reveals the answer of another question within the same exam.

Such conflicts might be frequent typically when a teacher is using shared questions authored by others, and especially in the automation of the exam creation process and when relying on systems such as the EQRS. As such, we integrate into Cadmus a module for Identification of Conflicts in Exam, or ICE (Hage & Aïmeur, 2006b). In order to detect these conflicts, ICE uses well established IR (Information Retrieval) techniques.

"Information retrieval (IR) deals with the representation, storage, organization of, and access to information items" (Baeza-Yates & Ribeiro-Neto, 1999). The aim of IR is to provide a user with easy access to the information of his interest. IR has branched into fields and applications such as retrieval of spoken information, information filtering, cross-language retrieval, and question answering.

Early IR systems used a complex combination of Boolean ANDs, ORs and NOTs to allow the user to specify his information needs. Boolean systems have several limitations, notably it is hard for a user to create a good search request, and, although Boolean systems typically order the matching documents (i.e. by date, author …), relevance ranking is usually not essential in Boolean systems. IR systems estimate the usefulness of a document to the user and rank them accordingly. IR systems usually assign documents a numeric score, used for ranking purposes. There are several models for this process (Baeza-Yates & Ribeiro-Neto, 1999), (Salton & McGill, 1983); some of the most common models in IR are the *vector space model* and the *probabilistic model* (Singhal, 2001). In the vector space model, a text or a document is represented by a vector of terms (Salton, Wong, & Yang, 1975). A term can be a word and/or a phrase. A term that belongs to the text is assigned a certain numeric value in the text-vector. Most vector based IR systems assign a positive, non-zero value. To assign a numeric score to a document for a certain query, the vector based IR system evaluates the similarity between the query vector and the document vector. Generally, the angle between the two vectors is used as a measure of divergence; the cosine of the angle has the property of being 1.0 for identical vectors and 0.0 for orthogonal vectors.

Probabilistic retrieval was first published by Maron and Kuhns in 1960 (Maron & Kuhns, 1960). Several models have been proposed since. Probabilistic model based IR systems are based on the principal that document ranking should be based on the probability of their relevance to a user's query, often referred to as the Probabilistic Ranking Principle

(PRP). Probabilistic model based IR systems resort to estimate the probability of relevance of documents to a query since true probabilities are not available. This estimation is the key part of the probabilistic model, and it is where various probabilistic models differ.

ICE is based on the vector space model, which relies essentially on a similarity function to determine how identical the two documents are.

4.2.1 Similarity Function

In the vector space model, text or a document is represented by a vector of terms. The Cosine of the angle between two term vectors is used to evaluate the similarity between the respective texts or documents. If the Cosine = 1 then both documents are similar (angle between vectors = 0), and if the Cosine = 0, then the two documents are orthogonal (angle between vectors = 90). Equation (2) highlights the similarity function used to evaluate the similarity (the cosine) between the query vector *q* and the document vector *d*.

$$sim(q,d) = \frac{\sum_{i=1}^{n} w_{i,q} \times w_{i,d}}{\sqrt{\sum_{i=1}^{n} w_{i,q}^2 \times \sum_{i=1}^{n} w_{i,d}^2}} \qquad \textbf{(2)}$$

In equation (2) $w_{i,d}$ represents the weight of the term *i* in the document *d* and $w_{i,q}$ represents the weight of the term *i* in the query *q*. In a regular IR system a query represents what the user is looking for, and the documents represent the search domain. In ICE, the documents are the questions within a specific exam, and the query is one of the exam questions to which ICE is trying to determine if any conflicts exist between this query question and the rest of the questions within that Exam. When a Teacher or an Author is adding a new question to the question base in Cadmus, he is required to specify one or more keywords relating to the content of the question. The terms that compose the query and document vectors are these keywords relating to the content of each question.

Now that the components of the document and query vector are defined, the weight of the keywords (w_{id} and w_{iq}) used in the similarity calculation must be determined.

4.2.2 TF-IDF weighting

The TF-IDF weighting scheme relies on the TF (Term Frequency) and IDF (Inverted Document Frequency) to determine the weight of a keyword in a certain document. The weight $w_{i,j}$ of a keyword i in a document j is calculated using Equation (3).

$$w_{ij} = tf_{ij} \times idf_i \qquad \textbf{(3)}$$

tf_{ij} represents the importance of the term i in the document j, and is calculated using Equation (4) where $mxfreq_j = \underset{i}{MAX}\{freq_{ij}\}$ and $freq_{i,j}$ is the frequency of term i in document j.

$$tf_{ij} = \frac{freq_{ij}}{mxfreq_j} \qquad \textbf{(4)}$$

idf_i represents the discriminating power of the term i and is determined using the formula in Equation (5), where N is the total number of documents, and n_i is the number of documents in which the term i appears in at least once. In this case lg refers to logarithm in base 2.

$$idf_i = \lg\left(\frac{N}{n_i}\right) \qquad \textbf{(5)}$$

4.2.3 ICE Process

Now that the similarity function and keyword weighing scheme is clear, let us put all the building blocks together. Figure 26 illustrates the ICE process. There are three stages in the ICE process, preparation (retrieving exam data and performing the TF-IDF calculation), conflict detection, and conflict reporting. In order to illustrate the ICE process, consider two Java questions Q_1 and Q_2 within the same exam (which has a total of 38 different questions). The respective keywords of the questions are: Q_1 {*order, class, hierarchy, extend, constructor, object, invoke*} and Q_2 {*class, constructor, default*}.

```
Step1: preparation
Retrieve Exam data from the Exam Authoring Environment

$N$ ← Number of Quesitons in Exam
Set $tf$ = 1
Evaluate the $idf$ values for respective keywords using Equation (5)
Calculate $w_{ij} = tf_{ij} \times idf_i$

Step2: conflict detection
For $i$ = 1, … , $N$-1 {
          $q$ ← question number $i$

For $j$ = $i$+1, … , to $N$ {
                    $d$ ← question number $j$
                    $S$ ← sim ($q$,$d$)
                    If $S$ > $Threshold$ then Mark conflict
                    }
          }

Step 3: conflict reporting
Report marked conflicts
```

Figure 26: ICE process

TF-IDF calculation

The first step of the ICE process is to prepare the TF-IDF values for the keywords. First, since exam questions are usually short, most keywords will appear only once in the question. Thus ICE assumes the TF for all the keywords to be always 1. Furthermore, during the Exam creation process, the Exam Authoring Environment keeps track of a counter for each of the various question's keywords; incrementing or decrementing the counter each time a question is added to, or removed from the exam. When the exam creation process is done, ICE iterates over the value of the keyword's counter and applies equation (5) to compute the respective IDF values, where N is the total number of questions in the exam (N = 38 in the case of this example), and n_i is the number of questions in which the keyword i appears in at least once. Finally, ICE applies equation (3) to evaluate the weights of the keywords. Table 7 illustrates the keywords of Q_1, and Q_2 their count (number of questions they appear in within the exam), TF and IDF value.

Table 7: Keywords summary

i	Keyword	n_i	TF	IDF = lg (N/ n_i)
1	order	3	1	lg(38/3) = 2.5390
2	class	10	1	lg(38/10) = 1.3350
3	hierarchy	2	1	lg(38/2) = 2.9444
4	extend	2	1	lg(38/2) = 2.9444
5	constructor	5	1	lg(38/5) = 2.0281
6	object	2	1	lg(38/2) = 2.9444
7	invoke	2	1	lg(38/2) = 2.9444
8	default	1	1	lg(38/1) = 3.6376

Conflict Detection

In order to detect conflicts within an Exam, ICE iterates the query vector (q_i) on the questions of the Exam, such that $i \leftarrow 1 , \dots , N\text{-}1$ (N is the total number of questions in the exam). Then, for each q_i, ICE iterates the document vector, d_j, on the remaining questions, where $j \leftarrow i\text{+}1 , \dots , N$. At each iteration (i,j), ICE calculates $S1$ = sim(q_i,d_j). If $S1$ is greater than or equal to the threshold T, then ICE reports Q_i and Q_j as redundant questions. The threshold T is set at 0.45. The value of T was determined through testing, such that the two conflicting questions with the smallest similarity are detected. Furthermore, at the same iteration (i,j), ICE will automatically extract the keywords of the correct answer(s) of Q_i, then adds these keywords to q_i, resulting in a new query q_i'. ICE then computes $S2$ = sim(q_i' ,d_j). If S2 is greater than or equal to the threshold T, then ICE reports the conflict between Q_i and Q_j: Q_j reveals the answer to Q_i. Moreover, at the same iteration (*i,j*), ICE will also automatically extract the keywords of the correct answer(s) of Q_j, then adds these keywords to d_j, resulting in a new query d'_j. ICE then computes $S3$ = sim(q_i, d'_j). If $S3$ is greater than or equal to the threshold T, then ICE reports the conflict between Q_i and Q_j: Q_i reveals the answer to Q_j. For example, applying the similarity function equation (2) on Q_1, Q_2 (Table 7) to calculate $S1$, results with the following:

$$S1 = sim(Q_1, Q_2) = \frac{\sum_{i=1}^{n} w_{i,Q1} \times w_{i,Q2}}{\sqrt{\sum_{i=1}^{n} w_{i,Q1}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,Q2}^2}}$$

$$\begin{aligned}
&\sum_{i=1}^{n} w_{i,Q1} \times w_{i,Q2} \\
&= (w_{order,Q1} \times w_{order,Q2}) + (w_{class,Q1} \times w_{class,Q2}) \\
&+ (w_{hierarchy,Q1} \times w_{hierarchy,Q2}) + (w_{extend,Q1} \times w_{extend,Q2}) \\
&+ (w_{constructor,Q1} \times w_{constructor,Q2}) + (w_{object,Q1} \times w_{object,Q2}) \\
&+ (w_{invoke,Q1} \times w_{invoke,Q2}) + (w_{default,Q1} \times w_{default,Q2}) \\
&= ((tf_{order,Q1} \times idf_{order}) \times (tf_{order,Q2} \times idf_{order})) \\
&+ ((tf_{class,Q1} \times idf_{class}) \times (tf_{class,Q2} \times idf_{class})) \\
&+ ((tf_{hierarchy,Q1} \times idf_{hierarchy}) \times (tf_{hierarchy,Q2} \times idf_{hierarchy})) \\
&+ ((tf_{extend,Q1} \times idf_{extend}) \times (tf_{extend,Q2} \times idf_{extend})) \\
&+ ((tf_{constructor,Q1} \times idf_{constructor}) \times (tf_{constructor,Q2} \times idf_{constructor})) \\
&+ ((tf_{object,Q1} \times idf_{object}) \times (tf_{object,Q2} \times idf_{object})) \\
&+ ((tf_{invoke,Q1} \times idf_{invoke}) \times (tf_{invoke,Q2} \times idf_{invoke})) \\
&+ ((tf_{default,Q1} \times idf_{default}) \times (tf_{default,Q2} \times idf_{default}))
\end{aligned}$$

$$
\begin{aligned}
&=((1\times 2.5390)\times(0\times 2.5390))\\
&+((1\times 1.3350)\times(1\times 1.3350))\\
&+((1\times 2.9444)\times(0\times 2.9444))\\
&+((1\times 2.9444)\times(0\times 2.9444))\\
&+((1\times 2.0281)\times(1\times 2.0281))\\
&+((1\times 2.9444)\times(0\times 2.9444))\\
&+((1\times 2.9444)\times(0\times 2.9444))\\
&+((0\times 3.6376)\times(1\times 3.6376))\\
&=0+1.7822+0+0+4.1134+0+0+0\\
&=5.8956
\end{aligned}
$$

On the other hand,

$$\sqrt{(\sum_{i=1}^{n} w_{i,Q1}^2)} = \sqrt{\begin{aligned}&(w_{order,Q1}^2 + w_{class,Q1}^2 + w_{hierarchy,Q1}^2 + w_{extend,Q1}^2 +\\ &w_{constructor,Q1}^2 + w_{object,Q1}^2 + w_{invoke,Q1}^2 + w_{default,Q1}^2)\end{aligned}}$$

$$=6.8572$$

and

$$\sqrt{(\sum_{i=1}^{n} w_{i,Q2}^2)} = \sqrt{\begin{aligned}&(w_{order,Q2}^2 + w_{class,Q2}^2 + w_{hierarchy,Q2}^2 + w_{extend,Q2}^2 +\\ &w_{constructor,Q2}^2 + w_{object,Q2}^2 + w_{invoke,Q2}^2 + w_{default,Q2}^2)\end{aligned}}$$

$$=4.3735$$

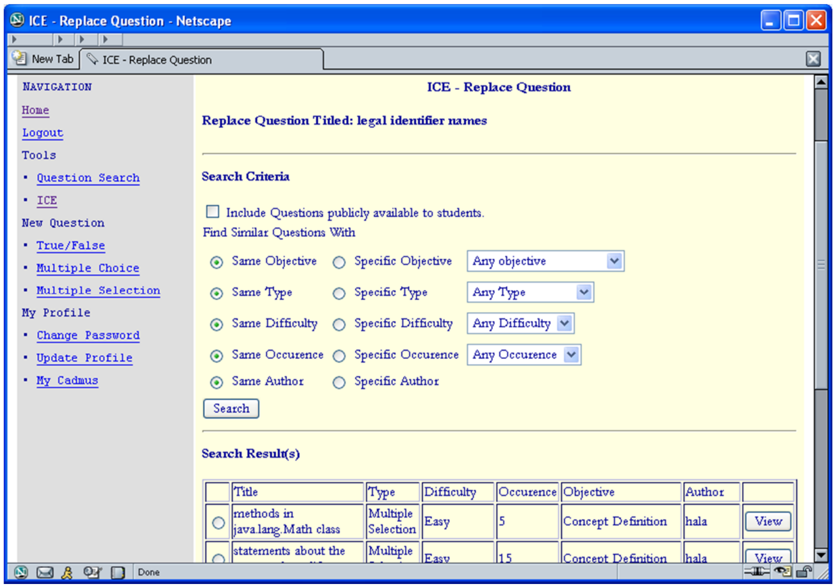
Thus

$$S1 = sim(Q_1, Q_2) = \frac{5.8956}{6.8572\times 4.3735} = 0.1966$$

Since *S1* < *T* then there is no conflict between Q_1 and Q_2.

Conflict Reporting

When ICE detects a conflict between two questions, that conflict is reported. Both questions are specified with the option to view or replace each of the questions. The view question option pops up a window with the question and its answers. To replace the question, the user is presented with the search option highlighted in Figure 27. The user can search for questions with the same criteria as the question to be replaced. Furthermore, the user can change one or more criteria to search for replacement questions. In the first case, the search for the replacement questions is done through a simple content based filter. All the questions with the same criteria as the question to be replaced are retrieved.



ICE - Replace Question

Replace Question Titled: legal identifier names

Search Criteria

□ Include Questions publicly available to students.

Find Similar Questions With

● Same Objective ○ Specific Objective Any objective

● Same Type ○ Specific Type Any Type

● Same Difficulty ○ Specific Difficulty Any Difficulty

● Same Occurence ○ Specific Occurence Any Occurence

● Same Author ○ Specific Author

Search

Search Result(s)

	Title	Type	Difficulty	Occurence	Objective	Author	
○	methods in java.lang.Math class	Multiple Selection	Easy	5	Concept Definition	hala	View
○	statements about the	Multiple	Easy	15	Concept Definition	hala	View

Figure 27: ICE – Replace Question

As a first attempt, ICE will try to retrieve all the questions with same criteria as Q_r (the question to be replaced) and none of its keywords. If no replacement questions were found, ICE will attempt a new search for questions with the same criteria and some of Q_r's keywords. In order to know which keywords to allow in the replacement questions, ICE first retrieves the keywords of Q_r, then ICE selects the prohibited keywords with the highest weight, such that if a replacement question had all Q_r's remaining keywords, the similarity will remain less than the threshold T. ICE will perform the new search for all the replacement questions with the same criteria as Q_r and none of the prohibited keywords. In the second case, when one or more search criteria is specified by the user, the search for replacement questions is performed using the EQRS techniques.

4.2.4 Testing and Results

ICE was tested on a questions bank of 200 Data Structures and Java questions. The test generates an exam by selecting between 10 and 40 questions randomly. After the creation of the random exam, ICE will detect the conflicts. There were a total of 204 randomly created exams with conflicts. The random exams had an average of 28 questions. There were no undetected conflicts; and a total of 512 reported conflicts. Since the same conflict between two questions might appear in several exams, recurring conflicts were grouped into conflict case. Grouping the recurring conflicts into cases resulted in a total of 93 different conflict cases, out of which 77 (83%) were true conflicts and 16 (17%) were not actual conflicts. These results are illustrated in Figure 28. Most of the invalid conflicts reported are due to keywords selection and weighing. Different questions with very similar keywords, such that the difference in the context of the questions is defined by only one of the keywords have a similarity greater than the threshold. Increasing the value of the threshold will result in true conflicts being undetected. Nonetheless, testing proved that setting *T* to 0.458 (*T* was 0.45 originally) increased the accuracy of conflict reporting, although now, there are undetected valid conflicts (Figure 29). A further increase in the value of *T* reduced the number of invalid conflicts reported, but did not ameliorate the accuracy since more true conflicts were passing undetected. Table 8 summarizes the results of the tests.

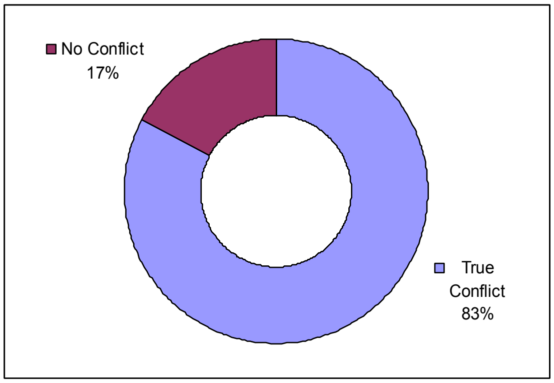


Figure 28: Preliminary Results

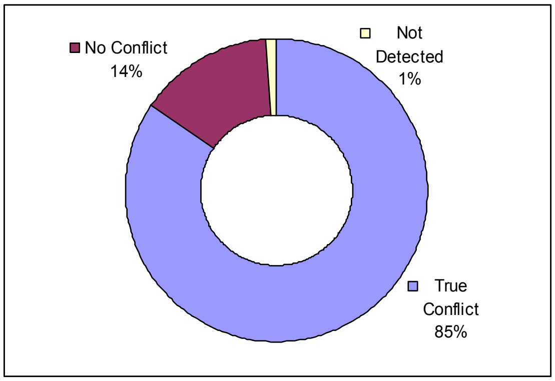


Figure 29: Results after increasing T

Table 8: Results Summary

	Total	No Conflict		True Conflict		Undetected Conflict	
Preliminary Results	512	111	21.68%	401	78.32%	0	0%
Refined Results	93	16	17.20%	77	82.80%	0	0%
Results T = 0.458	90	13	14.44%	76	84.44%	1	1.11%

Although ICE was tested only on questions with a limited topic, the accuracy of conflict detection will not suffer with other subjects since ICE relies mainly on the keywords specified by the author of the question. Initial testing on sample Artificial Intelligence and Databases questions have resulted with similar, high accuracy conflict detection.

Furthermore, testing on the available question base has revealed that whenever a question Q_i is detected to reveal the answer of a question Q_j, then both questions are similar enough in content to be detected by ICE as redundant questions. Although it is not a complete surprise (since it is logical to assume that for a certain question to reveal the answer of another question it should be similar in context), further testing on a bigger questions base, and searching for particular cases can help determine the need of testing for such conflicts (if Q_i reveals the answer of Q_j).

4.3 Topic Tree: increasing the accuracy of ICE

Knowledge sharing is very important; it helps tutors share their experience and knowledge in order to deliver better education. Moreover, sharing helps reduce the time and cost of redeveloping learning material which was already developed. Besides, in the case of E-testing, item statistics are very important for personalization (for example in Item Response Theory), therefore it is advantageous to reuse an item with well established statistics instead of just recreating it. An additional issue is *content coverage*. In the context of an exam, content coverage refers to the sufficient coverage of various subjects included in the exam. This is particularly important since a tutor needs to test all the learners' knowledge on various aspects of a certain subject. To ensure content coverage question pools (Paskey, 2001; Rudner, 1998) are regularly used. A question pool is a collection of questions pertaining to a specific subject such that a tutor can now specify a question pool for each subject covered within the exam. Therefore, to ensure content coverage, a certain number of questions are selected from each pool. Since populating these pools is a demanding task, question pools have a tendency to practically remain constant in size and content.

With the emergence of E-learning standards and specifications, such as IMS QTI (IMS Question and Test Interoperability), it has become easier to share E-learning and E-testing material and LOs (Learning Objects) repositories are created and made available. Repositories such as Schoolnet.ca, Ariadne-eu.org and Merlot.org, provide various LOs and research is performed on methods to help tutors find relevant material within these repositories (Ma, 2005; Tsai, Chiu, Lee, & Wang, 2006). Indeed, although IMS QTI offers item metadata, such as to specify the item's *topic*, to the best of our knowledge, there is no *standardization* of the values such fields can hold, thus making it harder to locate suitable material when sharing knowledge. Indeed, how is a tutor supposed to locate suitable material in the context of E-testing if he cannot, at least, accurately specify the subject of the items he is searching for? In this work, we introduce the Topic Tree. The topic tree offers a *hierarchy* of topics, or taxonomy, from which to choose the topic of an item. This standardization process, by increasing the accuracy of IR (Information Retrieval) systems, makes it easier to locate suitable E-testing material, thus simplifying the process of sharing knowledge and populating question pools. Moreover, as a proof of concept, we integrate and test the impact of the topic tree in Cadmus, an IMS QTI compliant assessment platform.

4.3.1 Related Work and Preliminaries

When searching existing LOs repository for resources, there is a certain organization of the LOs into a hierarchy of topics, for example: "*All > Science and Technology > Computer Science > Artificial Intelligence*" in Merlot. Moreover, the ACM offers the CCS, Computing Classification System (ACM, 1998), a classification system for computer science which is used to classify articles. The ACM classification has 11 first level nodes and is only 3 levels deep. Figure 30 highlights the hierarchy of first level node "Data".

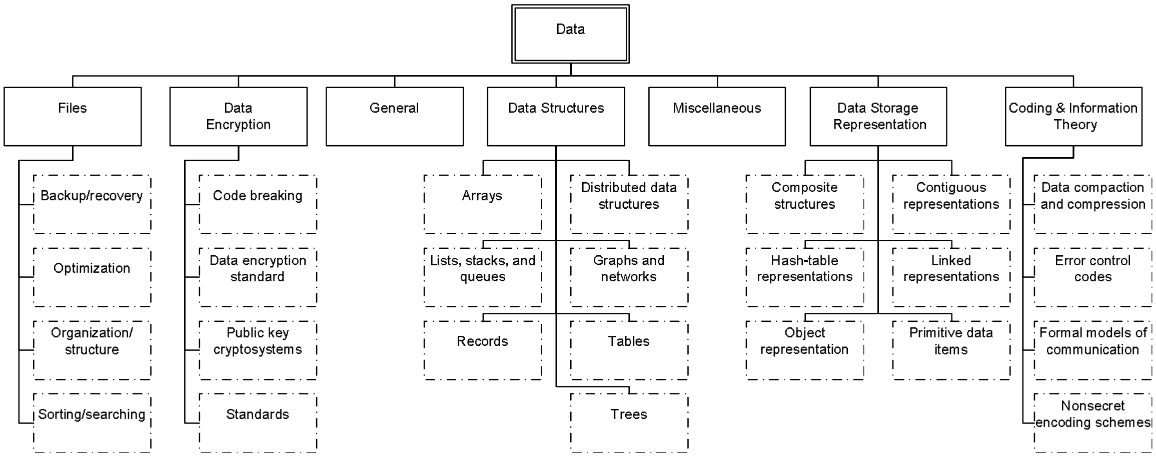


Figure 30: ACM CCS – partial

Although such a classification is suitable for articles and papers, it is clear that it is not sufficient in the case of LOs and test items, where the modules are more specialized and a more accurate classification is required. Nonetheless, this classification is an invaluable corner stone, and the topic tree that we propose is partially inspired by the ACM CCS.

4.3.2 Topic Tree

In IR (Information Retrieval), hierarchical structures are used to help a user locate information of value. Usually, when searching for information in a hierarchy, the user has some idea about the topic this information belongs to. In this context, the user's knowledge is used to browse the hierarchy, in a top-down manner, until the level that best describes his information need is reached, and the user will check the documents. If none of the documents contain the required information, the user may move up in the hierarchy to a more general topic. The topic tree is a taxonomy, or a hierarchy of topics that we are constructing. Based on the ACM CCS, several data structures books, including (Standish, 1998) and (Weiss, 1999) , and with the help of professors from the computer science department of the University of Montreal, we constructed a topic tree for the data structures domain. Due to space limitations, we will use a *cropped tree* (Figure 31) for our illustration.

Since each test question relates to a certain subject or topic, instead or just relying on *distinct* topics specified by either the E-testing system or the author, this topic would be selected from within the topic tree. For example, a question on Binary Trees will have a topic in the form of: *Data Structures* > *Tree*: *Binary Tree*. The advantage of such an approach is two fold: first, the item's *standardized* topic can be easily specified by the tutor such that it is easier to locate suitable E-testing material. Indeed, hierarchical structures enable a divide-and-conquer approach to IR, resulting in higher efficiency and accuracy (Gelbukh, Sidorov, & Guzman-Arenas, 1999). In particular, existing IR research suggests that rather than issuing general search queries, users would rather browse hierarchical catalogs and issuing specific queries (Tao & Ogihara, 2005). It is important to note though that a *proper classification* is imperative for a *better accuracy*. For this purpose, the selection of an item's topic can be selected by the item's author, and afterwards verified, or even specified automatically, using IR techniques such as proposed in (Bade, Hullermeier, & Nurnberger, 2006). The second advantage of our approach is in that the questions pool creation process could be modified such that a tutor would simply link a question pool to one or several levels within the topic tree. For instance, a pool on trees in general (as in a comprehensive exam) would be linked to *Data Structures* > *Tree*. Alternatively, the tutor would be able to link the question pool more selectively such as to *Splay Tree* and *AVL Tree* solely. It is important to note that the appellation topic tree is somewhat misleading: since some topics can be shared by several disciplines there will be more than one *path* from the *Root* to the appropriate node. Thus, the actual structure of the topic tree is more of a DAG (Directed Acyclic Graph). Nonetheless, this "tree" representation/appellation makes it easier to represent and grasp the concept.

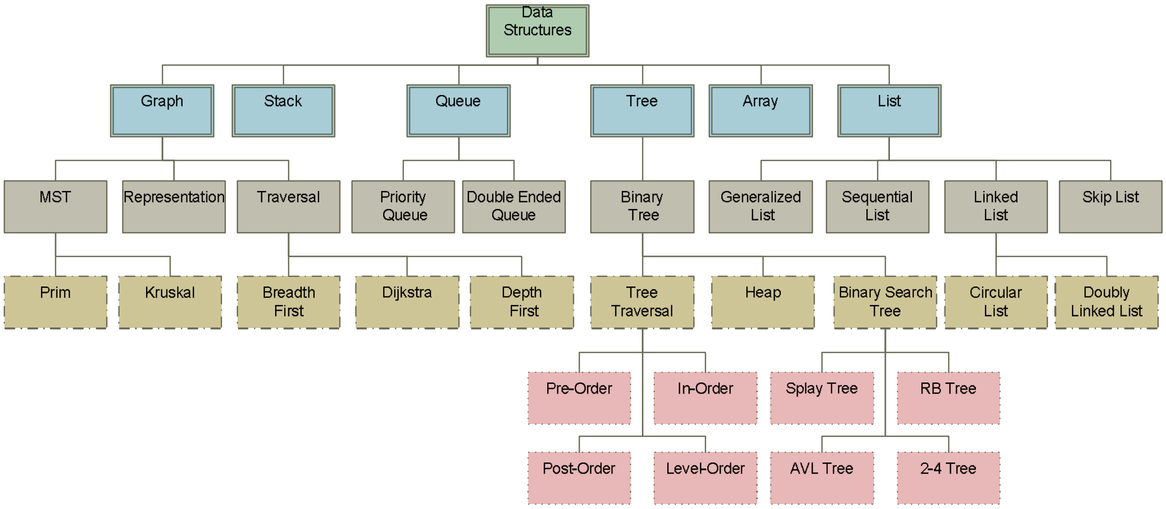


Figure 31: Topic Tree Sample

Generally, tutors need to assess various aspects of the learners' knowledge on a certain concept. For instance, tutors need to ascertain that a learner can clearly *define* a certain concept, or if the learner has *mastered* the concept in question. With this in mind, other than the obvious case where the tutor requires the learner to clearly define a certain concept, questions might contain references or comparisons to other concepts. For example, consider the following data structures question: "describe how to implement the stack ADT using two queues". Referring back to Figure 2, what would be the topic of the previous question? It should be either *stack*, or *queue*, but which one should be chosen? In order to resolve such cases, we propose combining the pedagogical objective of the item along with its topic. What follows is a description of the possible pedagogical objective of test items:

- **Concept Definition:** a question on properly defining a certain concept
- **Concept Application:** a question on applying a concept to resolve a certain problem
- **Concept Generalization:** a question on adapting or modifying a concept to resolve a certain situation
- **Concept Mastery:** a question to test the knowledge of several aspects of a concept

Thus, in the case of the previous question (on stacks and queues), since the solution to the question involves *adapting* the *queue* ADT to *implement/resolve* the *stack* situation, the topic is *queue*, and the pedagogical objective is *concept generalization*. Another similar question would be:

In order to implement a queue using a heap you would:

a) Use a heap with a pointer/index to the first and last element inserted into the heap
b) Use a heap with a counter as the key of the elements
c) There is no solution to this problem
d) None of the above

This question would have as *heap* as a topic, and again *concept generalization* as objective. On the other hand, a question which involves more than one “*related*” concept within the same sub-tree would use as topic the root (or the closest common ancestor) of the sub-tree, and as pedagogical objective *concept mastery*. For example, the following question would have a *binary tree* for topic, and *concept mastery* as objective:

Which of the following would traverse the nodes of a binary search tree in ascending order?

a) Pre-Order
b) In-Order
c) Post-Order
d) Level-Order
e) None of the above

Thus, an item that deals with one concept would have as an objective either concept definition or concept application, and an item that includes more than one concept would have as an objective either concept generalization or concept mastery.

4.3.3 Testing and Result

Prior to the inclusion of the topic tree into Cadmus, the topic of the items was based on a two level hierarchy in the form of: *Programming Languages > Java*. After introducing the topic tree, we *re-evaluated* the performance of both the EQRS and ICE.

ICE was tested on a repository of 200 questions. The test generated an exam by randomly selecting questions on the same subject. The number of questions within each exam was selected randomly between 10 and 40. After the creation of an exam, ICE detects and reports any existing conflicts. Initially, a total of 204 randomly created exams contained conflicts. Recurring conflicts (i.e. conflicts between the same two questions appearing in different exams) were grouped together, and the threshold T, used to determine if two questions were conflicting or not, was selected such that no existing conflicts may slip away undetected. There were a total of 93 different conflicts reported, out of which only 82.8% were true conflicts and 17.2% were not actual conflicts. Figure 32 illustrates the results of the initial tests of ICE.

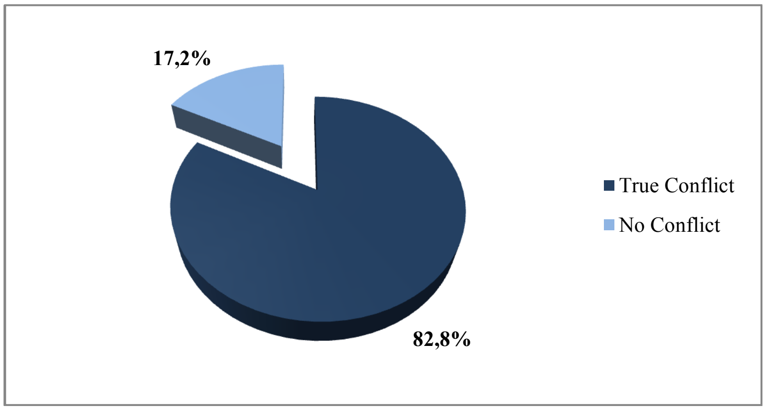


Figure 32: ICE Initial Results

Since ICE used keywords assigned to various questions to detect conflicts, most of the invalid conflicts reported were due to this assignment of these keywords. For example, consider the following two questions where ICE reported a conflict that did not exist:

Question1: What is the result of an In-Order traversal of the following binary tree?
Keywords: result, In-Order, traversal, binary tree

Question2: What is the result of a Post-Order traversal of the following binary tree?
Keywords: result, Post-Order, traversal, binary tree

Out of the four keywords specified for both questions, only one, in each case, made the difference as far as the question's content is concerned. Therefore, to increase the accuracy of conflict reporting, the topics of both questions, which are now selected from the topic tree, were compared. If the two conflicting questions have the same topic, the conflict was reported, otherwise it was ignored. Thus, in the case of questions 1 and 2, although ICE initially detected a conflict, it was dismissed since the topics of the questions are not the same: "*Data Structure > Tree > Binary Tree > Tree Traversal > In Order*" and "*Data Structure > Tree > Binary Tree > Tree Traversal > Post Order*" respectively. Introducing the topic tree lead to an increase of 6.5% in the accuracy of ICE (Figure 33), where now 89.3% of the reported conflicts were *true conflicts* and only 10.7% of the conflicts reported were actually *not conflicts*. Prior to the inclusion of the topic tree, the comparison of the questions topic was ineffective since both had the same topic.

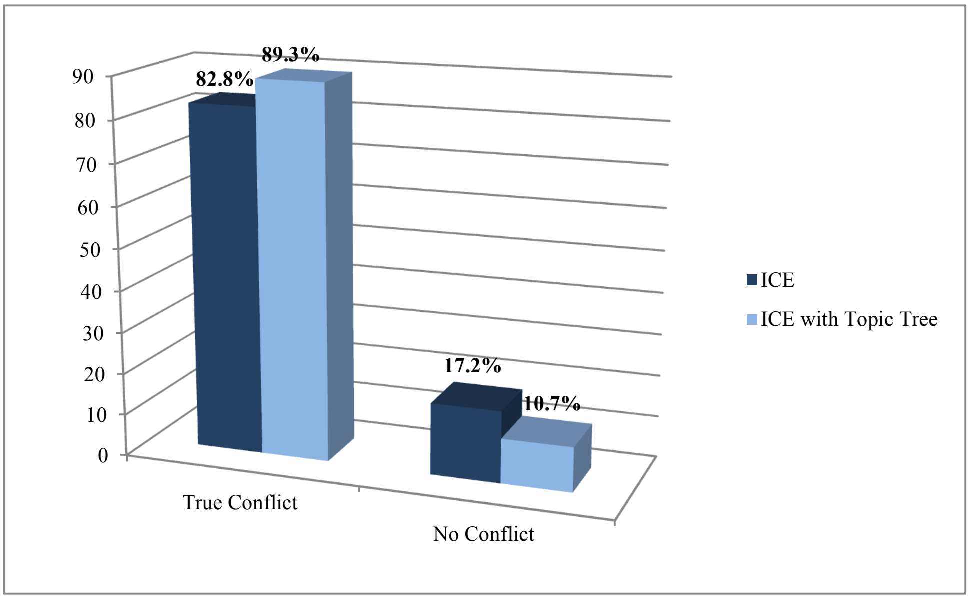


Figure 33: ICE Accuracy Comparison

This increase in accuracy is important, it does confirm that using the topic tree will help increase the accuracy of IR systems designed to locate E-testing material. ICE was accurate at almost 90% in detecting similar questions. Thus, using the same technique to locate E-testing resources with respect to a certain query would perform similarly. Indeed, since the EQRS relies partially on the content of the question, including its topic, to perform the recommendation, similarly, the accuracy of the recommendation increased. Unfortunately, at the current time, we did not get the chance to *properly* re-evaluate EQRS after the inclusion of the topic tree. In fact, the EQRS relies on the user's profile, as well as the question's content, to recommend questions, thus, the result of re-evaluating the performance of EQRS, without undergoing actual testing with users, will be imprecise.

Chapter 5 : SHAREK

Learners regularly access and use learning content and tools other than those defined by the tutor in the classroom. Looking back at the current state of combining E-learning platforms and Personal Learning Environments (Figure 19), we raised two concerns: first the tutor cannot *control what learners are accessing* through their personal PLE, and second, *locating the components* for an effective PLE can easily become a difficult task (Hage & Aïmeur, 2010a). Hence, in order to address both drawbacks, we propose an approach inspired by social bookmarking, consisting of adding support for a PLE within the LMS (Figure 34).

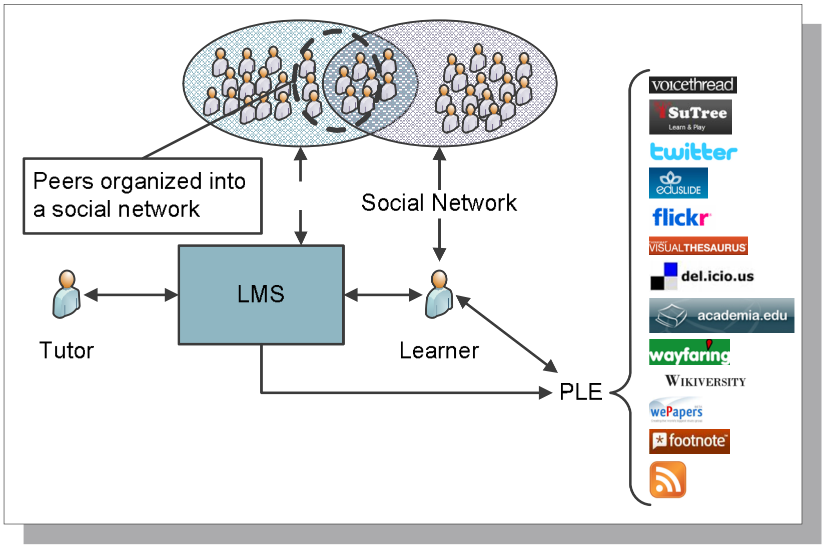


Figure 34: Combining LMS and PLE

Specifically, learners can add learning resources into the LMS and could *attach* them to a course, or a lecture within the course. As such, learners can access the PLE components by linking to them from within the course or lecture. For instance, a learner who relies on *VisualThesaurus* for an English course would add it as a resource to the course. Consequently, whenever the learner accesses this course, he can directly link to VisualThesaurus without having to search for it through a large set of bookmarks. Such an approach has many advantages. **First**, it enables learners to *organize* their PLE components with respect to the courses they are used in. **Second**, since these components are organized within the LMS, it enables the tutor to *supervise*, to some extent, the use of these components. Specifically, although the tutor cannot control or supervise the activities performed within the PLE, he can still know what components the learners are using in their PLEs. Hence, the tutor can take some actions accordingly: for instance, if he realizes that some learners are using Wikipedia for instance, he can make sure to advise them not to take that information for granted, since the content of wikis might be, in some cases, unreliable or biased. In short, the tutor can discourage and/or caution when *questionable* components are used in a PLE, and he can encourage and/or promote *valid* components. **Third**, this setup provides an ideal setting for the *discovery* of new components, whether for the learners or the tutor. Indeed, one can easily exploit this setup to allow learners to share their knowledge of learning resources. To the best of our knowledge, there are no existing mechanisms in E-learning systems which offer this setup, *efficiently* harnessing and taking advantage of learners' knowledge and the resources they locate and use. Supported by learner centered education, and the learners' access to a variety of learning resources, and inspired by Web2.0, we propose our system SHAREK (SHAring REsources and Knowledge) (Hage & Aïmeur, 2008b). SHAREK's primary goals are to *harness* the collective intelligence and knowledge of learners obtained through accessing various learning resources, and *sharing* this knowledge and resources with other learners (Hage & Aïmeur, 2008a). Specifically, each time a learner accesses a course's learning content (specified by the tutor), he would be able to also access additional learning resources located and used by other learners and classmates as part of their PLE. Within an E-learning system, SHAREK will be the components managing the link between the LMS and the learner's PLE as highlighted in Figure 34. On the other hand, Figure 35 illustrates the positioning of SHAREK within an ITS (Intelligent Tutoring System) architecture based on the IEEE LTSA (Learning Technology Systems Architecture) (IEEE Learning Technology Standards Committee, 2003).

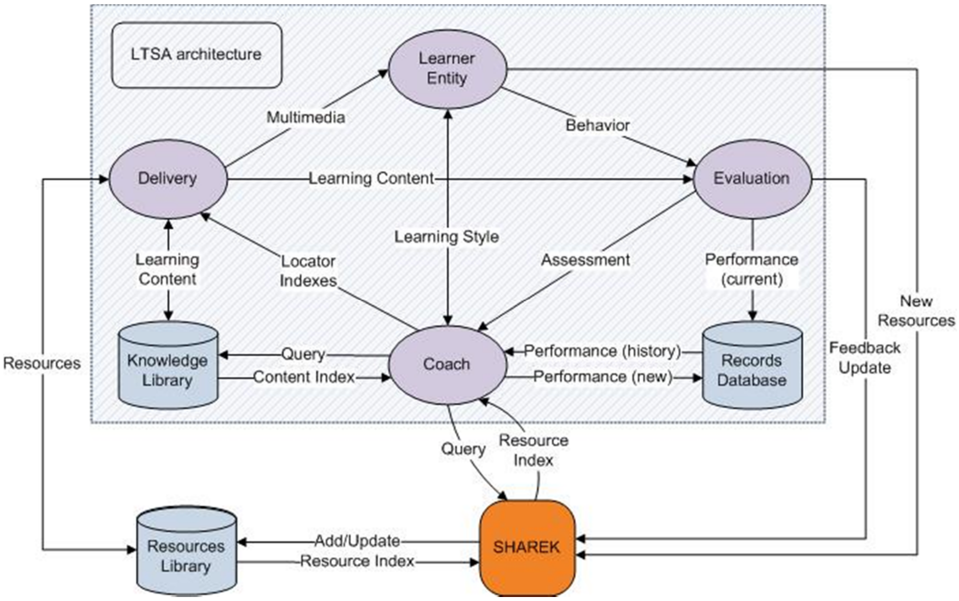


Figure 35: SHAREK within an ITS

Originally within the LTSA architecture, the Coach reviews a set of information, such as performance history and objectives, and searches, via the *Query* request, the *Learning Resources* for the proper learning content. In this case, the coach would send two Query requests, one to the Learning Resources for the learning content, and another to SHAREK for the learner added learning resources. SHAREK will process the coach's query and send back the appropriate resources indexes, which in turn are sent, along with the locator index to *Delivery*. *Delivery* will retrieve both the content and the resources and send them as multimedia to the learner. *Evaluation* will send any behavior feedback with regard to the resources (tags, rates, etc.) back to SHAREK which updates the *Resources Library*. Moreover, SHAREK handles the learners' requests to add new resources.

The following figure highlights the SHAREK's process. The learners have access to a multitude of resources, and can perform different actions within SHAREK. Depending on the action, SHAREK will send some feedback (events information through RSS, or the result of a search /recommendation) to the learners or the tutor. The following sections detail these processes further.

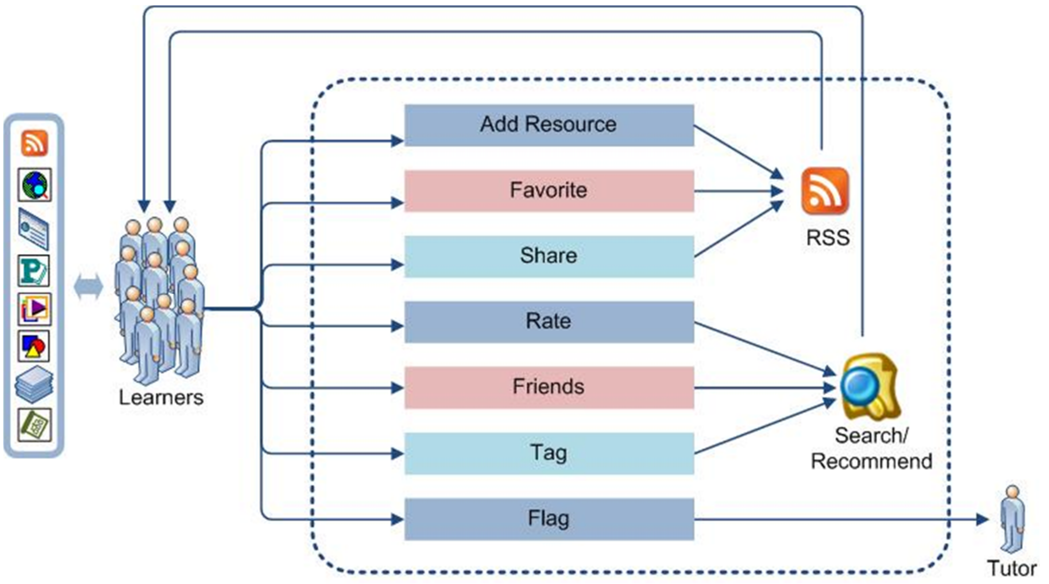


Figure 36: SHAREK process

5.1 Adding Resources

The first and most important step of the SHAREK process is for learners to be able to add, or attach new resources to a course or a lecture. What follows is the data gathered about the different resources added by learners. The model presented here is inspired and based on the IEEE LOM (Learning Object Metadata) (IEEE Learning Technology Standards Committee, 2002) standard. The data collected is divided into six categories highlighted in Figure 37. Note that, when adding a resource, only the dashed elements are required from the contributor. The solid elements are determined automatically by the system with the exception of the *rating* and *flag* elements, which are specified afterwards by the learners who view and use the resource.

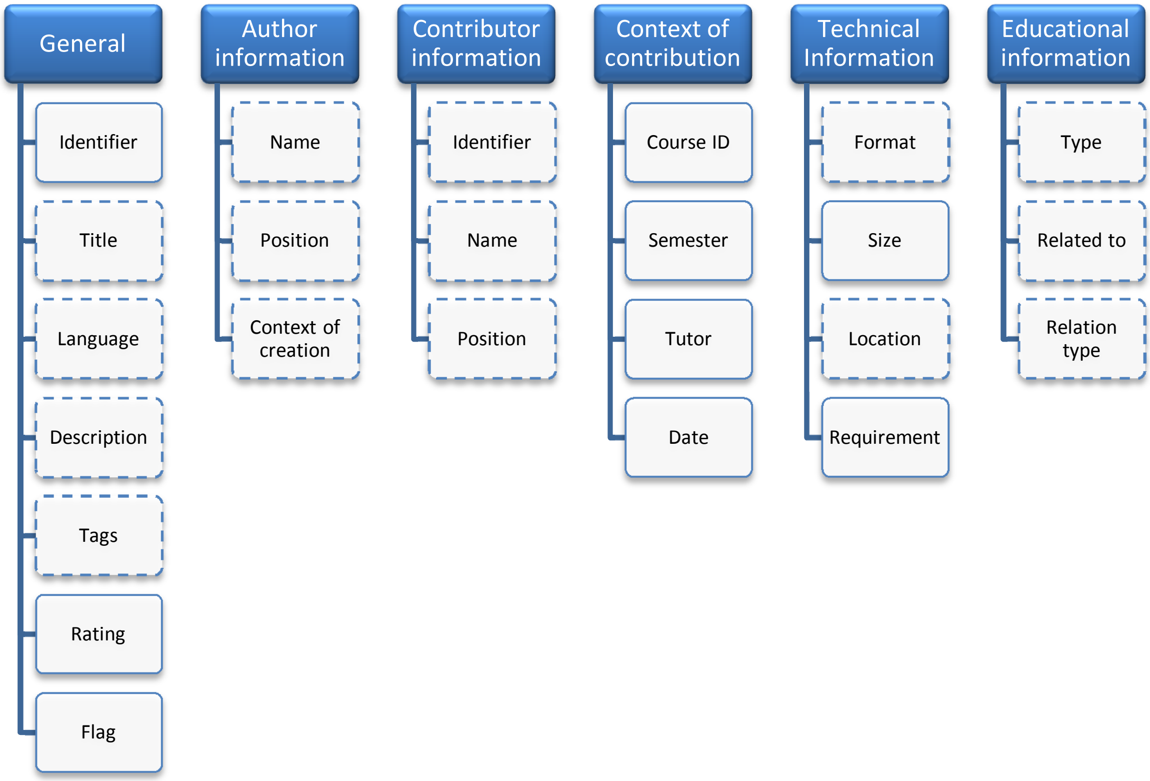


Figure 37: Resources data model

The **general** category contains information such as the resource identifier, a unique id assigned by the system, the title, language and a short description of the resource, which are provided by the contributor at the time of adding the resource. Moreover, the general category contains the Tags associated to the resource. Tags are keywords or terms associated with the resource to describe its content. Tags could be assigned by the contributor when adding the resource, or later on by other learners. Such a collaborative tagging approach, also known as *folksonomy*, is criticized due to its lack of terminological control: if tags are freely chosen, synonyms (multiple words with the same meaning), homonyms (a word with different meanings) and polysemes (a word with multiple related meanings) are likely to arise, thus lowering the efficiency of content indexing and searching. Yet, folksonomy has its strengths, and perhaps the most important is that it directly reflects the vocabulary of users. Indeed, a folksonomy, with its uncontrolled nature, can adapt quickly to user vocabulary changes and needs. Learner Ratings are also information also stored under the general category. A rating is a score (on a scale of 1 to 5) a learner gives to a resource, with regards to its *relevance* to the lesson, its *utility*, and its *clarity* and ease of use. Such scores are used to evaluate the relevance, utility, and clarity of a resource, and are further needed in order to recommend resources to learners (detailed in section 5.3). Moreover, the *flags* are stored in the general category. Learners can flag the resource as being *Inaccessible* (resource cannot be accessed, broken link, unavailable file, etc.), *Unrelated* (Resource content is unrelated to lesson), *Redundant* (resource already exist) or *Plagiarized* (contributor asserts himself as the author of a resource created by someone else). When a resource is flagged, the flag will reflect within the system to advise learners accessing the resource, and a message is automatically sent to the tutor in order to take the appropriate actions.

The **educational** category contains information such as the type of the resource, whether the resource is an exercise, an experiment, a lecture, etc. Moreover, the educational category contains the Related To, and Relation Type information. The first describes which part of the course, or which lesson the resource is related to, and the second describes the relationship of the resource to the lesson: whether to support the theory in the lesson, to contradict it, to illustrate the theory with an example, or to evaluate the knowledge in such a topic.

The **technical** category contains information related as to what are the technical requirements to access the resource (such as plug-in or specific software), the format of the resource (a document, a java applet, a web site, etc.), the resource size (if applicable) and its location. It is important to note the content of *location* varies depending on the format of the resource. Indeed, if the resource is an uploaded file, then the location indicates where this file is stored on the server. On the other hand, if the resource is located online (for example a website), the location, in this case, will contain the URL address.

The **contributor** category contains the information to identify the learner who contributed, or added the resource. The contributor' unique *identifier*, as well as his full *name* and *position* are stored here. The position indicates the title of the contributor, whether a student, a tutor, or even a teaching assistant.

The **contribution context** category holds information about the context in which the resource was added, including the course id, the semester, the tutor and the date the resource was contributed. Such information is important and relates to context in which the resource was created. Indeed, having many resources added shortly after giving a lecture *might* indicate that the students or learners did not quite grasp the material and had to do some research of their own.

Although the contributor could be the author of the resource, there will be many situations when he is not. Thus, the **author category** contains information about the actual author of the resource such as his name and position (student, tutor, etc.) and the context of creation of the resource (*why* the resource was created).

5.2 Sharing and Accessing Resources

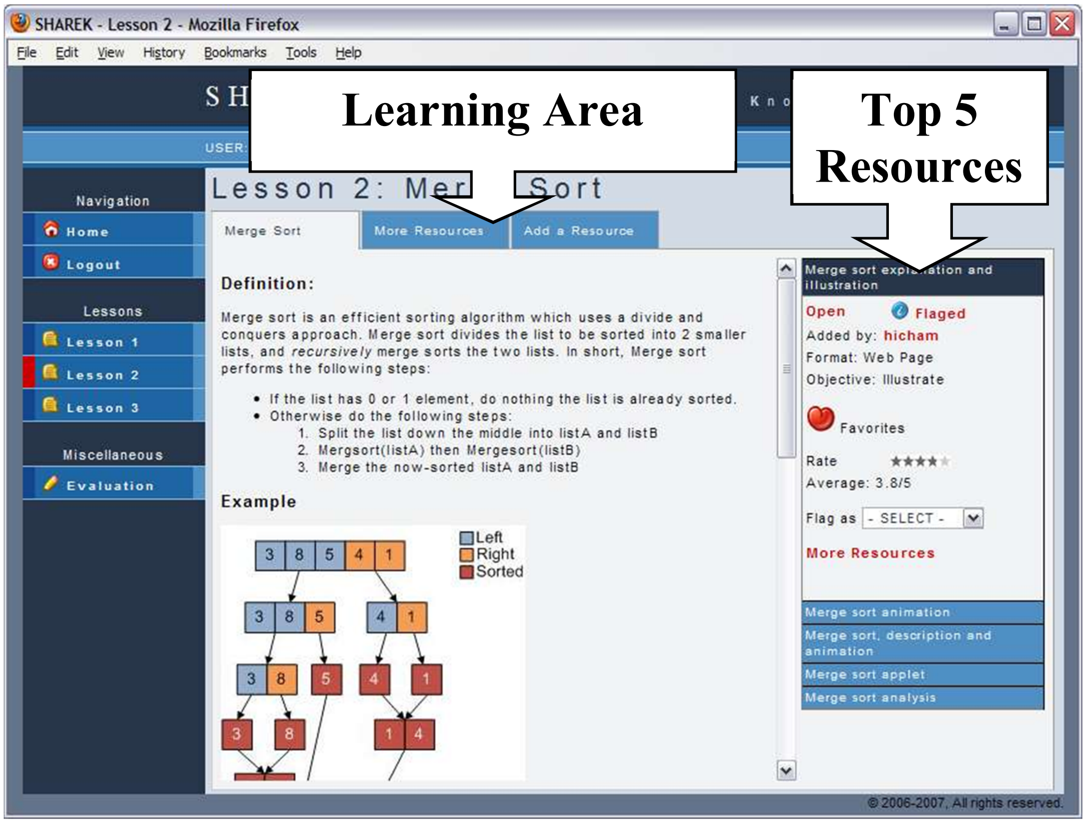
Within such a context, it is important to have a well designed scheme to help learners locate resources. Indeed, with the availability of a multitude of resources attached to a single lesson or lecture, how can a learner easily locate the most suitable resource? In other words, having access to a multitude of resources in the same space, with no means of *filtering* out the unwanted ones, may become cumbersome for the learners and cause a cognitive overload. Thus, it is important to help learners *filter* out resources that he would not like. A simple approach would be to use the best rated resources, but this approach alone is inadequate. Indeed, different learners prefer different resources. Moreover, when relying just on the ratings, new and suitable resources, which have not yet been rated, would be overshadowed by other, probably less adequate, resources that have been rated.

Moreover, that would also affect the performance of the recommender systems. For instance, a CF (Collaborative Filtering) recommender system accumulates user ratings of items, identifies users with common ratings, and offers recommendations based on inter-user comparison (Deshpande & Karypis, 2004). Such an approach suffers from the *cold start* problem, i.e. when dealing with new resources (sparsely rated) and/or with new users (with an undeveloped profile). Since we expect that there will always be new users and new resources within the system, we propose combining several different approaches inspired by Web2.0 to *advertise* resources and help circumvent the cold start. One such approach is the use of an RSS feed to advise the learners of certain events, effectively promoting new and existing resources. Specifically, each learner has a list of *friends*, to which he can add or remove other learners. The learner in question is then kept up to date on his *friends'* resources activity through RSS. Remember that RSS is a family of Web feed formats used to publish frequently updated content. An RSS document, which is called a *feed*, usually contains either a summary of content, from the associated web site or the full text. In short, RSS makes it easier to keep up with updates.

Hence, when a new resource is added, the learners will receive a notification through RSS advising them of this new addition, which will incite them to view, use and rate the new resource. This would in effect advertise new resources, resulting in the resources being used and rated, effectively reducing the effect of cold start. Moreover, when a certain learner adds a resource to his favorites, other learners are also advised of this event through RSS. The motivation behind this is to promote *useful* resources. Indeed, we believe that if a learner adds a certain resource to his favorite it implies not only that the learner appreciated the resource in question, but also his intention to use and reuse the specific resource (an *implicit*, positive feedback). As such, this differentiates the *favorited* resource form other resources used once and then discarded. Another method to promote resources is through *sharing*. In this case, *sharing* is used in the context of *recommending* a resource used by a user within SHAREK. Indeed, a learner who finds a resource within SHAREK can *explicitly* recommend, or *share* this resource with his friends or classmates. The sharing process is also performed through RSS, as such a learner will be advised through the RSS feeds about resources shared with him. It is important to note that in order not to overwhelm the learner with RSS feeds and information, each learner has control over the *granularity* of the information in the RSS feed. As such, each learner can specify the granularity of the information in the feed, controlling the flow of information. For instance, a learner can decide that he wants to be advised of any new addition of resources by any learner, yet, restrict the Favorite notification to only his friends. Thus he would be able to know whenever a learner adds a new resource within the course, and only when one of his friends adds a resource to their favorites.

On the other hand, SHAREK also provides a manual search tool and a CF recommender system. Specifically, learners can search for resources by specifying one or more of the following criteria: language, rating, tags, date added, format, by the educational type, relation type, or even within resources favorite by friends.

Additionally, in the context of the lesson the learner is presented with the Top5 resources attached to the specific lesson. The Top5 are determined using the CF approach (detailed further in section 5.3), and are presented alongside the learning area (Figure 38).



Lesson 2: Merge Sort

Definition:

Merge sort is an efficient sorting algorithm which uses a divide and conquers approach. Merge sort divides the list to be sorted into 2 smaller lists, and *recursively* merge sorts the two lists. In short, Merge sort performs the following steps:

- If the list has 0 or 1 element, do nothing the list is already sorted.
- Otherwise do the following steps:
  1. Split the list down the middle into listA and listB
  2. Mergsort(listA) then Mergesort(listB)
  3. Merge the now-sorted listA and listB

Example

Merge sort explanation and illustration

Open Flaged

Added by: hicham

Format: Web Page

Objective: Illustrate

Favorites

Rate ★★★★☆

Average: 3.8/5

Flag as - SELECT -

More Resources

Merge sort animation

Merge sort, description and animation

Merge sort applet

Merge sort analysis

Figure 38: SHAREK illustration

The CF process is composed of two steps: first determine the neighborhood of the learner, which consists of *k* learners with the highest ratings' similarity. The similarity between the learner *a* and his *neighbor u* is derived using Pearson correlation coefficient. Although the approaches detailed earlier help promote new resources to reduce the effect of the cold start, if the learner is new to the system (or has just a few ratings), the recommender system cannot efficiently determine the neighborhood, nor predict the learner's preference for resources. Thus, in this case, the Top5 will be selected with regard to their overall rating.

5.3 Multi-criteria recommender system

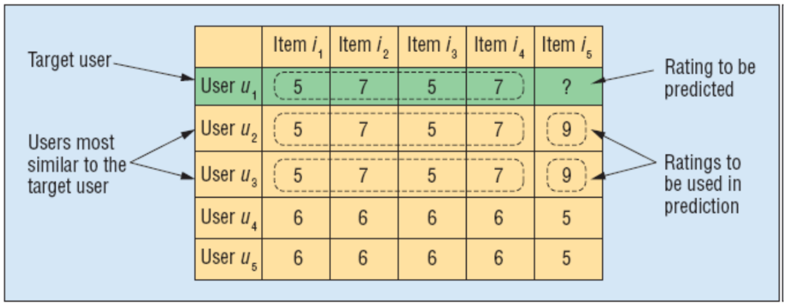
In this section we detail the recommendation technique used within SHAREK. Specifically, SHAREK uses a Multi-criteria recommender system, an approach which we later adapted to recommend scientific articles (Naak, Hage, & Aïmeur, 2009). In detail, Learners can rate each resource according to 3 criteria: *relevance*, *clarity* and *utility*. Usually, the CF process is composed of two steps: the first step consists of determining the neighborhood of the user, which consists of *k* users with the highest ratings' similarity. The similarity between the user *a* and his neighbor *u* is derived using Pearson correlation coefficient highlighted in the following equation:

$$sim_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m}(r_{u,i} - \bar{r}_u)^2}} \tag{6}$$

Where $r_{a,i}$ is the rating given by user *a* to resource *i*, $\bar{r}_a$ is the mean rating given by user *a*, and *m* is the total number of resources. The next step is to use this neighborhood to *predict* the user's rating to an unrated resource. Such predictions are computed as the weighted average of deviations from the neighbor's mean using the following equation:

$$p_{ai} = \bar{r}_a + \frac{\sum_{u=1}^{k}(r_{u,i} - \bar{r}_u) \times sim_{a,u}}{\sum_{u=1}^{k} sim_{a,u}} \tag{7}$$

Where p_{ai} is the rating prediction of user *a* for resource *i*, $sim_{a,u}$ is the similarity between users *a* and *u* obtained using equation (6), and *k* is the number of users in the neighborhood. As recommended in (Melville, Mooney, & Nagarajan, 2002), we use a neighborhood of *k* = 30 users. Figure 39 (Adomavicius & Kwon, 2007) illustrates the CF recommendation that is usually used and based on a single rating per item setting.

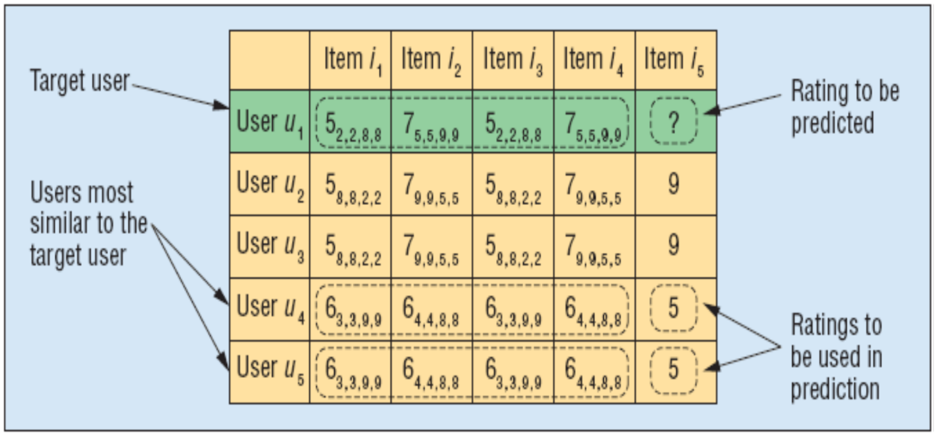


	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	5	7	5	7	?
User u_2	5	7	5	7	9
User u_3	5	7	5	7	9
User u_4	6	6	6	6	5
User u_5	6	6	6	6	5

Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in prediction

Figure 39: CF in a single rating setting (Adomavicius & Kwon, 2007)

One of the shortcomings of this approach is that the recommender system does not know why the user rated the item with a specific score. For instance, two learners might give the same resource the same low score, but for different reasons: one found the resource completely irrelevant to the course, and the second found it useless. In this case, by comparing how two users rate the various aspects of a resource, the CF recommender can determine more accurately how similar the two users are. Figure 40 (Adomavicius & Kwon, 2007) illustrates the multi-criteria approach in contrast to the single criterion. In this case, the overall rating of an item (the larger numbers) are broken down to four ratings on each item. Notice that although the overall ratings (large numbers) of the users u_2 and u_3 are closer to the overall ratings of the target user u_1, they have rated the various aspects (smaller numbers) of the item in a completely opposite manner to u_1, thus u_2 and u_3 actually have opposite preferences to u_1. Consequently, u_4 and u_5 have *closer* preferences when considering the multi-criteria ratings, and are more suited to perform the recommendation.



	Item i_1	Item i_2	Item i_3	Item i_4	Item i_5
User u_1	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$5_{2,2,8,8}$	$7_{5,5,9,9}$	?
User u_2	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$	9
User u_3	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$	9
User u_4	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	5
User u_5	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	5

Target user
Users most similar to the target user
Rating to be predicted
Ratings to be used in prediction

Figure 40: CF in a multi-criteria rating setting (Adomavicius & Kwon, 2007)

Within CF, it is imperative to properly determine the neighborhood, since the accuracy of the rating's prediction relies heavily on this neighborhood, and how *close* the neighbors are to the target user. Hence we propose and compare five various approaches at determining the neighborhood.

The **first approach**, which we refer to as the horizontal (**HZ**) approach, is primarily based on the work presented in (Adomavicius & Kwon, 2007). The similarity between the target user *a* and the potential neighbor *i* is determined for each of the 10 rating criteria, and then the average of these *partial* similarities is used to determine a *global* similarity. The *k* users with the highest global similarity are then used in the prediction process. Although this approach tends to minimize the overall inaccuracy, it does introduce some *noise* in the data, specifically when a neighbor with a high global similarity is very divergent on one or more criteria. For instance user *a* and his neighbor *u* rated most criteria in a similar manner except in two cases. When predicting the rating of *a*, the divergence of *u* will introduce some noise/inaccuracy in the process, even though the similarity is used as a weighing factor (refer back to equation (7)).

In the **second approach**, that we refer to as the vertical (**VL**) approach, the predictions for each of the criterion is performed in the same manner as in the *classical* single criterion rating settings. In other words, for each criteria *j*, the *k* users with the highest similarity in rating the criterion *j* are used in the prediction. The rationale behind this approach is to use the closest neighbors for each criteria instead the closest neighbors overall. As such, this approach takes advantage of the multi-criteria setting, while maximizing neighborhood's similarity with the target user for each criterion separately. Yet this approach suffers greatly when the neighborhood is not very close to the target user.

The size of the neighborhood as well as its similarity to the target user is important. When using Pearson correlation coefficient, a similarity value between 0.5 and 1 implies a high correlation, and a similarity value between 0.3 and 0.5 implies a medium correlation. As such, we set a similarity threshold *T* to 0.3, such that the neighborhood of a target user is close. Nonetheless, applying the threshold alone on the previous approaches is not sufficient, since the size of the neighborhood affects the accuracy of the prediction. Indeed, setting the threshold for the HZ and VL approaches reduced the neighborhood as well as the average performance. As such, in order to complement the neighborhood while maintaining a high similarity, we propose the next two approaches.

In the **third approach**, that we refer to as the horizontal then vertical (**HZ-VL**) approach, the neighborhood is composed first by the most similar neighbors whose average similarity is larger than *T*. If the number of neighbors is less than *k*, then the neighborhood for each criteria *j* is complemented by the neighbors with highest similarity to *a* with regards to criteria *j* and always larger than *T*.

In the **fourth approach**, that we refer to as the vertical then horizontal (**VL-HZ**) approach, the neighborhood is determined in a similar manner as the previous approach, but in this case it is determined *vertically* first, then complemented *horizontally*.

The **fifth approach**, which we refer to as horizontal without noise (**HZ-N**), is actually an enhancement of the first approach, since we expect to have some *noise* in the first approach. Specifically, consider a target user *a* and one of his *k* nearest neighbors *i*. When performing the prediction for criteria *j*, the user *i* will be considered in the prediction, even though his similarity with *a* for that specific criteria *j* is not close, hence introducing some *noise*. We consider such cases as *noisy* data that we simply disregard from the computation. We determine noisy data again by comparing the similarity to the threshold *T* = 0.3.

5.4 Managing Resources

The freedom offered to learners within SHAREK should be balanced with an equivalent amount of control. Indeed, SHAREK is able to control unintended abuses of learners. One such abuse is the uncontrolled and excessive addition of resources, which would sooner or later clog the system with unused and unnecessary resources. As such, we propose using techniques such as a *resource confidence* and *utility* measures, inspired by the confidence and support approach in Association Rules (Cios, Pedrycz, Swiniarski, & Kurgan, 2007), in order to decide whether to keep or remove a certain resource. Specifically, the *utility* of a resource is how useful it is, or how much the learners did like and use the resource. The utility measure is calculated by comparing the number of times the resource was accessed to the number of learners who accessed it. In short, the utility measures the *re-use* of a resource, i.e. the number of times learners re-visit and re-use a same resource. The utility is calculated using the following equation.

$$utility(r) = \frac{\alpha + A_r}{\alpha + L_r} \qquad \textbf{(8)}$$

Where A_r is the number of times the resource *r* was accessed, L_r is the total numbers of learners who accessed the resource, and α is a normalizing value such that when the values of A_r and L_r are small, the utility value of resources with small A_r and L_r values are be drawn towards the same average. In other words, the addition of the value α insures that the number of learners who used the resource is reflected in the utility measure. In order to better understand the need for α, consider the following example: r_1 is a resource accessed by 2 learners, and one of these two learners accessed the resource on 2 occasions. Thus, A_r = 3 and L_r = 2 and *utility*(r_1) = 3/2 = 1.5. Now, consider a new learner just accessed the resource, therefore the new values will be A_r = 4 and L_r = 3 and *utility*(r_1) = 4/3 = 1.333, which is a big variation of 17% in the value of *utility* caused by only one vote. On the other hand, recalculating both values while adding α = 20, the value of utility in the first case is *utility*(r_1) = (20+3)/(20+2) = 1.045, and in the second case *utility*(r_1) = (20+4)/(20+3) = 1.043, thus the variation caused by one vote is less accentuated.

It is important to note that the access counter to the resource is incremented whenever a learner actually opens and uses the resource, thus, it does not account for the cases where the learner *saves* a resource and accesses it from outside of SHAREK.

On the other hand, the *confidence* measure denotes the confidence of the resource, or *how good* the resource is. In short, confidence underlines the number of good evaluations of the resource. As such, we consider it a good evaluation when the resource receives a rating of 4 or 5 stars (scale of 1 to 5), when the resource is added by a learner to his favorites, or when a resource is shared by a learner with his friends or colleagues. The confidence is calculated using the following equation:

$$confidence(r) = \frac{\alpha + G_r + Fv_r + Sh_r}{\alpha + Ta_r} \qquad (9)$$

Where *Gr* is the number of good ratings resource *r* received, Fv_r is the number of time resource *r* was favorited, Sh_r is the number of times resource *r* was shared, Ta_r is the total number of actions performed in resource *r*, including the total number of ratings, Fv_r, Sh_r, and the number of times the resource was flagged, and finally, α is the normalizing value. As such, whenever $utility(r) \times confidence(r) < T$, where T is the experimentally determined threshold, a resource is removed from SHAREK and archived. Concerning new resources, the normalization variable in equations (8) and (9) will keep the utility and confidence within an acceptable average greater than *T*, such as not to discard the new resource.

Another issue to consider is the uncontrolled nature of tags. Undoubtedly, tagging has its strengths, and perhaps the most important is that it directly reflects the vocabulary of users. Indeed, tagging, with its uncontrolled nature, can adapt quickly to user vocabulary changes and needs. Yet, this strength is the source of a main disadvantage. In particular, having too many different tags for a single resource affects the quality of a search based on tags. For example, when searching for “bubble sort” on YouTube, the second video in the list retrieved is about Barack Obama (a candidate for the 2008 US presidential elections), and other returned videos include, among other, a bubble gum advertisement. As such, in order to still take advantage of the flexibility of tags, and increase the accuracy of retrieval, we propose to make use techniques, such as the *tf-idf* weighing scheme (Baeza-Yates & Ribeiro-Neto, 1999), in order to allow the most relevant tags to *float* above the rest. The *tf-idf* weighing scheme is a well established approach within the IR (Information Retrieval) field. Specifically, the term frequency tf_{ij} represents the relevance of the term, or tag i to a document j (in our case a resource j), and idf_i represents the discriminating power of the tag i. As such, the most repeated tags with the most discriminating power can be determined for each resource. Currently within SHAREK, for each resource, only the five highest ranking tags, are considered while the other tags are not discarded, such that they might still *float* to the top.

5.5 Implementation and Validation

In order to validate our approach, we implemented an E-learning prototype platform that supports the tools and functionalities described in this chapter. Specifically, the platform proposes three Data Structures lessons, one for each of the following sorting algorithms: *Bubble Sort*, *Merge Sort* and *Selection Sort*. Moreover, a minimum of 6 resources were originally attached to each lesson. The technical environment of the prototype is: PHP, JavaScript, AJAX, and MySQL.

After logging in for the first time, each learner is requested to complete a survey before having full access to the system. This initial survey collected background information about the learners, such as how often do they refer to learning material outside the regular class content, where do they look for these resources (search engines, Wikis, library books, etc.), how likely are they to share resources with friends, classmates or tutors, how likely are they to use resources referred by friends, and how familiar are they with computer based training and Web2.0. After completing the initial survey, the learners can then access the lessons and their respective resources, where they can add new resources, use and rate existing resources, and test the different functionalities of the system. Finally, after having used and tested the system, the learners complete an evaluation form, answering questions about how easy and intuitive the system is, would they use such a system if it existed, and would such a system encourage them to share resources. Moreover, they were asked to evaluate the functionalities of SHAREK, and whether there were any missing functionalities they would like to add, and finally they were asked to give an overall evaluation of the system, and to provide any comments.

5.6 Results and findings

There were a total of 93 learners who tested the system. The volunteers consisted mainly of graduate and undergraduate students at the Computer Science department of our university (Université de Montréal). We will first start by relating the findings with regard to the survey, then highlight the results of the evaluation.

The survey results mainly reinforce the hypothesis on which we based this work: learners do refer regularly to learning material outside the regular class content, and they are willing to share these resources with friends and classmates. Indeed, when asked, in **question #1** of the survey, to specify on a scale of 1 to 5 (1: never and 5: always) how often they refer to learning material other than what is recommended by the tutor, 93% responded they do so on a regular basis (Figure 41). Moreover, when asked to rate how likely they would use a resource recommended by a classmate or friend, on a scale of 1 to 5 (1: never and 5: always) none of the respondents answered with less than 3, and 73% answered higher than 3, which indicates the willingness of learners to use resources recommended by colleagues and friends. Finally, when asked in **question #4** of the survey, to rate on a scale of 1 to 5 (1: never and 5: always) how probably they would share these resources with the Tutor, Friends or Classmates, learners were the least eager to share resources with their tutor (actually 10% of the respondents answered Never), and most eager to share their resources with their friends. Figure 42 summarizes the averages of learners' responses. Note that female respondents averaged higher than their male counterparts.

With regard to the evaluation of SHAREK and its functionalities, when asked to rate, on a scale of 1 to 5 (1: Poor and 5: Excellent), the ease of use of SHAREK, the intuitiveness of the interface, and whether they would regularly use such a system, most respondents answered favorably, with ratings' averages around 4.

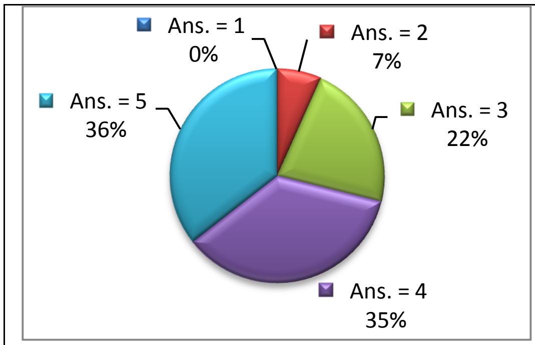


Figure 41: Partition of answers to the survey's question #1

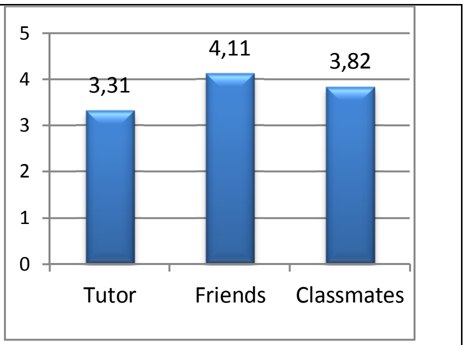


Figure 42: Average of answers to the survey's question #4

Moreover, when asked in **question #2** of the evaluation, if a system such as SHAREK would encourage then to share resources, on a rate of 1 to 5 (1: never and 5: definitely), most respondents answered with a 3 and higher (see Figure 43 for more details). In addition, learners who answered with a low score (1 or 2), where learners who initially, in the survey, had also given low scores to their willingness to share resources. In addition, when asked to rate the functionalities provided by SHAREK (such as the RSS feed, search tool, tagging, rating, etc.), on a scale of 1 to 5 (1: poor and 5: excellent) the respondents ratings averaged at 4.1. Moreover, when the learners were asked if there are any additional functionalities that they would require within SHAREK, there were several interesting suggestions, such as a forum and other student communication tools, functionalities which are usually part of E-learning systems (such as Blackboard or ATutor), which SHAREK is intended to complement.

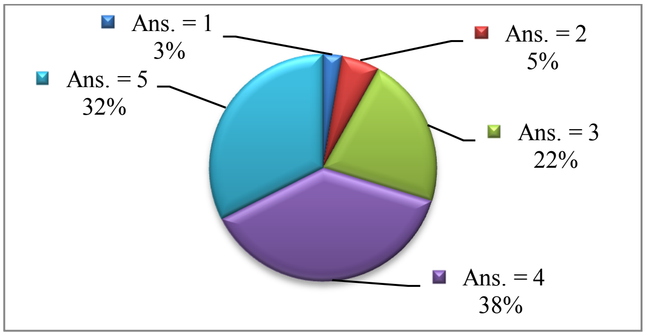


Figure 43: Partition of answers to the evaluation question #2

With regard to the efficiency of the CF resource recommender system, and the tf-idf approach to float the most relative tags, preliminary findings are encouraging, unfortunately the results were insufficient. In fact, for such analysis to be accurate, we require further testing and data. In our case, all the volunteers used the system only once to evaluate it. Thus, the learners' profiles were *sparse* (learners rated a couple of resources only) in order to perform CF recommendations, and the resources were not accessed and tagged intensively (as they would within a regular course) in order to analyze the tags floating approach. Nonetheless, we later relied on a constructed data set to validate our proposed recommendation technique.

5.6.1 Recommender System testing and results

In order to test and compare the accuracy of the five proposed techniques (Chapter 5.3), we use a leave one out approach. Specifically, we randomly select a user and a resource rated by that user. Afterwards we assume that the user hasn't rated the resource yet, and we attempt to predict his ratings. Finally we compare the *predicted* with the *actual* ratings in order to evaluate the accuracy of the prediction. The MAE (Mean Absolute Error) is a metric regularly used in order to evaluate the accuracy of such predictions. The MAE is derived using the following equation:

$$\boldsymbol{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \qquad (10)$$

Where n is the number of predictions, f_i is prediction i and y_i is the actual value. In short, MAE presents the *average* difference between the prediction and the actual value.

The dataset on which the tests were performed was built artificially in a pseudo-random manner. Specifically, SHAREK does not contain sufficient data in order to perform valid tests and draw conclusive results. Similarly, when we adapted the approach in (Naak*, et al.*, 2009), the profiles were again sparse and could not be used to properly test the approach. Alternatively, we contacted Yahoo! (during November 2008) to obtain their movie ratings dataset. Specifically, users can rate movies on the Yahoo! website with regards the following four criteria: Story, Acting, Direction and Visual, as well as provide a fifth, overall rating. However, the currently available dataset contains only the overall rating per movie. Specifically here is a part of the response we received from Yahoo!: “I just chatted with the research scientist and as I was afraid, we don't have the data in the configuration that you'd like”.

The dataset was built in two steps. First, a set of 20 users were created. Then, for each user, ratings for 30 different resources were specified randomly. In order to reduce the random effect of the ratings and to create a certain correlation between the users, we augment the initial dataset of 20 users in a pseudo-random manner. As such, for each of the *initial users*, we create 10 *additional users* whose ratings are based on the ratings of the initial user, varied in a *logical*, but *random* manner. For instance, consider that $R_{a,i}$ is the set of ratings of initial user *a* for resource *i*. Then the set of ratings of new user *j* is R_{ai} + Random values from {-1,0,1}. That is the ratings of *j* are equal to the ratings of *a* ± 1. In summary, based on each of the initial 20 users, 10 more users were created where their ratings *profile* was randomly chosen from the following sets: {-1,0,1}, {-2,-1,0}, {0,1,2}, {-3,-2,-1}, {1,2,3}, {-1,0,1,2}, {-2, -1,0,1}, where each set is at least chosen once in each case. As such, the dataset was composed of 220 different users (20 initial users + 200 additional users) where each rated 30 resources.

5.6.2 Results and findings

In order to test the approaches, a test set of 100 different user/resource pairs were selected randomly. Afterwards, the five approaches were utilized to predict the ratings of the test set, and then these predictions were compared to the actual ratings using MAE. The MAE of each criterion is recorded, as well as the average MAE over all the criteria. The average MAE over the 100 iterations is used to compare the performance of the various implemented approaches. Figure 44 highlights the best case, the worst case and the average case over the 100 iterations of the five approaches: the minimum MAE (MIN MAE), maximum MAE (MAX MAE) and the average MAE (AVG MAE) respectively. Overall, the least performing approach is the VL approach. Specifically, this approach suffered mainly in cases where the similarity between the user and his neighborhood is not *close* enough. The HZ approach addresses this issue where the global similarity is considered. As such, even when the user’s neighborhood is somewhat far for a specific criterion, the overall similarity reduced the overall error in the predictions. On the other hand both HZ-VL and VL-HZ maximize the similarity of the neighborhood, and do offer an overall better performance over HZ and VL separately. On the other hand, although HZ-N has the highest MAX MAE, this approach still offers the best overall performance. HZ-N takes advantage of the overall similarity, while reducing the *noise* induced by neighbors with a high overall similarity, but who are not very similar for a certain criterion.

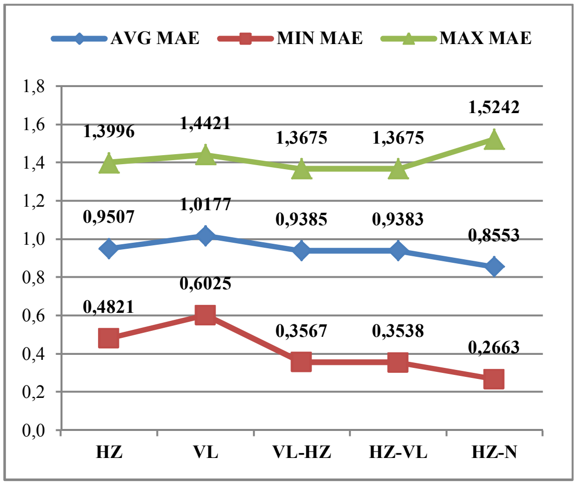


Figure 44: MAE comparison

In order to interpret the values of MAE, it is important to consider the scale on which the ratings are performed. Indeed, an MAE of 0.5 indicates that the predictions, on average, differed by 0.5 of the actual rating. In order to evaluate the impact of this difference, it is important to consider the scale of the predictions. Indeed, a difference on 0.5 on a scale of 1 to 5 is more significant than on a scale of 1 to 20. As such we compare the variation, or the MAE, to the scale to assess its actual impact; that is a MAE of 0.5 on a scale of 5 represents 10% whereas on a scale of 20, it only represents 2.5% and consequently a lower impact on accuracy. Figure 45 highlights the average MAE of the five approaches evaluated on a scale of 5.

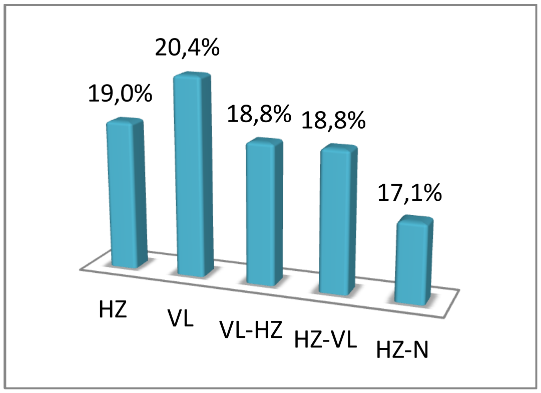


Figure 45: Average MAE interpretation

Although the results are encouraging, and the HZ-N does offer further enhancement of the accuracy over the other approaches, an MAE of 0.8 (or a variation of 17%) leaves some space for improvements. Nonetheless, we presume that the MAE of 0.8 is due essentially to the fact that the dataset is made of randomly generated data. Indeed, we believe that the value of the MAE will be smaller when testing the approaches on a real dataset. Furthermore, we also believe that the HZ-N will perform better than the other approaches, since all the testing was executed on the same dataset.

Chapter 6 : Privacy preserving E-learning

One of the main advantages of E-learning is its adaptability to the learner's specific needs and preferences. Nonetheless, to do so, the E-learning systems collect large amounts of information about the learner, information that could be misused, and therefore violating his *privacy*, which is the claim of individuals to determine what information about themselves is known to others, as well as when and how it is used (Westin, 1967). Other than the case of Head-in-the-sand privacy (by which the learner wants to keep secret his ignorance even from himself), learners might need to keep private different parts of their profile for personal, or competitive reasons. Existing E-learning standards offer some provisions for privacy and the security aspects of E-learning systems do offer some privacy protection; nonetheless it remains unsatisfactory on several levels (Hage & Aïmeur, 2010b). In addition, privacy preserving solutions that are applied in E-commerce environments are inadequate and unsuitable to the context of E-learning. Indeed, while in most E-commerce applications different transactions between the client and the system are fairly independent, in E-learning the interactions between the learner and system are intertwined into a developing process that depends heavily on the path the learner is following. For instance, in the context of E-commerce, the client's history is not required to initiate a transaction or a request. In contrast, within an E-learning environment, the learner must first prove that he has the necessary requirements and history to enroll in a certain program. Afterwards, the learner is constantly required to confirm that he has the proper prerequisites to register for a certain course, that he is indeed registered in the course and has the right to access the learning resources and to pursue the learning activities within that course. Finally, depending on the learner's *learning objective*, the learner is required to provide proof that he finished the required course(s) to an external entity, which could be the E-learning system itself (such as in a prerequisite to another course or to obtain a degree), his manager (to prove he finished his training) or any another entity. In order to go through this process without violating his privacy, the learner requires some mechanism to present his credential anonymously. In this chapter, we introduce the *Anonymous Credentials for E-learning Systems* (ACES), a set of protocols that preserves the learners' privacy. In particular, the ACES allows learners to provide *anonymous credentials* such as to prove that they possess the necessary requirements to register for a course, and/or to prove that they are the legitimate owners of an *Anonymous Transcript* or an *Anonymous Degree*. Although the concept of anonymous credentials is not novel, ACES takes into account the specificities of E-learning. Moreover, in order to prevent the misuse of privacy, ACES prevents the possibility of sharing credentials between colluding learners.

6.1 The Impact of Privacy on Learning

Existing research demonstrates the effect of emotions on learning (Zins, Bloodworth, Weissberg, & Walberg, 2007): positive emotions improve the performance whereas negative emotions hinder the thought processes. Additionally, studies are conducted to evaluate the impact of various factors on the learner's emotional state. The motivation behind these studies is to avoid situations which create negative emotions, while motivating the occurrence of situations which create positive emotions. Consequently, the question of emotional intelligence is becoming more important in education, especially in online learning (O' Regan, 2003). Specifically, tutoring systems rely on several approaches in order to determine the learner's emotional state: including asking the learner to report his emotional state, or by analyzing the learner's response to the tutoring system's actions (de Vicente & Pain, 1998, 2002), or even by recording and analyzing physical signals (Blanchard, Chalfoun, & Frasson, 2007). Additionally, studies are conducted to evaluate the impact of various factors on the learner's emotional state. The motivation behind these studies is to avoid situations which create negative emotions, while motivating the occurrence of situations which create positive emotions. For instance, in (Beck, 2007), Beck investigates the effect of learner control on learning, whereas in (Blanchard & Frasson, 2005), the authors investigate the effects of the learner's "culture" on his emotions and propose a culturally aware ITS (Intelligent Tutoring System). Yet, to the best of our knowledge, there were no studies performed to evaluate the effect of *privacy* on the learner's emotions.

Consequently, in a recent study (Hage & Aïmeur, 2009), we investigated the impact of privacy on the learner's emotions, and whether privacy had a positive or negative impact on learners. Specifically, in this the study, we attempt to determine, in the context of a *web-based assessment*, whether privacy would have a positive effect (effectively reducing stress and helping learners perform better), or a negative effect (learners would become reckless and careless about their grades). The following hypotheses were raised:

> **Hypothesis 1**: Learners are self-conscious, which inherently create performance related stress. Learners are stressed because they want to perform better so that they wouldn't be judged by their tutor and peers as low performers, not to mention cases where the tutor could be biased. Privacy preserving e-Learning will help relieve this pressure, helping learners perform better.
>
> **Hypothesis 2**: In this hypothesis, we consider a negative factor of privacy: since the test results will remain private, learners will tend to become more careless about their grades, and they will perform just good enough to pass.

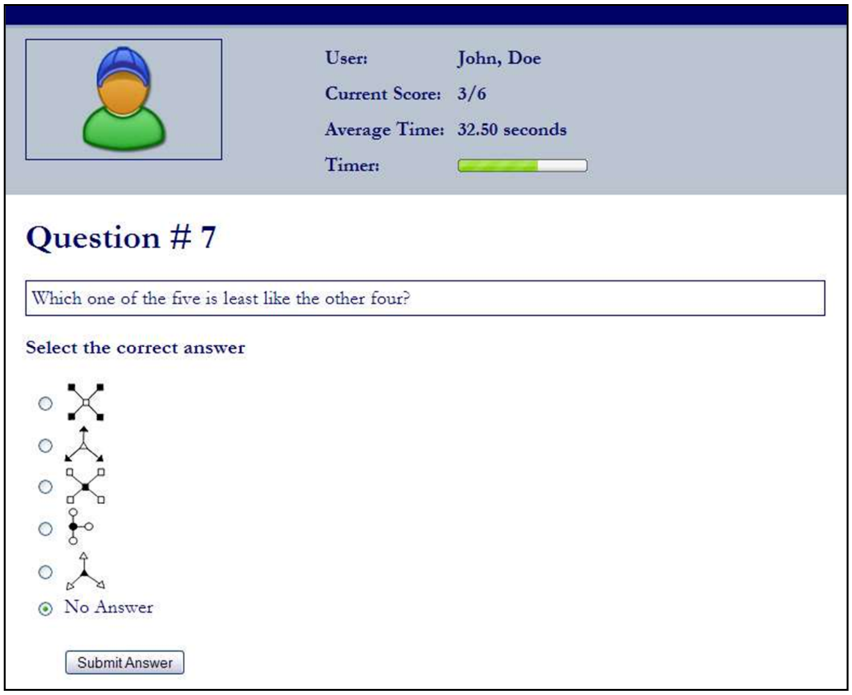
6.1.1 Testing procedure

The testing procedure is composed of three major sections: Registration and Instructions, the Survey, Test#1 performed in an environment with *no privacy*, and Test#2 performed in a *privacy preserving* environment.

Registration and Instructions: registration was required to collect demographic data, including the sex, age group, country of birth, as well as the education field and education level. Additionally, during registration the participants are required to choose one of the eight available avatars, instead of an identification image regularly required in exams. After registration, the participants are provided with instructions describing the whole procedure, as well as explaining the test environment and the various emotions they will be presented with to report their emotional state.

The Survey: the participants completed a short survey, whose purpose is to determine the participants' opinions and conceptions about Privacy and e-Learning. Specifically, the survey inquires about how comfortable the participants are when sharing their test results with the tutor, their friends or classmates, as well as how they would react if the access to their grades was in an anonymous manner.

Test#1, no privacy: after the survey, the participants will take two tests, each consisting of 15 IQ questions. The time allowed to answer each question is 45 seconds (a timer advises the participants of the remaining time), and the passing grade is set to 8/15. The first test (illustrated in Figure 46) is performed in a classical manner. That is the participant's actions are recorded and linked to him: his answers, score, rank and time spent to answer each question are accessible to the tutor. Moreover, during the test, the participants can see their photo (the avatar selected at registration) and their name The participants are presented with one question at a time, and after answering the system reports whether the answer is correct or not. The progressive score, as well as the average time spent to answer the questions are available throughout the duration of the test.



User: John, Doe
Current Score: 3/6
Average Time: 32.50 seconds
Timer:

Question # 7

Which one of the five is least like the other four?

Select the correct answer

○
○
○
○
○
● No Answer

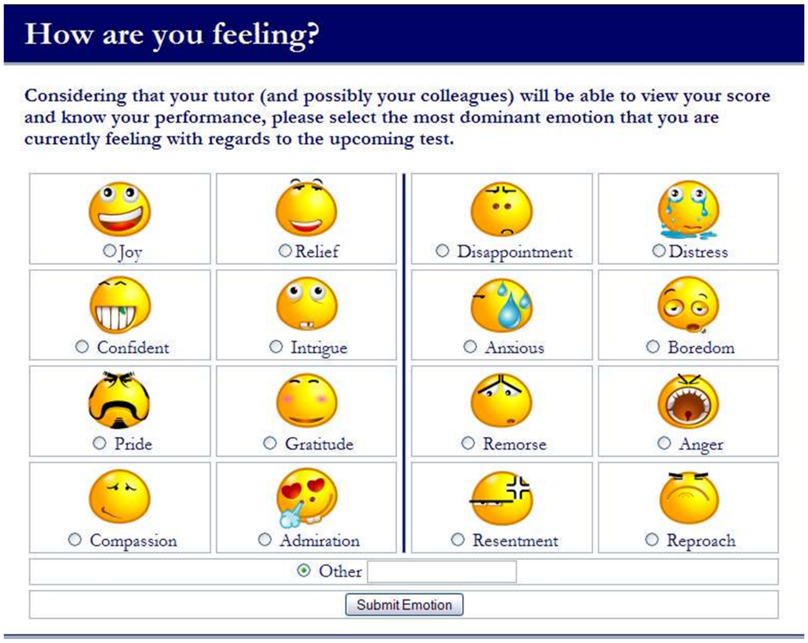
Figure 46: First test (no privacy) environment.

Test#2, with privacy: the second test is performed in a privacy preserving manner. The participant's actions are recorded, and he is still accountable for his results, nonetheless the tutor cannot link the recorded information to the specific participant. During the second test, the participant's image is not visible and his name is replaced with the *randomly generated id* (rid). The rid is used when recording the actions during the test, and ideally, it is known only by the participant, hence ensuring his privacy. Nonetheless, for the purpose of our study and analysis, the system is able to link the rid to the participant.

Design of the tests: since the participants undergo both tests, it is imperative to avoid the case where participants becomes familiar with the style of the IQ questions and the logic needed to answer them, which will clearly bias the results of the second test. Thus, different IQ questions styles were employed in each test. Moreover, to ensure that both tests were of comparable difficulty, all the questions where taken from the same IQ test[13], and we performed prior testing with the help of volunteers. During the testing with the volunteers, we were as well able to determine the time constraint of 45 seconds allotted to answer each question.

Recording the emotion: before and after each test, the participants are asked to express their presently most dominant emotion. To do so, the participants are presented with 16 different emotions, arranged on four columns by four rows (Figure 47 highlights capturing the emotional state of a participant before test#1). The two columns on the left contain *positive* emotions, whereas the two columns on the right contain *negative* emotions. Additionally, the participants have the option to manually specify their emotional state. Specifically, when inquiring about their emotional state before the test, the participants are asked to report their most dominant emotion while considering that the tutor, and possibly their colleagues will be able to view their grades and know their performance (or considering that the tutor and their colleagues will not be able to view their grades in the privacy preserving test). After each test, the participants are requested to report their most dominant emotion with regards to the grade that they just received.

[13] www.hostedtest.com/iq-tests.html



How are you feeling?

Considering that your tutor (and possibly your colleagues) will be able to view your score and know your performance, please select the most dominant emotion that you are currently feeling with regards to the upcoming test.

○ Joy	○ Relief	○ Disappointment	○ Distress
○ Confident	○ Intrigue	○ Anxious	○ Boredom
○ Pride	○ Gratitude	○ Remorse	○ Anger
○ Compassion	○ Admiration	○ Resentment	○ Reproach

⊙ Other

Figure 47: Capturing the Emotional state

6.1.2 Test results

Participants were solicited in various universities, most of which are located in Canada. A total of 84 participants took part to the experiment. Nonetheless, in our analysis we used only 77 records, due to incomplete data. Specifically, 6 participants did not complete the experiment and aborted at different stages, and one participant completed the whole experiment, but we believe that his/her answer selection was random at best: the average time to answer the test questions is around 2 seconds for the first test and 3 seconds for the second test whereas the respective scores is 2/15 and 3/15.

Out of the 77 participants, 26 are females and 51 males. Although the age group selection at registration contained 5 groups (<18, 18-21, 22-25, 26-30, >30), no participants fell in the younger two groups (Figure 48). The participant’s most dominant fields of study are Information Technology and Engineering related (71.5% combined) and the majority are graduate students (Figure 49).

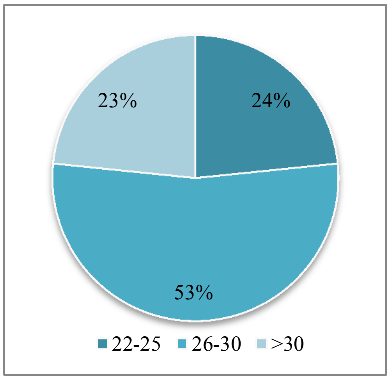


Figure 48: Age group partition

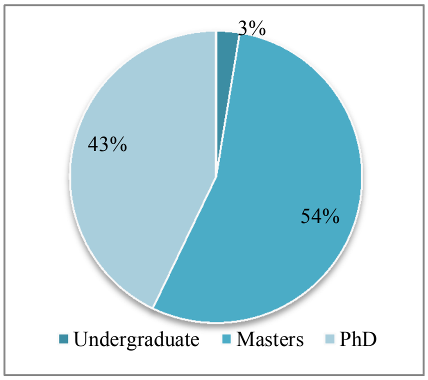


Figure 49: Education level partition

Survey Results

In the survey, 75% of the participant responded that they would rather keep their grades private, and when asked how comfortable they are sharing their grades with their tutor only, friends only, or everyone, 87% were comfortable sharing their grade with the tutor, compared to 38% with friends and only 8% with everyone. In particular, 69% responded that their willingness to share their grade is directly proportional to their performance, i.e. they are more willing to share their grades if they perform good and vice-versa, which is an indication that learners are self-conscious as highlighted in *Hypothesis 1*. In contrast, participants responded more favorably to sharing their grades anonymously, where 66% were comfortable in sharing their grades anonymously with everyone, while reporting that their performance in the case of anonymity had little impact on their willingness to share. Finally, 97% of the participants are in favor of Privacy within e-Learning. The following table summarizes the major findings of the survey.

Table 9: Summary of survey

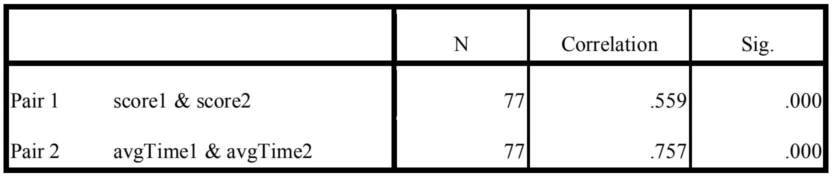
Not comfortable that the tutor can see their mistakes and performance	50%
In favor of the tutor performing a blind correction	87%
Would rather keep their grades private	75%
Comfortable sharing grades with the tutor	87%
Comfortable sharing grades with friends	38%
Comfortable sharing grades with everyone	8%
Willingness to share grades directly proportional to performance	69%
Comfortable sharing grades *anonymously* with everyone	66%
In favor of Privacy in e-Learning	97%

Testing Results

The purpose of this work is to evaluate the impact of privacy on the learner's emotional state. Specifically, we analyze, using SPSS[14], several recorded factors along with the participants' feedback in order to confirm or refute the hypotheses stated in section 6.1.

Hypothesis 1: Learners are self-conscious. Thus, privacy will help reduce performance related stress, and learners will tend to perform better. In order to validate this hypothesis, we compare (Table 10) the average score and the average time spent to answer a question for both tests, where *score1* is the average for the test#1 (no privacy) and *avgTime*1 is the average time spent per question in test#1. Respectively, *score2* and *avgTime*2 represent the score and the average time spent to answer a question in the test#2 (with privacy). Table 10 summarizes the correlation between the scores and the average time. Specifically, we can see that there is a strong correlation between *score1* and *score2* (r = 0.56, $p < 0.01$) and a strong correlation between *avgTime*1 and *avgTime*2 (r = 0.76, $p <$ 0.01). This is not very surprising since one would be expected to perform fairly similarly on both tests.

Table 10: Correlations for average score and time



		N	Correlation	Sig.
Pair 1	score1 & score2	77	.559	.000
Pair 2	avgTime1 & avgTime2	77	.757	.000

Additionally, the paired-samples t-test (Table 11) reveals differences between the scores, $t(76) = -5.913$, $p < 0.01$, which indicates that the mean *score2* is higher than the mean *score1*. Similarly, the paired-samples t-test (refer to Table 11) reveals differences between the average answer time, $t(76) = 15.008$, $p < 0.01$. These results indicate that the mean *avgTime*2 is lower than the mean *avgTime*1. Consequently, since the difficulty of both tests is comparable, we can deduce that on average, the participants performed better on the privacy enhanced test.

[14] http://www.spss.com/

Table 11: Paired samples t-test for average score and time

	t	df	Sig. (2-tailed)
Pair 1 score1 - score2	-5.913	76	.000
Pair 2 avgTime1 - avgTime2	15.008	76	.000

In order to evaluate the impact of the emotional state of the participants on their performance, an independent samples t-test was performed to determine the impact of the reported positive and negative emotions before each test and the obtained score. The results indicate that there was no significant difference in performance for the test#1 between participants who reported a positive emotion and those who reported a negative emotion, $t(72) = 1.4$, $p = 0.15$. Similarly for the test#2, there was no significant difference in performance $t(60) = 0.61$, $p = 0.14$. Consequently we could conclude that regardless of their reported emotional state (whether positive or negative), the participants performed similarly. Although it is not *statistically significant*, we can still note that participants with a positive emotion prior to the test performed better than participants with a negative emotion prior to the test. In addition, note as well that this difference did decrease in test#2 (with privacy), which tends to imply that negative emotions had a smaller impact within the *privacy preserving* test.

Hypothesis 2: Since the test results are private, learners will tend to be careless about the score, as long as they pass. For this hypothesis to be true, the participants' emotion should not be affected by the result of the test, and they would report a positive emotion. Accordingly an independent samples t-test was performed to determine the impact of the score after test#2 on the reported emotions (positive or negative). The results indicate that there was no significant difference between the scores of the participants who reported a positive emotion and those who reported a negative emotion after viewing the test results, $t(41) = -0.84$, $p = 0.41$. Consequently we conclude that the participants' reported emotional state, whether positive or negative, was not affected by the test score. Alternatively, when comparing the reported emotional state of the participants after the tests, a higher number of participants conveyed a positive emotion after the second test (with privacy) which implies that although the test score is private, it remains an important factor. Referring to Table 12, we notice that the number of participants who reported a positive emotion after the test has increased from 16 after test#1 (no privacy), to 29 after test#2 (with privacy). This variance is expected since on average the participants performed better in test#2, hence they are more satisfied with their performance, even though the score is private.

Table 12: Change in reported emotions after test1 and test2

		After test2			
		Positive	Negative	Neutral	Total
After test1	Positive	12	4	0	16
	Negative	17	42	0	59
	Neutral	0	0	2	2
Total		29	46	2	77

In conclusion, the conducted experiment which involved 77 participants, mainly Masters and PhDs, attempted to determine whether privacy will have a positive, or a negative impact on the learners in the context on a web-based assessment. In particular, we investigated two issues: (1) whether privacy will reduce performance related stress and help learners perform better on their test, and (2) whether privacy will make learners careless about their grades. The obtained results indicate that in the context of privacy, learners performed better, with an increased mean score and a shorter question response time, hence indicating that privacy has a positive effect on learners.

6.2 Privacy Preserving E-learning

In this section we start by introducing the learning process within a classical environment. Afterwards we present the framework for privacy preserving E-learning and then discuss the desirable properties required from a privacy preserving E-learning system in order to perform the learning process while respecting the privacy requirements set by our framework.

6.2.1 Learning Process

Whether in a classical, classroom environment, or in an E-learning environment, the learner must go through the learning process highlighted in Figure 50. At each step, the learner is required to provide some form of credentials.

First, in order to enroll into a certain program, the candidate must demonstrate that he has the proper qualification: for example, to enroll in a Bachelor program, the candidate must have at least finished his schooling! Similarly, in order to enroll in a Masters program, the candidate must have his Bachelor's degree in a related field, and a GPA (Grade Point Average) above a certain level. Thus, in the enrolment process, the candidate presents his credential to the E-learning system, which, in turn, validates these credentials with the proper authority or the issuer of these credentials.

Second, in order to register for a course, the learner must confirm he has the necessary qualifications: that he is indeed enrolled in the proper program, and that he has successfully finished the course's prerequisites. In this case, the E-learning system accesses the appropriate data (learner's profile, course prerequisites, etc.) in order to validate the request.

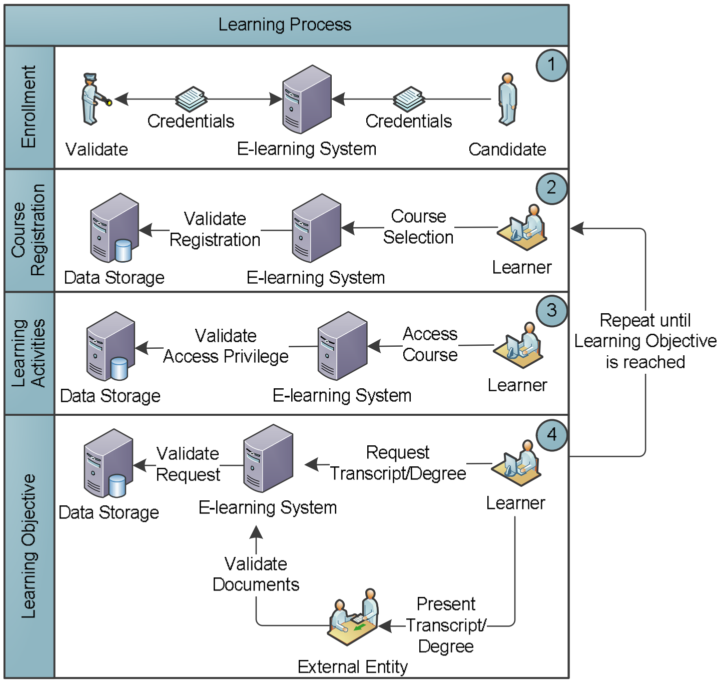


Figure 50: Learning Process

Third, to access the course's learning resources, the learner must prove that he is part of the class and has the right to access these resources and activities. Again, in this case, the E-learning system verifies the appropriate data (learner's profile, course structure and syllabus, etc.) in order to confirm the access privileges.

Finally, after successfully finishing the course, the learner must either select new courses (repeat steps 2 and 3) or, alternatively, he already reached his learning objective. In the later case, the learner might need to confirm that he has reached his goal to an *external entity*: the E-learning system (to receive a degree or pursue higher education), to his manager (to attest that he finished his training), or even to a potential employer. In this case, the learner first requests a transcript or a degree from the E-learning system, and then presents the acquired document to the external entity, which in turn, can validate the authenticity of the documents with the proper authority, or the issuer of the documents. In a classical E-learning environment, since the learner's data is readily available, the above illustrated process is performed without much difficulty.

6.2.2 Framework for Privacy Preserving E-learning

Different learners have different privacy requirements. With this in mind, consider the following components of the learner's data (illustrated in Figure 51), which are of interest from a privacy point of view.

The identity (id): refers to information that makes it possible to determine physically who the learner is (or at least to seriously circumscribe the possibilities). This includes data such as his name, address, and student id number.

The demographic profile (dp): refers to demographic characteristics of the learner, such as age, gender, weight, race, ethnic origin, language, etc.

The learning profile (lp): refers to information such as the learner's qualifications, his learning style, interests, goal and aspirations.

The course history (ch): lists the courses the learner has followed in the past, and their respective information such as the learner's activities within the course and his final grade.

The current courses (cc): lists the courses in which the learner is currently registered and those he is attending, as well as the courses' respective information such as the learner's activities within the course.

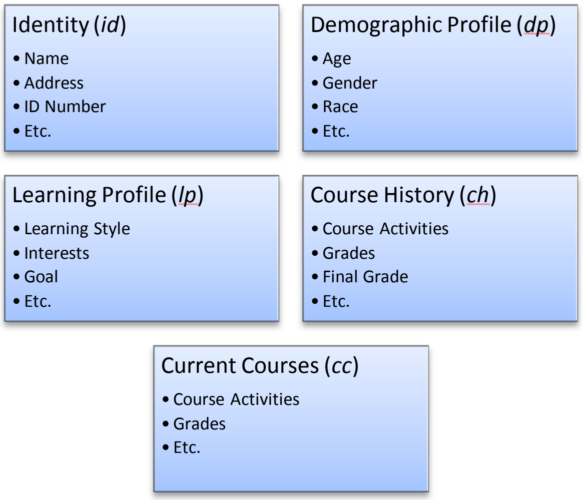


Figure 51: Learner's data

Moreover, we define, in this context, a learner's activity within a course as being any act involving one of the course's tools or resources. For example, an activity might involve the posting of a message in the forum, using one of the course's learning objects, or even taking a quiz or a test. The above elements constitute the personal information on which we base our privacy framework for E-learning systems (Aïmeur*, et al.*, 2007b) composed of the following privacy levels (Figure 52):

1. **No Privacy:** the learner doesn't wish, or doesn't care to keep private any of his information. He does not mind the compilation of a dossier about him that consists of his identity, demographic information as well as his learning history.
2. **Soft Privacy:** the learner wants to keep his identity and demographic profile private, but he does not mind if the tutor has access to his learning profile, course history and current courses.
3. **Hard Privacy:** the learner wants to keep his identity, demographic profile, course history and learning profile private, but he does not mind if his current courses are known.

4. **Full Privacy:** the learner wants to keep secret every component of his personal data.

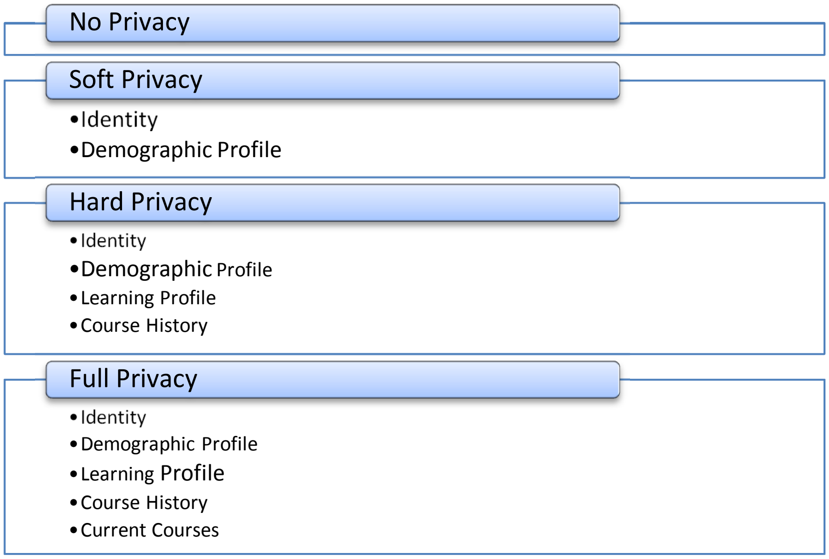


Figure 52: Proposed levels of privacy

Another dimension to consider, which is independent of the personal data listed above, is the tracking of learners within a course. Even under soft, hard or full-privacy constraints, some learners might not want the tutor to know their activities and navigation within the system. Thus, we introduce the following terminology (Aïmeur*, et al.*, 2007b), to account for the levels of tracking that different learners might accept:

1. **Strong Tracking:** the system can relate the activities performed within all the courses to the specific learner, even though that learner may be anonymous. In this case, the system can track the same learner u and his access to courses c_1, c_2 … c_n.
2. **Average Tracking:** the system can relate the activities within a course to the same learner u, but cannot relate them to other activities within other courses. In this case, the system can relate the activity of u_1 in c_1, of u_2 in c_2 … and of u_n in c_n, but cannot link u_1 to u_2 to … u_n.
3. **Weak Tracking:** in this case, although the system recognizes the learner u as a regular visitor, it cannot link him to a course nor trace his activities.
4. **No Tracking:** in this case, the system cannot even recognize the learner u as a recurring user of the system.

In order to illustrate the levels of tracking, consider the following example: a learner, Alice, is using a privacy-aware E-learning system to take the following courses: CSC101 and CSC102. In the case of **Strong Tracking**, Alice creates a pseudonym, A, and uses it to access and perform the learning activities in CSC101 and CSC102. Hence, the system can track the activities of A within both courses, but cannot link A to Alice. In the case of **Average Tracking**, Alice creates two pseudonyms, A1 and A2, one for each course, such that the system cannot relate A1 and A2 to Alice, nor to each other. Hence, whenever Alice needs to access and perform the learning activities in CSC101 or CSC102, she uses respectively A1 or A2. In the case of **Weak Tracking**, the system only records that Alice was logged in, but leaves no trace of her activity (nor identity). And, in the case of **No Tracking**, the system cannot even trace that Alice was logged in at all. Selecting No Tracking is similar to using a guest account to access a demo of the E-learning system.

6.2.3 Desirable Properties and Considerations

In a Privacy-Preserving E-learning environment, the learner must provide his credentials throughout the learning process highlighted in

Figure 50, while maintaining his privacy. As such, the learner must be able to present the necessary proofs, without revealing his personal information!

Enrollment

Ideally, in a privacy preserving E-learning environment, first, the learner would be able to prove that he satisfies the requirements in order to enroll in a certain program, without revealing who he is – whether those previous studies were performed in a privacy-preserving environment or not! Indeed, there are four cases to consider: whether the learner comes from a privacy-preserving E-learning environment or not (cases 1 and 2), and whether he wants to continue his learning in a privacy-preserving environment or not (cases 3 and 4). Moreover, the E-learning system must include provision to allow the learner to switch during his studies, from privacy-preserving to classical E-learning and vice versa.

Course Registration

In this case, the privacy-preserving considerations highly depend on the learner's preferred levels of privacy and tracking. Indeed, other than the case of No Privacy with

Strong tracking (the case of non privacy-preserving E-learning) the various combinations of Privacy and Tracking levels require different approaches. In this work, we limit our discussion to the case of hard privacy with average tracking.

As such, in the case of *Hard Privacy*, for the registration process, the learner must prove that he has completed the prerequisites for the desired courses without revealing his course history. Specifically, knowing that a learner has completed the requirements of a certain course partially reveals his course history. Thus, it is important for a learner, in order to register for several courses, to provide a separate anonymous credential indicating that he satisfies the requirements for each different course independently. Consequently the E-learning system cannot *piece together* the learner's course history. Similarly, the E-learning system must not be able to *keep track* of the learner's registration process; otherwise the E-learning system would be able to link the separate anonymous credentials back to the learner.

On the other hand, it is important for any mechanism that provides such functionalities to have specific provisions, such as to avoid learners taking advantage and abusing their *state of anonymity*. Indeed, one must consider the fact that learners might *abuse* the cover of anonymity and share their anonymous credentials. For instance, consider two learners Alice and Bob. Alice successfully completed the course C_1, and Bob successfully completed the C_2 course. Moreover, both C_1 and C_2 are prerequisites for the course C_3. In order to register for the course C_3, Alice can use her anonymous credentials, stating that she successfully finished C_1, along with Bob's anonymous credentials for C_2. Likewise, Bob can use his anonymous credentials for C_2 along with Alice's for C_1 to register for C_3. Thus, these anonymous credentials must be valid only to their legitimate owner, and hard, if not impossible to share.

Learning Activity

Again, in this case, the privacy-preserving considerations highly depend on the learner's preferred levels of privacy and tracking. As such, in the case of Hard Privacy, the learner must use at least the Average tracking option. Indeed, if the learner uses strong tracking, then the E-learning system can relate to the same learner all his current courses, and thus, successfully piece together the learner's partial course history. As such, the system can relate the activity of u_1 in c_1, and the activity of u_2 in c_2. Moreover, the E-learning system recognizes that u_1 is registered in c_2, and that u_2 is registered in c_2, but cannot link u_1 to u_2.

It is important to note that in this case, sharing the anonymous credentials to access the learning material of a course does not increase the risk of *consensual impersonation* (where the actual learner asks someone else to take the course or perform the learning task in his stead), since it is an existing issue in E-learning.

Learning Objective

In the case of *Hard Privacy*, after completing all of the courses' learning material and activities, first, the learner's performance must be evaluated, afterwards, his profile, notably his course history, must be updated consequently. Evaluating the performance of the learner in the course is fairly straightforward: in the case of Average Tracking, the activities within the same course are traceable to the same learner. On the other hand, updating the learner's profile must be completed such that the E-learning system is unable to track the learner and gradually construct a profile of the learner. Indeed, other than piecing together the learner's course history, by tracking the learner's activities across several courses, the E-learning system could successfully build the learner's Learning Profile thus effectively reducing the level privacy from Hard to Soft.

After updating his profile, the learner can register for new courses, or place a request for an *anonymous transcript* or an *anonymous degree*. The learner can then present his anonymous transcript/degree to an external entity, which must be able to verify the validity of such a document, and confirm that the learner is the rightful owner of the document without infringing his privacy.

On the other hand, it is important to have specific provisions, such as to avoid learners taking advantage and abusing their *state of anonymity*. Essentially, one must consider the fact that learners might share their anonymous credentials, in a similar scenario as described previously, in order to obtain a degree.

6.3 Related work on privacy in E-learning

E-learning systems use information about a learner in order to adapt the learning activity and the interactions of the E-learning system. Such information is referred to as the learner profile or learner model. Many E-learning systems use their own internal representation of the learner model. Nonetheless, there are several standards and specifications to represent the learner model, including the IEEE LTSC Personal and Private Information draft standard (LTSC) and the IMS Learner Information Package (IMS). Although these specifications contain some attributes and means that may uphold learner privacy, the detailed specification is still missing. Moreover, the learner involvement in deciding which information is private or not is not enabled (Jerman-Blazic & Klobucar, 2005).

On the other hand, there were concerns raised with regards to security. There exists literature, such as (Franz*, et al.*, 2006; Raitman*, et al.*, 2005b), on how to achieve basic security requirements: *confidentiality, integrity* and *access control*. The security of existing E-learning systems (such as Blackboard, WebCT, or Atutor) does provide a certain level of privacy. As such, **integrity** guarantees that the data is not maliciously or accidentally tampered with or modified: for example, when the learner submits his test, he requires the guarantee that his test answers are not modified after his submission. Moreover, **confidentiality** assures that the data and information is kept secret and private and is disclosed only to the authorized person(s): for example, test scores must be accessible only to the appropriate tutor. The confidentiality of the information is considered at two different stages: while it is being transmitted to/from the E-learning system, and when it is stored within the E-learning system. In the first case, the data can be encrypted such that only the appropriate receiver can read the data. In the second case, **access control** mechanisms can be employed to restrict access to the data. Access control cannot totally guarantee the privacy of the learner: first of all, it does not protect against a *super user* with full access privileges. Moreover, none of the previously mentioned security mechanisms can be used to observe the *core* of the definition of privacy, in such that the learner has no control on what information about him is being gathered by the E-learning system and how it is used. Although Privacy Policies have been provided for this purpose (Yee & Korba, 2003), they cannot restrict unwanted access to the data.

As such, although security is important within E-learning systems, it is not enough to properly protect learner privacy, and it does not provide the means for anonymous credentials.

6.4 Preliminaries

In this section we present some preliminaries required to better understand the proposed solution in the next section.

6.4.1 Pseudonymous and Anonymous Credentials

Certificate Authorities (CA) are trusted entities whose central responsibility is certifying the authenticity of entities (persons or organizations) and their public keys. More precisely, an entity certificate issued and signed by a CA acts as proof that the legitimate public key is associated with the entity. Usually, the CA makes the decision to issue a digital certificate based on evidence or knowledge obtained in verifying the identity of the owner. In the context of privacy-preserving systems, the CA cannot be used to protect user private data and transactions. Therefore, new approaches are considered.

In 1985, Chaum (Chaum, 1985) introduced the concept of pseudonymous credentials to protect individual privacy. More precisely, the resulting system enables users to engage in several anonymous and untraceable electronic transactions with organizations. Two years later, the implementation of this concept was proposed by Chaum and Evertse (Chaum & Evertse, 1987). However, it was not suitable in practice because it relied on the existence of a semi-trusted third party participating in all communications.

In 1995, Chen (Lidong, 1995) proposed an approach that relies on blind signatures based on discrete-logarithms. Although efficient, this approach does not address the colluding users problem. Moreover, in order to use the same credential untraceably several times, the user must obtain a different signature form the issuing party for each instance.

In 2000, Brand (Brands, 2000) used several properties of Chaum's original concept to introduce a privacy-enhanced certificate system. Here, the system consists of two entities (Organizations and Users) and two protocols (Issue and Show). Unfortunately, Brand's approach is also limited for practical implementations. For instance, every Brand's credential is unique, thus it can be showed only once; otherwise, transactions by the same user could be linked. To overcome this limitation, the system needs to be extended by introducing recertification or batch issuing mechanisms (Brands, 2000).

Another implementation of Chaum's proposal is the credential system proposed by Camenisch and Lysyanskaya (Camenisch & Lysyanskaya, 2001), which is based on previous work by Lysyanskaya *et al.* (Lysyanskaya, Rivest, Sahai, & Wolf, 1999). Here, users first register with the root pseudonym authority before using the system. Thus, users are unable to build up several parallel identities and they can be traced in case of fraudulent transactions. Users are limited to at most one pseudonym per organization. Each credential is related to a single attribute and an expiry date. Moreover, users are able to choose which statement to prove about an attribute, such as choosing to prove that the value of attribute "age" is greater than 18. While considered an interesting implementation of the concept of pseudonyms and credentials, Brand's solution has the drawback of being based on zero knowledge proofs, thus the system is difficult to implement in environments with limited resources.

Although the previous general solutions for anonymity, pseudonyms and credentials can be used to solve issues related to user privacy in various domains, we aim to use the specific structure of an E-learning setting in order to seek more efficient solutions. Therefore, we introduce the concept of Anonymous Credentials for E-learning Systems, to enable privacy-enhanced E-learning.

6.4.2 Cryptographic preliminaries

This section reviews some cryptographic tools and primitives that are used in the development of the Anonymous Credentials for Privacy-Preserving E-learning paradigm.

Public key cryptography

Public Key Cryptosystems (PKCs) were introduced independently by Merkle (Merkle, 1978) and by Diffie and Hellman (Diffie & Hellman, 1976). Formally, a PKC consists of three efficient algorithms: a *Key-Generation Algorithm* that generates pairs of Secret Key (SK) and Public Key (PK); an *Encryption Algorithm, E*, that computes the ciphertext for a message given the public key; and a *Decryption Algorithm, D*, that computes the cleartext message back from the ciphertext, given the secret key.

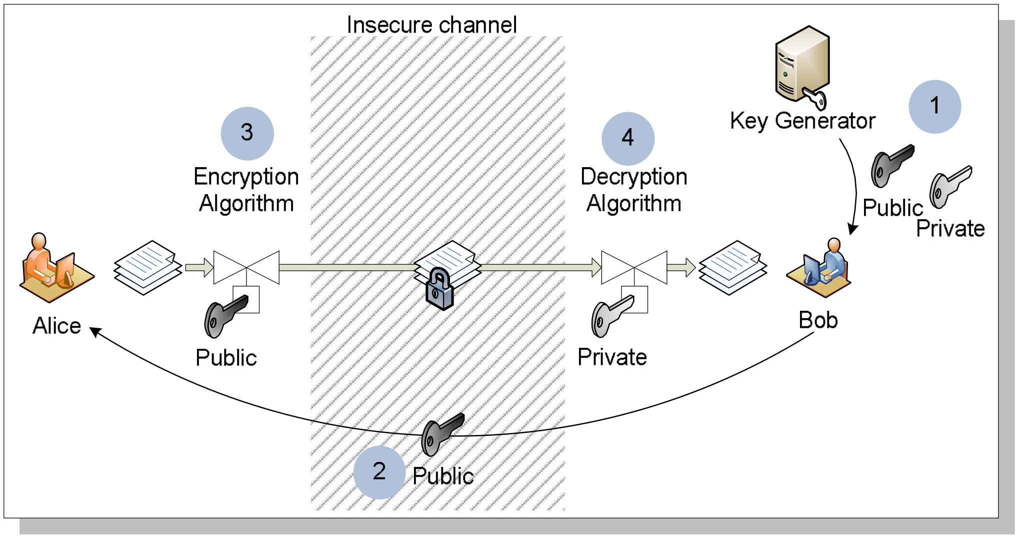


Figure 53: Public Key Cryptosystem

An example of PKC is RSA (Rivest, Shamir, & Adleman, 1978), from the names of its authors Rivest, Shamir, and Adleman. RSA is based on the difficulty of factoring the product of large integers. To use the RSA scheme, one needs to generate a public modulus $N = p \cdot q$ (where p and q are large prime numbers), a public exponent e, and a secret exponent d. The encryption of a message, m, is therefore $c = m^e \bmod N$, and the decryption process is $m = c^d \bmod N$.

Public Key Infrastructures (PKIs) have been introduced to make it possible to provide security services on the basis of PKCs. A PKI enables a security environment through a set of policies used to integrate and manage all the security parameters suitable for a great number of services, such as the authentication of entities, digital signature, secure communication between entities (learners, school partners, employers, etc.). A PKI aims at managing certificates and pairs of secret and public keys, including the ability to issue, maintain, recover and revoke public key certificates. PKIs make use of Certification Authorities (CAs), which are trusted entities whose central responsibility is certifying the authenticity of users and their public keys. More precisely, a user certificate issued and signed by a CA acts as proof that the legitimate public key is associated with the user.

Digital Signature and Blind Signature

In (Diffie & Hellman, 1976), Diffie and Hellman also introduced the related notion of digital signatures, by which the party receiving a message can ascertain the identity of the sending party as well as the integrity of the message. This process works with two keys as well: A secret signing key is used on the message to generate its signature, but anyone can use the corresponding signature verification key to make sure that the message is legitimate and that it has not been modified in transit. In our work, we consider that the signing key is the entity's private key and the corresponding verification key is the entity's public key.

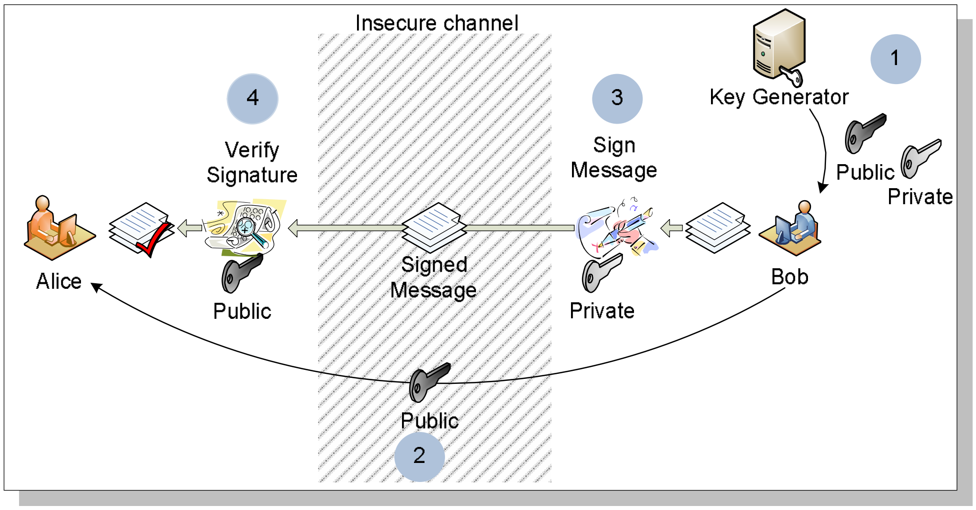


Figure 54: Digital Signature System

A blind signature scheme is a digital signature scheme which allows the signer to sign a message, u, without knowing anything about u. The main purpose of blind signatures is for applications where the privacy of the entity requesting the signature (the learner in our case) needs to protect its privacy. In our work, we use the blind signature scheme based on the RSA digital signature. To use the RSA scheme, one needs to generate a public modulus $N = p \cdot q$ (where p and q are large prime numbers), a public exponent e, and a secret exponent d. In this work, we use the blind RSA signature scheme described hereafter, where ES stands for the E-learning System in which the learner is enrolled, and all the computations are performed mod N.

RSA blind signature (u : thelearnermessage)

- N, e, d: the ES's RSA parameters
- Output: s, a signature on u.

1. The learner chooses a random value r, such that GCD(r, N) = 1. He then computes $t = u{\cdot}r^e$, and then sends t to the ES.

2. The ES computes $t' = t^d = (u \cdot r^e)^d = u^d \cdot r$, and then sends t' to the learner.

3. The learner removes the blinding factor, r, by multiplying t' by the inverse, r^{-1}, of r (such an inverse exists since GCD(r, N) = 1). Thus, he obtains the valid RSA signature on u: $s = S_d(u) = t' \cdot r^{-1} = u^d$, such that anybody can check the validity of s using the ES's public key e.

6.5 Anonymous Credentials for Privacy-Preserving E-learning

The ACES system consists of a set of protocols for a learner to obtain and show anonymous credentials from/to an E-learning system.

6.5.1 Blind Signature on a Pseudonym

Suppose a learner, L, who wants to enroll in a new ES. For this purpose, he requires some credentials from the ES, E_0, where he performed some learning activities in the past. Before enrolling in the new ES, say E_1, he chooses a pseudonym u, to be used within E_1. Thus, any credential obtained from E_0, and to be presented to E_1 must be linked to u, without neither revealing u to E_0 nor L to E_1. In fact, revealing both L and u to either E_0 or E_1 enables the constitution of a dossier infringing the learner's privacy. Therefore E_0

blindly signs the pseudonym *u* along with the appropriate credentials (see below), using the RSA blind signature (Section 3.3.2)

6.5.2 Obtaining Anonymous Credentials

We define anonymous credentials at the following levels: Learner, Course, Transcript, and Degree. In the remainder of this section ES stands for the E-learning system issuing the anonymous credentials, and EE stands for the external entity to which the issued credentials are intended.

Anonymous Learner Credential (ALC)

The Anonymous Learner Credential is used to prove to an External Entity (EE) that the learner has been, or is still enrolled in a given ES, even if his identity is unknown. This is similar to obtaining a registration certificate from a College or University. In other words, the ES delivers a letter saying that the learner, *L*, is currently, or has been a student in its institution. The Anonymous Learner Credential consists of two kinds of data: the message, *m*, describing the registration certificate, and a pseudonym that the learner intends to use within the EE, which could be another ES or organization. The Anonymous Learner Credential is generated as follows (the || symbol denotes concatenation):

1. The learner creates a pseudonym, u, in the EE.

2. The learner, identified by L in the ES, requests from the ES to digitally sign m, the contents of the registration certificate.

3. The ES signs m as $S_{SK}(m)$, and sends the result to the learner.

4. The learner forms $M = u||m||S_{SK}(m)$, and asks the ES to appose a blind signature on M.

5. The ES blindly signs M, obtains s = RSA_blind_signature(M), and then sends s, which is effectively the Anonymous Learner Credential, to the learner.

The remaining anonymous credentials are generated in a similar manner as illustrated in Figure 55.

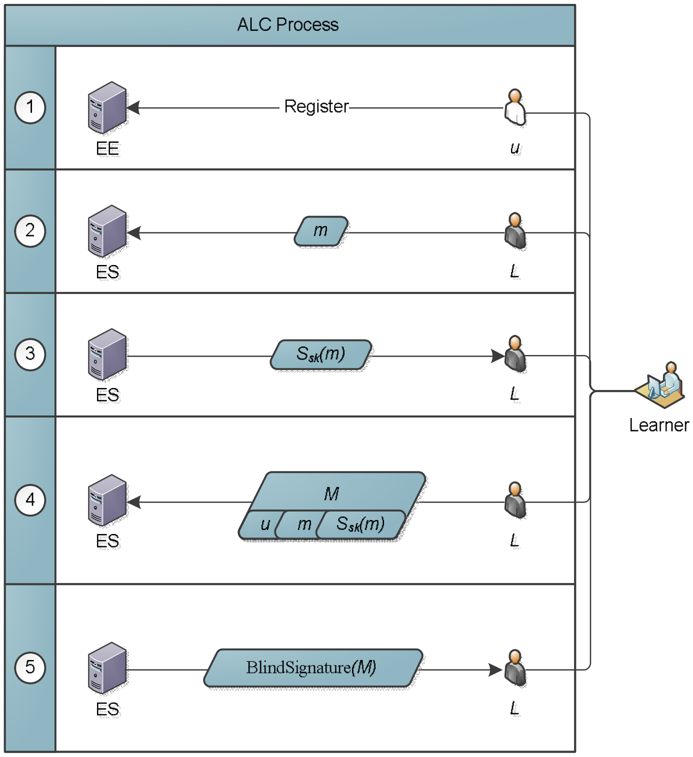


Figure 55: ALC process

It is important to use a secure communication channel (such as by using SSL) between the ES and the learner such as to prevent an intruder, Eve, from intercepting the $S_{SK}(m)$ at step 3 from the protocol, and performing steps 4 and 5 instead of the actual learner L, thus successfully impersonating him.

Anonymous Course Credential (ACC)

The Anonymous Course Credential is used to prove to an EE that the learner has successfully completed a given course within the ES, even if the learner's identity is unknown. This credential is generated as follows:

1. At the end of the course session, the ES generates a short transcript from the grades obtained from all the activities related to the course. Without loss of generality, we consider that the short transcript consists of the following data: the course identifier (*course_id*), the *semester*, and the *grade*. The course credential is therefore defined as: C = cred[course_id, semester, grade]. The ES signs C as $C' = S_{SK}(C)$, using its private key, SK, and then sends C and C' to the learner.

2. The learner checks the signature C' on C, using the ES's public key, PK. If the checking operation is successful, the learner stores (C, $S_{SK}(C)$) in his credentials database.

3. The learner creates a pseudonym, u, in the EE.

4. The learner forms and sends the message $M = u||C||S_{SK}(C)$ to the ES for a blind signature.

5. The ES blindly signs M, obtains s = RSA_blind_signature(M), and then sends s, which is effectively the Anonymous Course Credential, to the learner.

The Anonymous Transcript Credential is used to prove to an EE that the learner has successfully completed more than one course within the ES without revealing the learner's identity. The Anonymous Transcript Credential is generated as follows:

1. At the end of the semester, the learner obtains a set of anonymous course credentials: $T = (s_1, \ldots, s_v)$, where s_i is the Anonymous Course Credential of course i, and v is the number of courses that were taken by the learner within the same ES. T is effectively the Anonymous Transcript Credential.

6.5.2.1 Anonymous Degree Credential (ADC)

The Anonymous Degree Credential is used to prove to the EE that the learner has obtained a degree related to his learning activities within the ES without revealing the learner's identity. The Anonymous Degree Credential is granted when the learner has successfully passed all the courses necessary for obtaining the degree. The Anonymous Degree Credential consists of two kinds of data: the message, *m*, describing the degree, and a random number *u* that the learner intends to use as an authentication tag.

1. The learner provides all the necessary Anonymous Course Credentials for obtaining the Anonymous Degree Credential to the ES.

2. The ES verifies the various Anonymous Course Credentials.

3. If the verification process is successful

> **(a)** The ES forms the message m = (*degree_title*, *year*, *description*), which constitutes the contents of the degree.
>
> **(b)** The learner chooses a random number, u, and invites the ES to digitally sign m.
>
> **(c)** The ES signs m as $S_{SK}(m)$, and sends the result to the learner.
>
> **(d)** The learner forms $M = u||m|| S_{SK}(m)$, and requests the ES to appose a blind signature on M.
>
> **(e)** The ES blindly signs M, obtains s = RSA_blind_signature(M), and sends s, which is effectively the Anonymous Degree Credential, to the learner.

4. If the verification process was not successful, the learner is invited to obtain the needed additional Anonymous Course Credential(s) before applying for the Anonymous Degree Credential.

6.5.3 Prevention of Anonymous Credentials Sharing

Each anonymous credential is identified by a pseudonym, *u*, secretly chosen by the learner. After validating the anonymous credential, the EE inserts the pseudonym in a public Revocation of Anonymous Credentials List (RACL). The Revocation of Anonymous Credentials List contains all the anonymous credentials that have been shown to different EEs, such that no anonymous credential can be used twice. In particular, the validation of a given anonymous credential includes two steps:

- The verification of ES's signature
- The search in the Revocation of Anonymous Credentials List for a duplicate of *u*

This approach successfully prevents two learners from sharing the same credential issued by a certain ES. Nonetheless, it does not prevent the case were one learner gives away a credential. That is, learner Alice receives a properly deserved anonymous credential, which she gives to Bob who can now *uniquely* use this credential.

6.5.4 Showing an Anonymous Credential

After obtaining an anonymous credential from the ES, the learner provides it to an EE, which then checks the validity of the anonymous credential by using the ES's public key to verify that it has been properly signed by the ES. If the anonymous credential is valid, the EE then verifies that it is not part of the Revocation of Anonymous Credentials List.

6.5.5 Legitimacy of Anonymous Credentials

To prove that a given learner is the legitimate owner a set of anonymous credentials, he needs to create a Blind Digital Certificate (BDC) for each entity he deals with. A BDC is a digital certificate that doesn't reveal the learner's personal information. As such let *P* be the learner's personal information. The learner generates a pair of public/private key (*PK*, *SK*), then encrypts *P* as $y=E_{PK}(P)$ and obtains a digital certificate on $z=[y, PK]$ from a

CA. *z* is a blind digital certificate, and *y* becomes a digital identity that the learner can use within an ES.

If the learner requires proving to his manager, for example, that he did completed the training/course in the ES successfully, the blind digital certificate can be “revealed” to his manager, while maintaining the competitive or personal reasons mentioned in Section 2.1. The required document is presented to the manager, who verifies its authenticity using the ES public key. The blind digital certificate is revealed by the learner who decrypts *y* into *P* using his private key, thus allowing the manager to check again that $E_{PK}(P)=y$.

In the case of a confidential training, the learner’s supervising entity can directly act as the CA, thus providing the learner with a unique pseudonym *y*. In short, the blind digital certificate acts as a means to ensure that the learner is the legitimate owner of a given anonymous credential.

6.6 Discussion

In the previous sections, we introduced various privacy considerations. However, we only focused on the case of hard privacy and average tracking. The full privacy option is more challenging and requires more cryptographic tools. For instance, in the case of full privacy, not only has the E-learning system no information about the courses currently taken by the learner, but the system must also evaluates the learner’s activities for these unknown courses! Nonetheless, this can be achieved by performing the computations with encrypted functions (CEF) (Sander & Tschudin, 1998). However, we leave the adaptation of the CEF technique, as well as a prototype for a more complete privacy-preserving E-learning system for future work.

In addition, in this work we provided tools and methods to protect learner privacy within an E-learning system. Admittedly, there are other aspects to consider: since most E-learning systems are web-based, a learner could be easily tracked through his IP address, thus violating his privacy. However, this issue can be addressed by using well-known anonymous Web surfing systems. In more general context, there is a need to address the privacy issues related to tracking. However, we also leave this for future work.

Moreover, although our proposed approach, ACES, is technically sound, there is still work to be done in this field. Indeed, in this work we considered the case of Hard Privacy with Average Tracking. Although ACES can be easily adapted to the other privacy and tracking levels, considerations are due to the complexity of the protocol, and the implied overhead with the higher levels of privacy and tracking.

Chapter 7 : Conclusions and future work

E-learning emerged over 20 years ago, and was merely book like text displayed on a computer screen. With the changes and advances in technology, E-learning has come a long way, providing interactive rich content, the ability to personalize the learning experience, even the possibility to pursue a complete accredited education online at a virtual university. Today, E-learning is again going through major changes. Indeed, with the proliferation of E-learning systems and content authoring tools, as well as established standards, it has become easier to actually share and reuse learning content instead of just recreating what already exists. Moreover, learning content that has been used and evaluated is imperative in many cases for personalization, such as when using Item Response Theory. Moreover, with the shift to learner centered education and the effect of Web2.0 techniques and technologies, learners are no longer just recipients of the learning content, but can play an active role into enriching such content. Additionally, with the amount of information E-learning systems can gather about learners, and the impact this has on their privacy, concerns are being raised in order to protect learners' privacy. Nonetheless, to the best of our knowledge, there hasn't been any work to address the various facets of these issues. In this work, we address these issues by presenting Cadmus, SHAREK, and privacy preserving E-learning. Specifically, Cadmus is an IMS QTI compliant web based assessment authoring tool, offering the proper framework to enable tutors author and share questions and exams. Moreover, Cadmus provides functionalities such as the EQRS (Exam Questions Recommender System) to help tutors locate suitable questions, ICE (Identification of Conflicts in Exams) to help resolve conflicts between questions within the same exam, and the topic tree, designed to help tutors better organize their exam questions and easily ensure the content coverage of their exams. The various components of Cadmus have been successfully tested, where the EQRS was tested by a total of 33 different authors/users, and provided good recommendations: with an average of 40 questions in each recommendation, more than half of the questions selected by the tutors where ranked within the top 10. On the other hand, ICE reported conflicts with an accuracy of 83%, a very satisfactory result considering the short size of questions, and the high similarity in the textual content.

Moreover, the implementation of the topic tree enhanced the accuracy of ICE, effectively increasing its accuracy from 83% to almost 90%. As future work, we are considering adapting the EQRS in a more general context in E-learning, for recommending learning resources in general and not just exam questions; specifically adapting a multi-criteria recommender system approach to further enhance the accuracy of the recommendation.

On the other hand, SHAREK (Sharing REsources and Knowledge) provides the framework to take advantage of both the rigidity of E-learning systems and the flexibility of PLEs (Personal Learning Environment) while enabling learners to enrich the learning content, and helping them locate new learning resources. Specifically, in order to promote new learning resources and help learners locate suitable content, SHAREK utilizes a multi-criteria content based recommender system, and combines Web2.0 technologies and techniques such as RSS and social web. Specifically, we propose and test different approaches in the multi-criteria recommender systems to determine the most accurate approach. SHAREK was successfully evaluated with the help of 93 participants, where the evaluations of its functionalities were very favorable. Indeed, most learners reported they regularly access and use resources from outside the classroom, and are likely to use learning material recommended by friends and peers, as well as share/recommend their own resources. Moreover, most respondent were satisfied with the functionalities provided by SHAREK, and reported that such an environment would encourage them to further share their knowledge. Still there are some aspects that still require some future work. Specifically, further validation of the multi-criteria recommender system, on a real dataset, is required to further validate and to further enhance our approach.

Lastly, in order to address the various needs for privacy in E-learning, we propose a framework with four levels of privacy, and four levels of tracking. When presented with the possibility to choose a privacy level and a tracking level, most learners opted to protect their privacy, selecting one of the three privacy levels (as opposed to No privacy). Additionally, in order to achieve the various privacy and tracking levels, we propose ACES (Anonymous Credentials for E-learning Systems), a set of protocols, based on well established cryptographic techniques, which allows the learner to perform an essential function within a privacy preserving E-learning environment: obtain and show anonymous credentials from and to an E-learning system. Although there was no *technical evaluation* of ACES, we did demonstrate the validity of the protocol. Moreover, in order to evaluate the impact of privacy on learners, we conducted an experiment with 77 participants, and the results indicated that privacy actually had a positive impact on learners. Indeed, the study indicated that learners tend to perform better in a privacy enhanced environment. Yet, the results include a bias, due to the fact that the study was conducted outside a classroom, where there was no real consequence to performing poorly. As future work, more validation, in the context of an actual course, is necessary to further evaluate the impact of privacy on learners' attitudes. Additionally, we intend to implement the ACES protocol and evaluate the overhead it introduces to the E-learning process.

References

ACM. (1998). The ACM Computing Classification System. Retrieved 04, 2007, from http://www.acm.org/class/1998/

Adomavicius, G., & Kwon, Y. (2007). New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems, 22*(3), 48-55.

Aïmeur, E., Brassard, G., Fernandez, J. M., & Mani Onana, F. S. (2007a). ALAMBIC: A Privacy-Preserving Recommender System for Electronic Commerce. *ACM Transactions on Internet Technology*.

Aïmeur, E., Hage, H., & Mani Onana, F. S. (2007b). A Framework for Privacy-Preserving E-learning. *Joint Itrust and PST conferences on Privacy, Trust Management and Security (IFIPTM 2007)* (pp. 223-238), Moncton.

Aïmeur, E., Hage, H., & Mani Onana, F. S. (2008). Anonymous Credentials for Privacy-Preserving E-learning. *The Montreal Conference on eTechnologies 2008 (MCETECH2008)* (pp. 70-80), Montreal.

Aïmeur, E., ManiOnana, F. S., & Saleman, A. (2006). SPRITS: Secure Pedagogical Resources in Intelligent Tutoring Systems. *Intelligent Tutoring Systems, 8th International Conference (ITS 06)* (pp. 237-247), Jhongli, Taiwan.

Aktas, M. S., Pierce, M., Fox, G. C., & Leake, D. (2004). A Web based Conversational Case-Based Recommender System for Ontology aided Metadata Discovery. *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing* (pp. 69- 75).

Alexa. (2008). Retrieved January, 2008, from http://www.alexa.com

Anwar, M., & Greer, J. (2009). Implementing Role-and Relationship-based Identity Management in E-learning Environments. *14th International Conference on Artificial Intelligence in Education (AIED 2009)* (pp. 608-610), Brighton.

Arroyo, I., & Woolf, B. P. (2005). Inferring learning and attitudes from a Bayesian Network of log file data. *International Conference on Artificial Intelligence in Education (AIED 2005)* (pp. 33–40), Amsterdam.

Aysoon.com. Retrieved January, 2008, from http://blog.aysoon.com/

Bade, K., Hullermeier, E., & Nurnberger, A. (2006). Hierarchical Classification by Expected Utility Maximization. *Sixth International Conference on Data Mining (ICDM '06)* (pp. 43-52), Hong Kong.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retreival. Reading, MA: Addison Wesley.

Baghaei, N., & Mitrovic, A. (2006). A Constraint-Based Collaborative Environment for Learning UML Class Diagrams. *Intelligent Tutoring Systems, 8th International Conference (ITS 06)* (pp. 176-186), Jhongli, Taiwan.

Baker, F. (2001). The Basics of Item Response Theory. College Park, MD.: ERIC Clearinghouse on Assessment and Evaluation, University of Maryland.

BBC. (2009a). Hackers hit Twitter and Facebook. *BBC News*. Retrieved from http://news.bbc.co.uk/2/hi/8188201.stm

BBC. (2009b). US man 'stole 130m card numbers' *BBC News*. Retrieved from http://news.bbc.co.uk/2/hi/americas/8206305.stm

Beck, J. E. (2007). Does learner control affect learning? 13th International Conference on Artificial Intelligence in Education (AIED 07).

Blackboard. (2004). *Blackboard Academic Suite (Release 6.1) Instructor Manual*: Blackboard Inc.

Blanchard, E., Chalfoun, P., & Frasson, C. (2007). Towards Advanced Learner Modeling: Discussions on Quasi Real-time Adaptation with Physiological Data. 7th IEEE conference on Advanced Learning Technologies (ICALT 2007).

Blanchard, E., & Frasson, C. (2005). Making Intelligent Tutoring Systems culturally aware: The use of Hofstede's cultural dimensions. International Conference in Artificial Intelligence (ICAI2005).

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., & Orchard, D. (2004). Web Services Architecture. Retrieved February, 2009, from http://www.w3.org/TR/ws-arch/

Brands, S. (2000). Rethinking Public Key Infrastructure and Digital Certificates – Building in Privacy. Cambridge: MIT Press.

Brooks, C. A., Panesar, R., & Greer, J. E. (2007). Awareness and Collaboration in the iHelp Courses Content Management System. *European Conference on Technology Enhanced Learning* (pp. 34-44), Crete, Greece.

Brusilovsky, P., & Miller, P. (1999). Web-based testing for distance education. *WebNet 99 World Conference on the WWW and Internet Proceedings* (pp. 149-154), Honolulu, Hawaii, U.S.A.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction 12*(4), 331-370.

Burke, R. (2004). Hybrid Recommender Systems with Case-Based Components. *Advances in Case-Based Reasoning, 7th European Conference (ECCBR 2004)* (pp. 91-105), Madrid.

Butler, D. L. (2003). The Impact of Computer-Based Testing on Student Attitudes and Behavior. *The Technology Source* Retrieved July, 2006, from http://technologysource.org/article/impact_of_computerbased_testing_on_student_attitudes_and_behavior/

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Znasi, A. (1997). Discovering Data Mining: From Concept To Implementation. Upper Saddle River, NJ: Prentice-Hall.

Camenisch, J., & Lysyanskaya, A. (2001). An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. *Advances in Cryptology – EUROCRYPT 2001* (pp. 93–118), Innsbruck.

Canny, J. (2002a). Collaborative Filtering with Privacy. *IEEE Conf. on Security and Privacy* (pp. Oakland.

Canny, J. (2002b). Collaborative Filtering with Privacy via Factor Analysis. *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 238-245), Tampere.

Chaum, D. (1985). Security without identification: Transaction systems to make Big Brother obsolete. *Communications of the ACM, 28*(10), 1030–1044.

Chaum, D., & Evertse, J. (1987). A Secure and Privacy-protecting Protocol for Transmitting Personal Information Between Organizations. *Advances in Cryptology – CRYPTO '86* (pp. 118–167), Santa Barbara.

Cios, K. J., Pedrycz, W., Swiniarski, R. W., & Kurgan, L. A. (2007). Data Mining: A Knowledge Discovery Approach. London: Springer.

de Vicente, A., & Pain, H. (1998). Motivation Diagnosis in Intelligent Tutoring Systems. 4th International Conference On Intelligent Tutoring Systems (ITS 1998). Retrieved from http://dx.doi.org/10.1007/3-540-68716-5_14

de Vicente, A., & Pain, H. (2002). Informing the Detection of the Students' Motivational State: An Empirical Study. 6th International Conference On Intelligent Tutoring Systems (ITS2002). Retrieved from http://dx.doi.org/10.1007/3-540-47987-2_93

Deshpande, M., & Karypis, G. (2004). Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems, 22*(1), 143-177.

Desmarais, M. C., & Pu, X. (2005). A Bayesian Inference Adaptive Testing Framework and its comparison with Item Response Theory. *International Journal of Artificial Intelligence in Education, 15*, 291-323.

Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*(6), 644–654.

Dolog, P., Henze, N., Nejdl, W., & Sintek., M. (2004). Personalization in Distributed eLearning Environments. *13th World Wide Web Conference* (pp. 170-179), New York.

EduTools. (2006). EduTools Homepage. Retrieved 09, 2006, from http://www.edutools.info/index.jsp?pj=1

Flaherty, G. (1992). The learning curve. *Vocational Education Journal, 67*(6), 32-56.

Franz, E., Wahrig, H., Boettcher, A., & Borcea-Pfitzmann, K. (2006). Access Control in a Privacy-Aware eLearning Environment. *International Conference on Availability, Reliability and Security (ARES 2006)* (pp. 879–886), Vienna.

Gaudiosi, E., & Boticario, J. (2003). Towards web-based adaptive learning community. *International Conference on Artificial Intelligence in Education (AIED 2003)* (pp. 237-244), Sydney.

Gelbukh, A., Sidorov, G., & Guzman-Arenas, A. (1999). Document comparison with a weighted topic hierarchy. *Tenth International Workshop on Database and Expert Systems Applications* (pp. 566-570), Florence.

Hage, H., & Aïmeur, E. (2005). Exam Question Recommender System. *International Conference on Artificial Intelligence in Education (AIED 2005)* (pp. 249-257), Amsterdam.

Hage, H., & Aïmeur, E. (2006a). ICE: A System for Identification of Conflicts in Exams. *The 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06)* (pp. 980–987), Dubai.

Hage, H., & Aïmeur, E. (2006b). Using Information Retrieval for Conflict Detection and Resolution in Exams. *Journal of e-Learning and Knowledge Society, 2*(1), 111-121.

Hage, H., & Aïmeur, E. (2007). Topic Tree: Increasing the Accuracy of Item Retrieval. *World Conference on E-learning in Corporate, Gouverment, Healthcare, and Higher Education (E-learn 2007)* (pp. 6933-6940), Quebec.

Hage, H., & Aïmeur, E. (2008a). Combining Web2.0 and E-learning: enabling learners to enrich the learning content. The International Group of e-Systems Research and Applications (TIGERA 2008).

Hage, H., & Aïmeur, E. (2008b). Harnessing learner's collective intelligence: a Web2.0 approach to E-learning. *9th International Conference on Intelligent Tutoring Systems (ITS 2008)* (pp. 438-447), Montreal.

Hage, H., & Aïmeur, E. (2009). The impact of privacy on learners in the context of a web-based test. 14th International Conference on Artificial Intelligence in Education (AIED 2009).

Hage, H., & Aïmeur, E. (2010a). E-learning for the new generations, a Web2.0 approach. In M. Buzzi (Ed.), *E-Learning* (pp. 1-18): INTECH.

Hage, H., & Aïmeur, E. (2010b). Preserving Learners' Privacy in E-learning. In R. Mizoguchi, J. Bourdeau & R. Nkambou (Eds.), *Advances in Intelligent Tutoring Systems* Springer Verlag.

Howard, G. (1998). *Identifying learning styles*. Montgomery WV: The Annual Summer Workshop for beginning Vocational Education Teachers.

IEEE Learning Technology Standards Committee. (2002). *Standard for Learning Object Metadata*.

IEEE Learning Technology Standards Committee. (2003). *Standard for Learning Technology - Learning Technology Systems Architecture (LTSA)*.

IMS-QTI. (2006). IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model. Retrieved October 21, 2006, from http://www.imsglobal.org/question/qtiv2p1pd2/imsqti_infov2p1pd2.html

IMS. IMS Global Learning Consortium. Retrieved September, 2007, from http://www.imsproject.org/

IMS. (2006). IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model. Retrieved October 21, 2006, from http://www.imsglobal.org/question/qtiv2p1pd2/imsqti_infov2p1pd2.html

Israel, J., & Aiken, R. (2007). Supporting Collaborative Learning With An Intelligent Web-Based System. *International Journal of Artificial Intelligence in Education, 17*(1), 3-40.

Jerman-Blazic, B., & Klobucar, T. (2005). Privacy provision in e-learning standardized systems: status and improvements. *Computer Standards & Interfaces, 27*(6), 561-578.

Jordan, P., Makatchev, M., & Vanlehn, K. (2003). Abductive Theorem Proving for Analyzing Student Explanations. *International Conference on Artificial Intelligence in Education (AIED 2003)* (pp. 73-80), Sydney.

Jovanovic, J., Gasevic, D., & Devedzic, V. (2006). Ontology-Based Automatic Annotation of Learning Content. *International Journal on Semantic Web & Information Systems, 2*(2), 91-119.

Keefe, J. W. (1979). *Learning Style: An Overview. In Student Learning Styles: Diagnosing and Prescribing Programs*. Reston, VA.: National Association of Secondary School Principals.

Lau, R., Ip, R., Chan, M., Kwok, R., Wong, E., Wong, S., & So, J. (2010). Podcasting: An Internet-Based Social Technology for Blended Learning. *Internet Computing, IEEE, PP*(99), 1-1.

Lee, J. (2009). Cyberattack rocks South Korea. *GlobalPost*. Retrieved from http://www.globalpost.com/dispatch/south-korea/090710/cyberattacks

Lenhart, A., Madden, M., Macgill, A. R., & Smith, A. (2007). Teens and Social Media. Retrieved from http://www.pewinternet.org/PPF/r/230/report_display.asp

Lidong, C. (1995). Access with Pseudonyms. *International Conference on Cryptography: Policy and Algorithms* (pp. 232-243), Brisbane.

Lin, N. H., Korba, L., Yee, G., Shih, T. K., & Lin, H. W. (2004). Security and privacy technologies for distance education applications. *18th International Conference on Advanced Information Networking and Applications (AINA 2004)* (pp. 580-585).

LiNe-Zine. (2000). Learning in the new economy. Retrieved 08, 2006, from http://www.linezine.com/elearning.htm

LTSC. Learning Technologies Standardization Committee (LTSC). Retrieved September, 2007, from http://www.ieeeltsc.org/

Lysyanskaya, A., Rivest, R. L., Sahai, A., & Wolf, S. (1999). Pseudonym Systems. *Selected Areas in Cryptography: 6th Annual International Workshop, SAC'99* (pp. 184–199), Kingston.

Ma, W. (2005). Learning Object Recommender Systems. *IASTED International Conference on Education and Technology* (pp. 113-118), Calgary.

Mabrouk, M. (2006). *UMAKE : Adaptation et recommandation d'outils d'aide d'un quiz pour l'auto-évaluation.* Université de Montréal, Montreal, Qc.

Maron, M. E., & Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM, 7*(3), 216-244.

Mathan, S., & Koedinger, K. R. (2003). Recasting the Feedback Debate: Benefits of Tutoring Error Detection and Correction Skills. *International Conference on Artificial Intelligence in Education (AIED 2003)* (pp. 13-20), Australia.

Mbendera, A. J. K., C.; Sun, L. (2010). Towards Development of Personalised Knowledge Construction Model for e-Learning. *Second International Conference on Mobile, Hybrid, and On-Line Learning (ELML 2010)* (pp. 29-35).

Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for Improved recommendations. Eighteenth National Conference on Artificial Intelligence.

Merkle, R. C. (1978). Secure communications over insecure channels. *Communications of the ACM, 21*(4), 294-299.

Messmer, E. (2010). DDoS attacks, network hacks rampant in oil and gas industry, other infrastructure sectors *Network World*. Retrieved from http://www.networkworld.com/news/2010/012710-ddos-oil-gas.html

Miller, B., Konstan, J., & Riedl, J. (2004). PocketLens: Toward a Personal Recommender System. *ACM Transaction on Information Systems, 22*(3), 437-476.

Naak, A., Hage, H., & Aïmeur, E. (2008). Papyres: a Research Paper Management System. IEEE Joint Conference on E-Commerce Technology (CEC 08) and Enterprise Computing, E-Commerce and E-Services (EEE 08).

Naak, A., Hage, H., & Aïmeur, E. (2009). A Multi-criteria Collaborative Filtering Approach for Research Paper Recommendation in Papyres. *4th International MCETECH Conference on e-Technologies (MCETECH 2009)* (pp. 25-39), Ottawa.

O'Dell, J. (2009). RockYou Hacker: 30% of Sites Store Plain Text Passwords *The New York Times*. Retrieved from http://www.nytimes.com/external/readwriteweb/2009/12/16/16readwriteweb-rockyou-hacker-30-of-sites-store-plain-text-13200.html

O'Reilly, T. (2005). What Is Web 2.0. Retrieved from http://www.oreillynet.com/

O' Regan, K. (2003). Emotion and E-Learning. *Journal of Asynchronous Learning, 7*(3), 78-92.

Paskey, E. M. (2001). Employing a Web-Based Assessment System to Obtain the Best IT Professionals for the Federal Government. *Assessment Council News*, 3-5.

Porter, D., Curry, J., Muirhead, B., & Galan, N. (2002). *A Report on Learning Object Repositories*: CANARIE & Industry Canada.

Pusnik, M., Sumak, B., & Hericko, M. (2010). Investigation of Virtual Learning Environment in the Context of Web 2.0. *Mobile, Hybrid, and On-Line Learning, 2010. ELML '10. Second International Conference on* (pp. 1-6).

Raitman, R., Augar, N., & Zhou, W. (2005a). Employing Wikis for Online Collaboration in the E-Learning Environment: Case Study. *3rd International Conference on Information Technology and Applications (ICITA 2005)* (pp. 142-146), Sydney.

Raitman, R., Ngo, L., Augar, N., & W.Zhou. (2005b). Security in the online e-learning environment. *IEEE International Conference on Advanced Learning Technologies (ICALT 2005), 5*(8), 702–706.

Rivest, R., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM, 21*(2), 120–126.

Rudner, L. M. (1998). An On-line, Interactive, Computer Adaptive Testing Tutorial. Retrieved 10, 2006, from http://edres.org/scripts/cat/catdemo.htm

Salton, G., & McGill, M. (1983). Introduction to modern information retrieval New York, NY: McGraw-Hill.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for information retrieval. *Communications of the ACM, 18*(11), 613-620.

Sander, T., & Tschudin, C. (1998). Towards mobile cryptography. *IEEE Symposium on Security and Privacy* (pp. 215–224), Oakland.

Schauer, B. (2005). Crucial DNA of Web 2.0. Retrieved from http://adaptivepath.com/ideas/essays/archives/000547.php

Sclater, N., & Low, B. (2002). IMS Question and Test Interoperability: An Idiot's Guide. Retrieved october 21, 2006, from http://www.scrolla.ac.uk/resources/s2/idiots_guide.pdf

Shim, J. P., Shropshire, J., Park, S., Harris, H., & Campbell, N. (2007). Podcasting for e-learning, communication, and delivery. *Industrial Management & Data Systems, 107*(4), 587-600.

Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin, 24*(4), 35-43.

Soller, A., Martínez-Monés, A., Jermann, P., & Muehlenbrock, M. (2005). From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education, 15*(4), 261-290.

Standish, T. A. (1998). Data Structures in Java: Addison-Wesley Longman Publishing Co., Inc.

Takeuchi, M., Hayashi, Y., Ikeda, M., & Mizoguchi, R. (2006). A Collaborative Learning Design Environment to Integrate Practice and Learning Based on Collaborative Space Ontology and Patterns. *Intelligent Tutoring Systems, 8th International Conference (ITS 06)* (pp. 187-196), Jhongli, Taiwan.

Tang, T., & McCalla, G. (2003). Smart Recommendation for an Evolving E-Learning System. *International Conference on Artificial Intelligence in Education (AIED 2003)* (pp. 699-710), Sydney.

Tao, L., & Ogihara, M. (2005). Music genre classification with taxonomy. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)* (pp. v/197-v/200).

Tsai, K. H., Chiu, T. K., Lee, M. C., & Wang, T. I. (2006). A Learning Objects Recommendation Model based on the Preference and Ontological Approaches. *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp. 36-40).

Vassileva, J. (2004). Harnessing P2P Power in the Classroom. *Intelligent Tutoring Systems, 7th International Conference (ITS 04)* (pp. 305-314), Alagoas, Brazil.

Ventura, M. J., Franchescetti, D. R., Pennumasta, P., Graesser, G. T., Jackson, G. T., Hu, X., Cai, Z., & Group, t. T. R. (2004). Combining Computational Models of Short Essay Grading for Conceptual Physics Problems. *Intelligent Tutoring Systems, 7th International Conference (ITS 2004)* (pp. 423-431), Brazil.

Walker, A., Recker, M., Lawless, K., & Wiley, D. (2004). Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence and Education, 14*, 3-28.

WebCT. (2003). *Getting Started Guide: WebCT Campus Edition 4.1*: WebCT, Inc.

Webilus. (2008). Retrieved January, 2008, from http://webilus.com/toutes-les-images

Weiss, M. A. (1999). Data Structures and Algorithm Analysis in Java: Addison-Wesley Longman Publishing Co., Inc.

Westin, A. (1967). Privacy and Freedom. New York, NY: Atheneum.

Wikipedia. (2010). Wikipedia, the free encyclopedia. Retrieved March, 2010, from http://en.wikipedia.org/wiki/Wikipedia:About

Yee, G., & Korba, L. (2003). The Negotiation of Privacy Policies in Distance Education. *IRMA International Conference* (pp. Philadelphia.

Zeller, W., & Felten, E. W. (2008). Cross-Site Request Forgeries: Exploitation and Prevention.

Zins, J. E., Bloodworth, M. R., Weissberg, R. P., & Walberg, H. J. (2007). The Scientific Base Linking Social and Emotional Learning to School Success. *Journal of Educational and Psychological Consultation, 17*(2), 191 - 210-191 - 210.