Lin, Y-T., Lin, Y.-C., Huang, Y.-M., & Cheng, S.-C. (2013). A Wiki-based Teaching Material Development Environment with Enhanced Particle Swarm Optimization. *Educational Technology & Society*, *16* (2), 103–118.

A Wiki-based Teaching Material Development Environment with Enhanced Particle Swarm Optimization

Yen-Ting Lin[1], Yi-Chun Lin[1], Yueh-Min Huang[1*] and Shu-Chen Cheng[2]
[1]Department of Engineering Science, National Cheng Kung University, Taiwan, R.O.C. // [2]Department of Computer Science and Information Engineering, Southern Taiwan University, Taiwan, R.O.C. // ricky014@gmail.com // jellyplum@gmail.com // huang@mail.ncku.edu.tw // kittyc@mail.stut.edu.tw
[*]Corresponding Author

(Submitted October 30, 2011; Revised April 05, 2012; Accepted June 04, 2012)

ABSTRACT

One goal of e-learning is to enhance the interoperability and reusability of learning resources. However, current e-learning systems do little to adequately support this. In order to achieve this aim, the first step is to consider how to assist instructors in re-organizing the existing learning objects. However, when instructors are dealing with a large number of existing learning objects, manually re-organizing them into appropriate teaching materials is very laborious. Furthermore, in order to organize well-structured teaching materials, the instructors also have to take more than one factor or criterion into account simultaneously. To cope with this problem, this study develops a wiki-based teaching material development environment by employing enhanced particle swarm optimization and wiki techniques to enable instructors to create and revise teaching materials. The results demonstrated that the proposed approach is efficient and effective in forming custom-made teaching materials by organizing existing and relevant learning objects that satisfy specific requirements. Finally, a questionnaire and interviews were used to investigate teachers' perceptions of the effectiveness of the environment. The results revealed that most of the teachers accepted the quality of the teaching material development results and appreciated the proposed environment.

Keywords

Particle swarm optimization, Wiki-based revision, Material design

Introduction

Over the last decade, e-learning has become widely applied in the educational domain. A major aim of e-learning is to increase interoperability and reusability of learning objects. Thanks to the establishment of various standards such as IEEE Learning Object Metadata (LOM) and Sharable Content Object Reference Model (SCORM), several authoring tools have been developed to assist instructors in producing and packaging learning objects with metadata that are compliant with the standards to enhance the interoperability. For example, in 2005, García and García complied with LOM to propose an authoring tool, namely HyCo, to facilitate the composition of hypertext, which are stored as semantic learning objects in a backend database (García & García, 2005). Furthermore, Wang et al. (2007) designed a rich-client authoring environment for creating learning contents that are compatible with various e-learning standards without redundant efforts. Additionally, Kuo and Huang (2009) presented an authoring tool that can produce adaptable learning content to support both e-learning and m-learning, complying with SCORM standard. Although the above approaches significantly enhanced the interoperability of the learning objects, the support of the reusability for such learning objects is not enough.

In fact, teachers often have to design and produce individual teaching materials for specific subject matter by themselves. Moreover, a typical approach to content design consists of five stages, known as ADDIE (ADDIE, 2004), short for analysis, design, develop, implement, and evaluate, and this process requires that teachers spend a considerable amount of time and effort. Furthermore, such costs obviously increase unnecessarily when different individuals are working to develop similar teaching materials for the same course units simultaneously. Therefore, e-learning materials could be very useful resources for further education, because instructors can reuse existing learning objects to re-produce specific teaching materials more efficiently and effectively for different contexts.

As mentioned above, in order to solve this we first have to consider how to assist instructors in assembling such materials, and one problem with this is the huge amount of learning objects that may need to be considered. Furthermore, in order to form well-structured teaching materials, instructors also have to take more than one factor or criterion into account simultaneously, adding to their already challenging workload. Although previous studies have applied query expansion techniques to address the first problem, they do not take multi-criteria into account to fit the real-world situation (Jou & Liu, 2011; Shih, Tseng, & Yang, 2008).

Bearing this in mind, this study aims to develop a rapid prototyping approach by employing particle swarm optimization (PSO) with multi-criteria to accelerate the development of drafts of teaching materials, as well as utilizing wiki-based techniques to enhance the revision quality of the materials thus produced. The ultimate aim of the study is to reduce the time, effort, and cost associated with the development of high-quality teaching materials.

Background and related work

Particle swarm optimization

PSO is a population-based optimization algorithm. Kennedy and Eberhart proposed the algorithm in 1995, inspired by the social behaviors of fish schooling and bird flocking, because they thought swarm intelligence could increase both the speed and the success rate for certain processes (Kennedy & Eberhart, 1995).

To carry out the PSO, each investigator has to formulate a fitness function according to the requirements of different optimization problems. Following this, a swarm of particles is generated and then distributed over a problem space, where each particle represents a potential solution to the optimization problem and is able to "remember" its own past status. During the optimization process, the PSO algorithm quantifies the location of each particle through the fitness function, and then utilizes the velocity function to produce the next generation until the process is terminated. Simultaneously, each particle can keep track of its own coordinates in the N-dimensional problem space that are related to the optimal solution it has achieved so far.

The velocity function consists of two models, cognition-only and social-only, which are both composed of two main parameters, called personal best location (PBest) and global best location (GBest). The formulas of the velocity function are described in the following paragraphs.

Cognition-only model

$$V_{id} = V_{id} + C_1 \times rand() \times \left(P_{id} - X_{id}\right) \tag{1}$$

Social-only model

$$V_{id} = V_{id} + C_2 \times rand() \times \left(P_{gd} - X_{id}\right) \tag{2}$$

Where V_{id} is the velocity vector of the ith particle in d dimension of the problem space, P_{id} is the personal best position vector of the particle in d dimension, P_{gd} is the global best position vector of the particle in d dimension, X_{id} is the current position vector of the ith particle in d dimension, C_1 is the personal cognitive learning rate, C_2 is the social learning rate, and rand() is a random real number in [0,1].

As the velocity function relies on the social-only and cognition-only models, the following formula specifies the complete velocity function, which combines Equation (1) with Equation (2).

$$V_{id}(t+1) = V_{id}(t) + C_1 \times rand() \times \left(P_{id} - X_{id}(t)\right) + C_2 \times rand() \times \left(P_{gd} - X_{id}(t)\right) \tag{3}$$

Each particle's velocity and direction are evaluated by Equation (3), and its current position is updated through Equation (4).

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \tag{4}$$

In addition, Kennedy and Eberhart further presented a discrete binary version for the PSO algorithm in 1997 (Kennedy & Eberhart, 1997), and this is used for combinational optimization, where each particle is structured by a binary vector of length d. Moreover, the velocity of a particle is represented by the probability that a decision variable will take the value 1 to update each particle's current position. In short, the bit of a particle will be restricted to zero or one, where each V_{id} represents the probability of bit X_{id} taking the value 1.

Wiki technology

The concept of wiki was first proposed by Ward Cunningham in 1995, who used the word to name an environment he developed for co-workers to share specifications and documents for software design.

The specific functionality of a wiki is called open editing, and the inherent characteristics of this mean that such systems can be excellent tools to support group processes and to create knowledge repositories in an online environment (Leuf & Cunningham, 2001). Moreover, some investigators have suggested that wiki systems can be useful tools for building communities of practice (Lo, 2009; Shih, Tseng, & Yang, 2008). In recent years, many wiki sites have been built on the Internet, with the most famous being Wikipedia, an online open-source encyclopedia (Wikipedia, 2004). In the educational domain, many studies have been inspired by Wikipedia to investigate the effectiveness of such systems with regard to teaching and learning, as well as to develop practical approaches for online collaboration (Ebersbach, Glaser, & Heigl, 2006; Wheeler, Yeomans, & Wheeler, 2008).

Problem description

In this study, we propose an enhanced particle swarm optimization (EPSO) method to model a teaching material generation problem under different assessment criteria, and the EPSO aims at minimizing the differences between execution results and instructors' actual requirements. Three indicators are usually considered in the literature with regard to instructors developing teaching materials (Hofmann, 2004; Shih, Tseng, & Yang, 2008), and thus this study adopts these as the assessment criteria, namely: the difficulty of the material, the expected lecture time, and the relevant topics.

In this study, a learning object, in compliance with the IEEE LOM standard, is a digital entity containing a lecture about a particular topic. With regard to the difficulty and lecture time, IEEE LOM has defined two elements, namely difficulty and typical learning time, to describe these (IEEE, 2002). A five-rating scheme is used to describe the difficulty of learning objects from very easy to very difficult, while the typical time required to learn an object is obtained using an open text field that developers can enter their own responses in. According to these two elements, educators can obtain both the desired difficulty and lecture time of the learning objects, and thus better plan their courses. Generally, the difficulty and time required to learn a learning object can be determined by domain experts, but this is a time-consuming task. To cope with this problem, several studies have proposed automatic metadata generation approaches (Meire, Ochoa, & Duval, 2007; Motolet, & Baloian, 2007). Furthermore, with regard to topic relevance, several researchers have proposed various approaches that can help teachers to relate learning objects and topics (Hwang, 2003; Jong, Lin, Wu, & Chan, 2004), and educators can obtain this information in different ways, based on their specific requirements. Therefore, this study assumes that such information is already available when it works on the teaching material generation problem.

More specifically, assume there is a learning object repository (LOR) consisting of n learning objects, $O_1, O_2, \ldots, O_i, \ldots, O_n$. An instructor requires a teaching material which aims at k topics, $T_1, T_2, \ldots, T_x, \ldots, T_k$, and the lecture time is expected to range from l seconds to u seconds. Moreover, suppose the instructor requires the teaching material with a specific difficulty degree, D. Therefore, to organize the teaching material, c learning objects will be selected from the LOR. Furthermore, each learning object selected cannot be repeated in the final combinational result and is relevant to one or more of the specified topics. Naturally, the c learning objects are a subset of the n learning objects, $c \in n$. The variables used in this model are given as follows:

- n, the number of learning objects in the learning object repository
- k, the number of topics to be provided by the teaching material
- c, the number of learning objects to be selected to organize a draft of the teaching material
- O_i, $1 \le i \le n$, the i^{th} learning object in the learning object repository which consists of n learning objects
- T_x, $1 \le x \le k$, the x^{th} topic to be provided by the teaching material which aims at k topics
- s_i, $1 \le i \le n$, s_i is 1 if the O_i is organized in the draft of the teaching material, 0, otherwise
- D, $0 < D \le 1$, the target degree of difficulty for the draft generated

- d_i, $1 \le i \le n$, $0 < d_i \le 1$, the degree of difficulty of O_i
- r_{ix}, $1 \le i \le n$, $1 \le x \le k$, the degree of association between the learning object O_i and topic T_x. r_{ix} is 1 if the O_i is relevant to T_x, 0, otherwise
- e_i, $1 \le i \le n$, the expected lecture time needed for O_i
- l, the lower bound of the expected lecture time needed for the teaching material
- u, the upper bound of the expected lecture time needed for the teaching material

The formal definition of EPSO is as follows:
Minimize $Z(\mathbf{P}y) = f + C_1 + C_2 + C_3$ (5)
The *Equation* (5) is a fitness function designed for addressing this problem. The aim of this function is to minimize the difference between the learning objects selected by EPSO and the target assigned by instructors on each assessment criterion.

$$f = \left| \frac{\sum_{i=1}^{n} s_i d_i}{\sum_{i=1}^{n} s_i} - D \right| \tag{6}$$

f indicates the difference in difficulty between the selected learning objects and the target. This formula first computes the average difficulty of the selected learning objects and then further measures the difference between the average and target difficulties.

$$C_1 = \frac{\sum_{x=1}^{k} \left(1 - \frac{\sum_{i=1}^{n} s_i r_{ix}}{\sum_{i=1}^{n} s_i} \right)}{k} \tag{7}$$

C_1 represents the degree of relevance between the selected learning objects and assigned topics. The function is used to compute the average relevance degree of the selected learning objects with regard to the k-assigned topics. Moreover, in order to satisfy the fitness function, a reverse computation is designed to obtain the minimized value in this function.

$$C_2 = \max\left(\min\left(l - \sum_{i=1}^{n} s_i e_i, 1 \right), 0 \right) \tag{8}$$

$$C_3 = \max\left(\min\left(\sum_{i=1}^{n} s_i e_i - u, 1 \right), 0 \right) \tag{9}$$

C_2 and C_3 indicate that the expected lecture time needed for the selected learning objects is outside the specified lower or upper bounds. The two functions can sum up the expected lecture time of the selected learning objects and then compute the difference with the lower and upper bounds. If the expected lecture time of the selected learning objects is satisfied the lower and upper bounds respectively, the results of the two functions would be minimized.
As mentioned previously, $Z(\boldsymbol{P}_y)$ is the fitness function which consists of four assessment criteria to solve the teaching material generation problem. Since the discrete binary version of PSO is adopted in this study for combination optimization, all decision variables of the teaching material generation problem take binary values (either 0 or 1). To satisfy this, a particle can be represented by $\boldsymbol{P}_{y,i} = [s_1 s_2 \ldots s_i \ldots s_n]$, which is a vector of n binary bits, where $\boldsymbol{P}_{y,i}$ indicates the i^{th} bit of the y^{th} particle, s_i is equivalent 1 if the learning object O_i is organized in the draft of the teaching material, and 0 otherwise.

In addition, the velocity function is also a vital part of EPSO. According to the discrete version of PSO, a logistic transformation $S(v_{y,\,i})$ is used to update the velocity and position of each particle, and it is used as the probability scale in [0.0, 1.0] to determine which particle bit will have the value of 1. In this study, we apply the sigmoid function to transform velocities into probability, as follows:

$$S\left(v_{y,i}\right)=\frac{1}{1+e^{-v_{y,i}}} \tag{10}$$

Wiki-based teaching material development environment

The wiki-based teaching material development environment is a web-based system that is integrated with a LMS named ANTS (Agent-based Navigational Training System) to facilitate the teaching material generation process (Jeng, Huang, Kuo, Chen, & Chu, 2005; Lin, Lin, Huang, 2011). In order to describe the system in detail, this section will be organized into two sub-sections to depict the architecture of the system and the procedures of content development.

Architecture

Figure 1 shows the architecture of the wiki-based teaching material development environment that consists of four components, which are described below.

- *Learning object repository*. The contents of learning object repository are organized based on information about the learning objects. This consists of several pieces of information, such as the title, description, keywords, difficulty, lecture time, and so on. Each learning object can be defined or associated with different topics according to these.
- *Teaching material generation module*. The module is used to organize a tailor-made draft of teaching materials for each instructor based on specific requirements by measuring the fitness and velocity functions.
- *Wiki-based revision site*. The site was developed to allow instructors to revise drafts collaboratively. Inappropriate teaching materials would thus be revised until they are reliable.
- *Instructor interface*. The wiki-based teaching material development environment provides user-friendly interfaces for instructors who can administer the entire process through them.

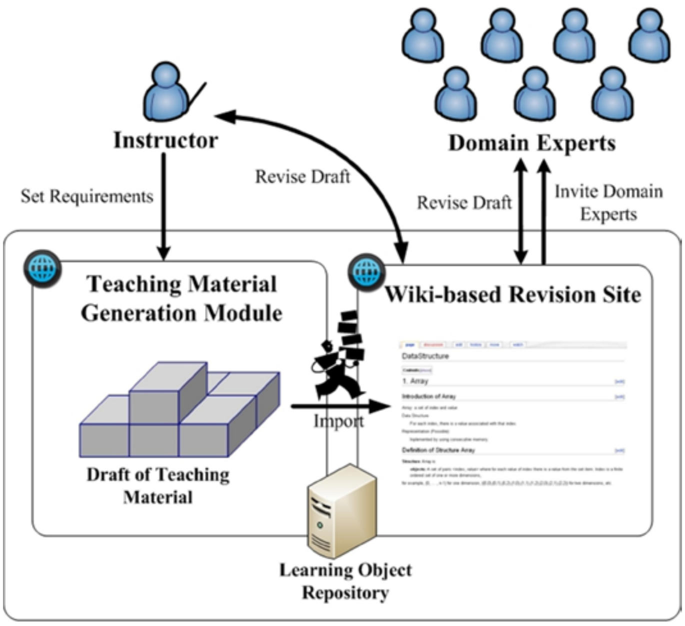

Figure 1. Architecture of Wiki-based teaching material development environment.

Procedure

Figure 2 schematically depicts the flow path of the complete system. The proposed approach is composed of three main phases, which will be described in detail in the following paragraphs.

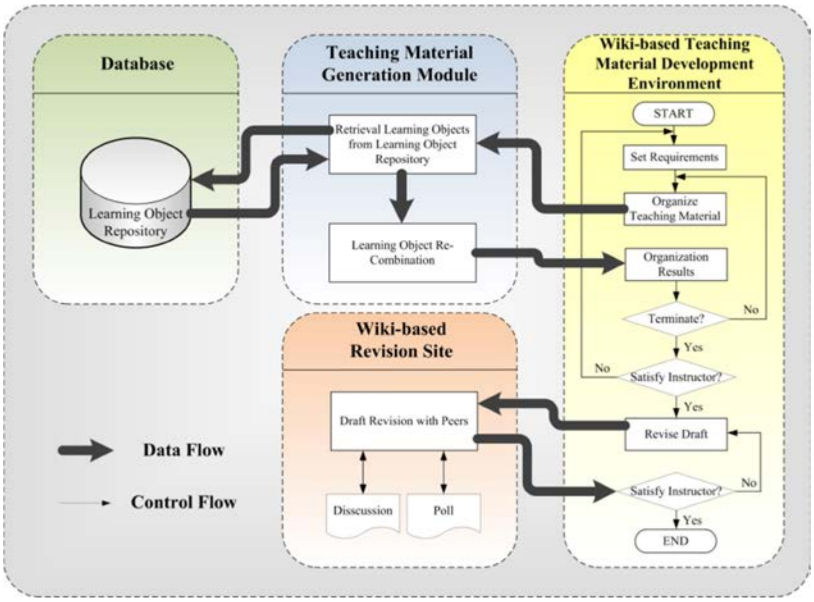

Figure 2. Logical system flow of wiki-based teaching material development environment

Phase 1. Requirement verification

This phase requires instructors to specify the relevant requirements for a teaching material, which include k topics $T_1, T_2, \ldots, T_x, \ldots, T_k$, the target difficulty level D, the lower bound lecture time, l, and the upper-bound lecture time, u, as shown in Figure 3.

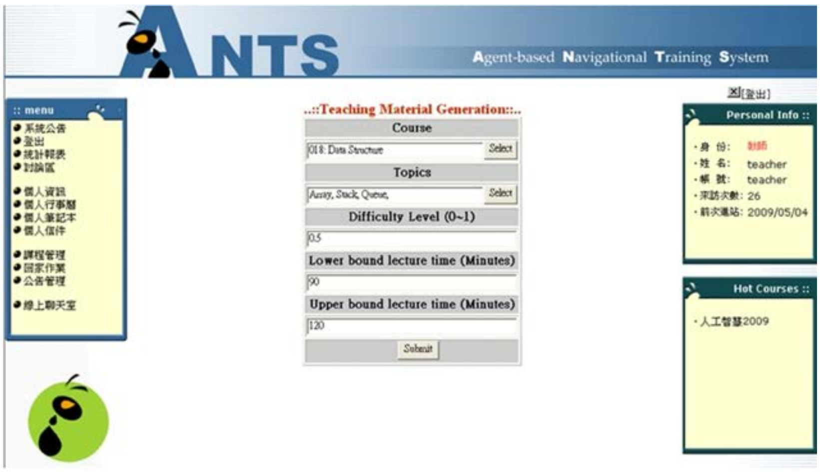

Figure 3. Screenshot of the parameter-setting interface

Phase 2. Learning object re-combination

To retrieve and re-combine relevant learning objects from LOR, an initial swarm is generated by the teaching material generation module. Because the module can obtain the expected lecture time for each learning object from the LOR, the number of selected learning objects in any particle can be bounded in an integrity rule, $[l/\max_{i=1\sim n}\{e_i\},\ u/\min_{i=1\sim n}\{e_i\}]$, during the generation of the initial swarm. Moreover, to obtain a quality initial swarm, a selection rule is developed that can give higher selection probability to the learning objects that have difficulty levels closer to the target. Formally, the selection rule is defined as $(S-|d_i-D|)/S$, where d_i is the degree of difficulty of learning object O_i and S is a constant.

After initiating the particle swarm, the module applies Equation 5 to measure the quality of each particle and then conducts particle iterations. In order to make sure that the best particle in each iteration survives, the elitist concept of the genetic algorithm (GA) has been incorporated into EPSO (Lin, Huang, & Cheng, 2010). If the best particle of the present iteration is worse than that of the previous iteration, the latter would replace the worst particle of the present iteration. By using Equation 10, each particle can update its velocity and position. Until the iteration terminates, the draft is displayed in a web-based interface and the instructor can check the results based on her or his own expertise, as shown in Figure 4. If the instructor is unsatisfied with the results, then she or he can require the module to produce another draft of the teaching material or revise the draft in the wiki-based revision site, as shown in Figure 5.

Figure 4. Screenshot of the draft generation interface

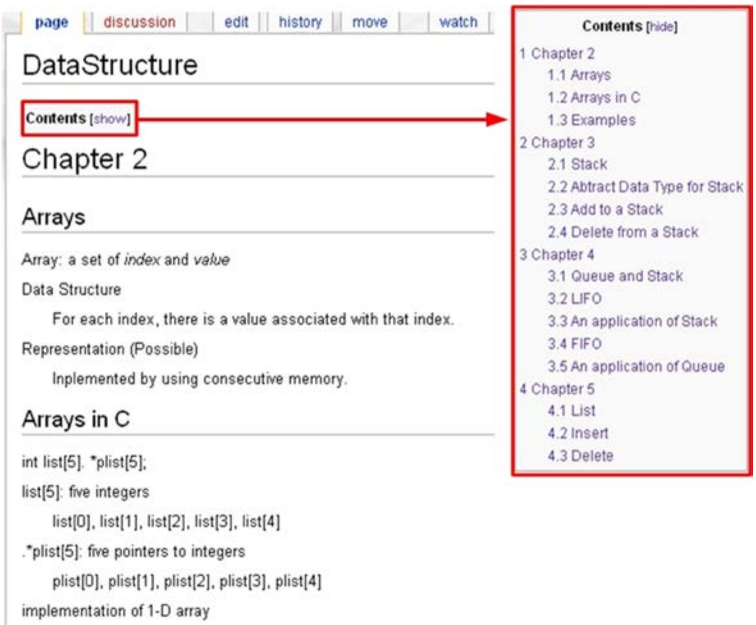

Figure 5. The draft of the teaching material

Phase 3. Wiki-based revision

Through the wiki-based revision site, the instructor can collaboratively improve the draft with peers or domain experts. Finally, the revised draft can be a formal version for use by instructors and learners, as shown in Figure 6.

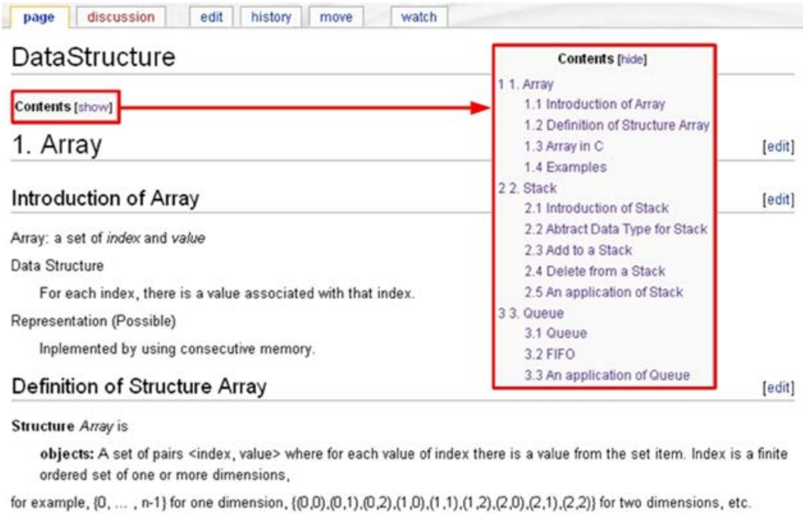

Figure 6. The final version of the teaching material

Experiments

The performance of the proposed approach is analyzed according to a series of experiments. First, we demonstrate that EPSO can adequately deal with the teaching material generation problem. Second, we analyze the robustness of EPSO against the variance between repeated runs and different problem scenarios. Third, we evaluate whether the wiki-based revision site can really help teachers to revise draft teaching materials. Finally, we investigate teachers' perceptions with regard to using the system.

Experiment settings

To analyze the comparative performance with other competing algorithms, nine simulation datasets were generated by varying the parameters. Table 1 shows the features of each dataset.

Table 1. Description of the experimental datasets

Dataset	Number of learning objects	Average Difficulty (ranging from 0 to 1)	Average expected lecture time for each learning object (seconds)
1	15	4.933	1078.333
2	20	4.050	953.500
3	50	5.100	1079.300
4	100	4.310	1095.970
5	300	4.963	1061.410
6	500	4.758	1090.926
7	1000	4.971	1072.026
8	1500	5.047	1123.973
9	2000	5.012	1101.681

Before conducting the experiments, we repeatedly ran EPSO with various values for the number of particles (P) and the maximal number of generations (G), as shown in Table 2, in order to obtain the best performance of EPSO. The results are tabulated in Table 3. Considering the computational time and optimal fitness value, the results indicate that the best performance of EPSO is when administering up to 10 generations with 20 particles, where the computational time (6.671 seconds) required is only longer than that needed for 10 generations with 10 particles, and the fitness value (0.045) obtained is the second best among all trials. Therefore, in the following experiments, EPSO is set with 20 particles and 10 generations.

Table 2. Combination of various values of P and G

P	*G*
10	10, 50, 100, 200, 300, 400, 500
20	10, 50, 100, 200, 300, 400, 500
30	10, 50, 100, 200, 300, 400, 500

Table 3. Computational times and the fitness values derived from EPSO with various values of P and G

	$P = 10$		$P = 20$		$P = 30$	
G	*t* (sec)	*f*	*t* (sec)	*f*	*t* (sec)	*f*
10	4.922	0.139	6.671	0.045	9.687	0.108
50	21.969	0.092	33.234	0.089	43.890	0.105
100	38.250	0.059	65.796	0.043	82.906	0.068
200	73.781	0.085	131.734	0.084	156.625	0.073
300	107.172	0.093	182.437	0.077	234.690	0.082
400	139.094	0.086	249.233	0.081	301.509.	0.052
500	198.641	0.103	321.057	0.094	392.052	0.065

Evaluation of EPSO performance

In this experiment, we evaluated EPSO by comparing its performance with those of three competing algorithms: non-enhanced particle swarm optimization (NEPSO), random method (RM), and exhaustive method (EM), using the nine simulation datasets tabulated above. The characteristics of the four competing algorithms are explained below.

- EPSO: The characteristics of EPSO are described in detail in earlier sections. In particular, all of EPSO trials are conducted with 20 particles and 10 generations according to the preliminary analytical results.
- NEPSO: NEPSO generates teaching materials by selecting learning objects randomly to meet all of the requirements, and discards the elitist mechanism of the genetic algorithm during the process determining PBest and GBest. As with EPSO, to obtain the best performance, we repeatedly run NEPSO with various values for the number of particles and the maximal number of generations, as shown in Table 2. The results indicate that the best performance of NEPSO is when administering up to 10 generations with 20 particles, where the computational time (5.582 seconds) required is only longer than that needed for 10 generations with 10 particles, and the fitness value (0.103) obtained is the second best among all trials. Therefore, NEPSO also uses 20 particles and 10 generations in all runs.
- EM: The characteristic of EM is that it guarantees to find out the optimal fitness value in each run because it exhaustively explores all possible solutions to the teaching material generation problem.
- RM: RM merely generates candidate solutions to the teaching material generation problem, rather than compute all possible solutions. Therefore, the optimality of the final solution is not guaranteed.

Since EPSO, NEPSO, and RM are stochastic-based methods, the performances of the three approaches were assessed according to the average of 10 runs on the nine datasets. In addition, the exhaustive method was run once so that it could enumerate all possible solutions. In order to conduct the performance experiment, we used the four programs to organize teaching material from the nine simulation datasets. The teaching material aimed at three topics, the target degree of difficulty was set at 0.6, and the expected lecture time ranged from 90 to 120 minutes.

Since EM is guaranteed to obtain the true optimal fitness value, we can evaluate the quality of solutions derived from EPSO, NEPSO, and RM by examining the differences between the four methods. As shown in Table 4, the fitness values derived from EPSO are very close to those produced by EM for the four smallest problems. However, the results also show that EM can only tackle the four smallest problems within a reasonable time. For the other large-scale cases, the computational time needed by EM would grow exponentially with the problem size. Although the computational time required by EPSO also increases with problem size, the rate of increase is relatively low.

Table 4. Comparison of the performances of EPSO, NEPSO, RM, and EM (with five particles and different numbers of iterations)

n	EPSO		NEPSO		RM		EM	
	t (sec)	*f*	*t* (sec)	*f*	*t* (sec)	*f*	*t* (sec)	*f*
15	0.702	0.043	0.609	0.113	0.067	0.586	1.343	0.000
20	1.017	0.086	0.913	0.127	0.083	0.755	3.281	0.000
50	4.531	0.092	3.953	0.117	0.217	0.814	85.828	0.000
100	7.684	0.062	6.503	0.120	0.397	0.827	700.547	0.000
300	19.864	0.046	17.103	0.133	1.167	0.789	N/A	
500	33.880	0.058	29.265	0.152	1.935	0.830	N/A	
1000	68.316	0.085	61.540	0.127	3.873	0.795	N/A	
1500	108.041	0.068	96.027	0.168	5.854	0.823	N/A	
2000	143.231	0.064	128.068	0.150	8.634	0.816	N/A	

We then compared the performance of EPSO with that of RM. With regard to the computational time, the amount needed by RM is rarely affected by the factors used and always remains acceptable in practice. In contrast, the computational time required by EPSO is affected with the number of learning objects. Nevertheless, even though EPSO needs a little more time than RM in all cases, the quality of the final solutions it finds is significantly better.
As for EPSO and NEPSO, the results of their comparison provide evidence as to the effects of the integrity rule, selection rule, and elitist mechanism on the solutions. As shown in Table 4, approximate solutions to all of the datasets can be obtained in a reasonable time, from 0.609 seconds to 143.231 seconds. However, the solution quality derived from EPSO is better than that with NEPSO, especially as the size of the datasets increases.

Figure 7 shows the variations with regard to the fitness values obtained by EPSO, NEPSO, and RM as the number of learning objects increases. The result shows that the fitness value increases as the problem size becomes larger for NEPSO and RM. Nevertheless, the rate of increase for the fitness value obtained by NEPSO is less than that for RM. In addition, the fitness values are of a relatively smaller magnitude for EPSO in all cases.

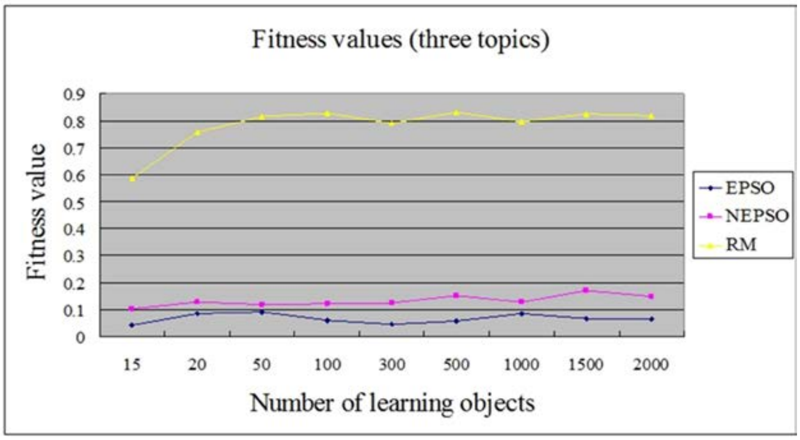

Figure 7. Variations in the optimal fitness value derived by EPSO, NEPSO, and RM as the number of learning objects increases

To summarize the performance experiment, EPSO can meet the requirements of most real-world applications for rapidly organizing teaching materials efficiently and effectively. Moreover, the initial rules (integrity and selection rules) and elitist mechanism are useful in aiding EPSO to deliver better solution quality.

Evaluation of EPSO robustness

The robustness is evaluated from two aspects. First, we evaluate the three methods with different numbers of learning objects. Second, we evaluate the standard deviation of the optimal fitness values obtained by the three

algorithms for different numbers of topics. In order to conduct the two evaluations, the three stochastic methods were run 10 times each and the standard deviation of the fitness value was computed over the 10 runs.
Figure 8 and Figure 9 show the variations of standard deviation of the optimal fitness values derived from EPSO, NEPSO, and RM, with different numbers of learning objects and topics. In the two evaluations, the standard deviations of EPSO are smaller than those of NEPSO and RM. The results demonstrate that EPSO is the most suitable and reliable method.

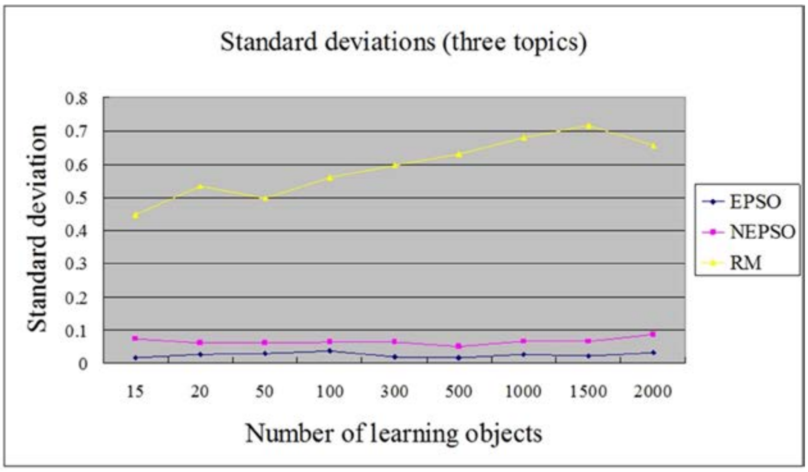

Figure 8. Variations in standard deviation of the optimal fitness values derived by EPSO, NEPSO, and RM

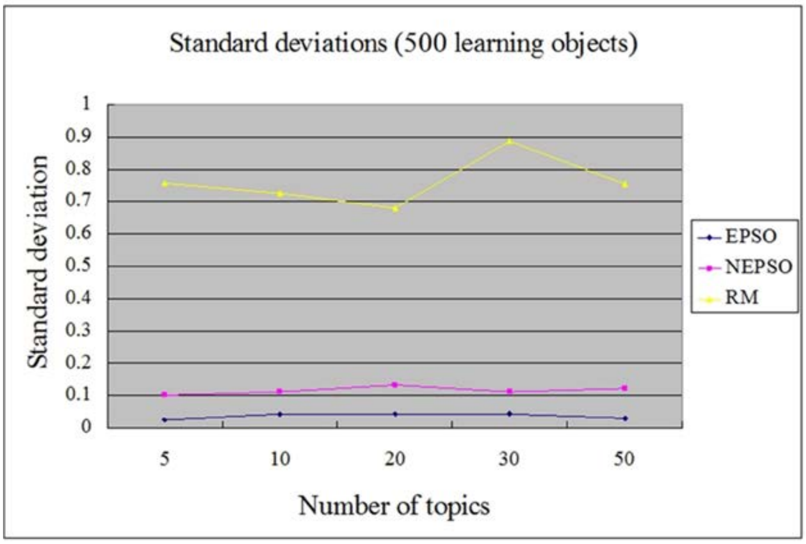

Figure 9. Variations in the standard deviation of the optimal fitness values derived by EPSO, NEPSO, and RM

Evaluation of the wiki-based revision approach

This experiment is to evaluate whether the revision time required for the teaching materials using the wiki-based revision site is shorter than without the proposed approach. Therefore, in this experiment a treatment group (using the wiki-based revision site) and a control group (not using the wiki-based revision site) were organized to investigate the effectiveness of the proposed function. The participants were 24 data structure course teachers, including 16 lecturers and eight professors. The average age of the teachers was 34. In the experiment, the 24 teachers were randomly divided into two groups of 12, each with eight lecturers and four professors. One group served as the experimental group, which had the wiki-based revision site to use throughout the revision process. The other served as the control group, working without the aid of the site, and thus they could only revise the teaching

materials manually by working alone. In contrast, teachers in the experimental group were able to form a wiki community to revise the materials with their peers or domain experts. All the participants were assigned the same teaching material formed by EPSO. The teaching material consisted of four learning objects that were selected from 20 learning objects. The parameters of each learning object were determined by a panel of experts. In addition, all the participants were provided with an Internet-enabled environment that meant they could search for information online while engaging in the revision process. At the end of the revision process, two data sources were used to evaluate the effectiveness of the proposed approach, including data logs and interviews with the experimental group.
First of all, we analyzed the revision time required by the two groups. As shown in Table 5, an independent *t*-test was used to examine whether the experimental treatment could really help the teachers to revise the materials more than the control group at a selected probability level (alpha 0.05 was selected in the analysis). The results reveal that there was a significant difference in the amount of time required between the two groups.

Table 5. Means, standard deviations, and independent t-test of the two groups in the evaluation study

Variable	Mean	Std. dev.	t-test(22)
Experimental group	46.799	9.739	2.022^{*}
Control group	55.677	11.678	

Note: $n = 24$ for all measures. $^{*}P < 0.05$

We next analyzed the behaviors of the teachers in the experimental group. A total of 42 comments were posted, and 25 comments were sent as replies to coordinate the process in the 10 teachers' wiki communities. To clearly present the data logs, this study synthesized the comments into three main topics with regard to opinion expression, opinion decision, and information sharing, as illustrated in Table 6.

Table 6. Example comments for the three topics

Inductive topics	Sample comments
Opinion expression	I thought that before teaching the queue unit, we should teach them the stack concept.
	I disagree with this arrangement, because the stack concept has been taught in the previous chapter.
	I would have preferred more interactions with students in this course.
	I felt the concept is difficult to grasp for students, therefore, we should add more examples to explain the concept.
Information sharing	I found a resource from this hyperlink. I thought that it could help you to complete this work.
	If you need more examples with regard to this concept, you can refer to this book.
	The students in the experiment group often gave feedback and asked questions.
Opinion decision	Do you agree to teach stack unit before teaching queue unit?
	Do you prefer which instruction strategy to teach this unit?

With regard to opinion expression, most of comments posted and replied to express personal opinions about revising the teaching materials. Working in this way, the teachers could not only get feedback from their peers, but also be stimulated to consider and help solve the problems that others were experiencing. By sharing what they know, the group of teachers using the wiki were able to access more information than would have had been possible had they been working alone. This assistance enabled them to produce the teaching materials more efficiently and effectively. However, despite the sharing of opinions and knowledge, sometimes a consensus could not be reached. In such circumstances, the teachers can use the poll function of the wiki to focus more clearly on an issue. In addition, by using a poll, some otherwise silent participants can be stimulated to offer their opinions. As mentioned above, we observed that the wiki-based revision site did indeed act as an effective medium to help the teachers revise the teaching materials collaboratively.

Finally, we interviewed the teachers in the experimental group to capture their perceptions of using the wiki-based revision site in more detail. As noted earlier, we found that the comments that were collected came from only 10 teachers' wiki communities. Therefore, in the interviews we first surveyed the two teachers who did not post any comments during the revision process. The two teachers indicated that they could revise the teaching materials by themselves. Hence, they used the wiki-based revision site alone. We also interviewed the other 10 teachers in the experimental group who felt that they could revise the teaching materials more efficiently because they could discuss issues with their peers or experts by posting messages on discussion pages, as shown in Figure 10. Furthermore, we asked the three teachers about their attitudes with regard to the polls that they used, and they all indicated this function helped to speed up decisions that needed to be made on a debatable revision, as shown in Figure 11. In addition, four teachers suggested that the site should integrate a tool to help novice teachers find collaborators and domain experts, as they felt that this would not be easy for them to do, since they tend to lack the necessary professional connections.

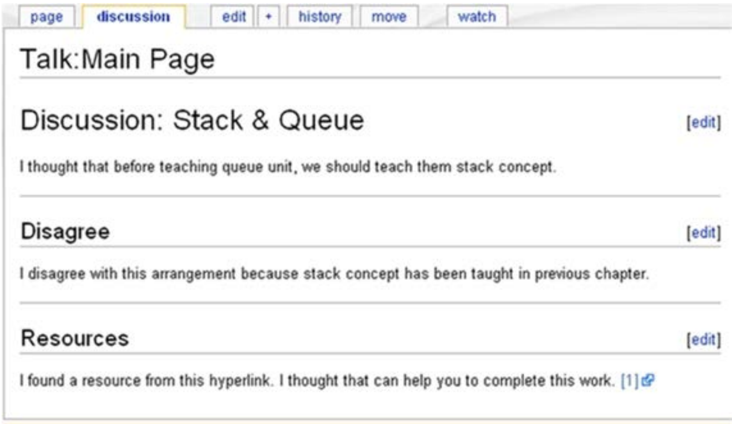

Figure 10. Screenshot of discussion page

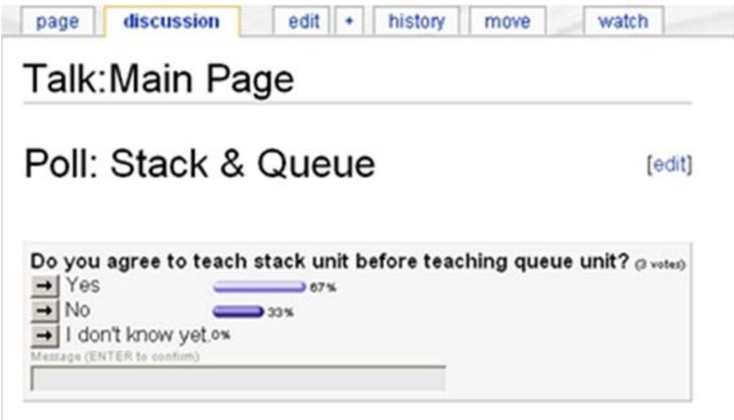

Figure 11. Screenshot of poll function

Perceived usefulness of the wiki-based teaching material development environment

In this experiment, the 24 teachers in the above experiment were asked to use the system. The perceived usefulness scale of the technology acceptance model (TAM) consists of five questionnaire items with a seven-point Likert scale (Davis, Bagozzi, & Warshaw, 1989), and it was applied to measure the users' perceptions with regard to the usefulness of the environment. The Cronbach's alpha value of the questionnaire items was .8040.
The results shown in Table 7, show that 78.3% of the users were in favor of using the wiki-based teaching material development environment.

Table 7. Users' perceptions of using the wiki-based teaching material development environment

#	Question	EU (%)	QU (%)	SU (%)	Neither (%)	SL (%)	QL (%)	EL (%)	Mean
1	Using the environment in teaching material development would enable me to organize appropriate learning objects more effectively	0%	8.3%	4.1%	4.1%	41.6%	20.8%	20.8%	5.25
2	Using the environment would improve my performance in developing teaching materials	0%	0%	4.1%	25%	37.5%	25.0%	16.6%	5.08
3	Using the environment in teaching material development would increase my productivity	0%	0%	4.1%	16.6%	54.1%	12.5%	12.5%	5.13
4	Using the environment would make it easier to carry out teaching material development	0%	4.1%	37.5%	29.1%	12.5%	12.5%	4.1%	4.96
5	I would find the environment useful in teaching material development	0%	0%	0%	16.6%	45.8%	41.6%	0%	5.21

Note. EU: Extremely unlikely; QU: Quite unlikely; SU: Slightly unlikely; SL: Slightly likely; QL: Quite likely; EL: Extremely likely

Next, the 24 teachers were interviewed to capture their perceptions of using the proposed approach in more detail. According to the interview results, the majority of teachers indicated that they felt the user interface of the proposed environment was clear and straightforward. Moreover, they felt that the teaching material development process could be carried out more easily and efficiently with the proposed approach, and that they had much more time to prepare the incoming instructions. Additionally, during the development process, the teachers could first think over the teaching materials. This added period of consideration helped them to prepare a better final product. However, one fourth of the teachers felt that if the teaching material generation module could provide extra information to explain the development results, then this could further reduce the development time.

With regard to the use of EPSO, a statistical result revealed that each teacher averagely use EPSO 2.333 times to obtain suitable results. Two teachers indicated that they could purposely and curiously run EPSO several times to see what the next result produced by EPSO. By excluding the phenomenon, the average times of using EPSO can be reduced to 1.916 times per teacher. Therefore, each teacher can spend only several seconds to obtain the suitable results. In addition, most of the teachers also stated that although the teaching materials developed by EPSO needed some manual modifications or even re-organization, the results were acceptable because the system had already saved a lot of time, and the computational time of EPOS is short. Moreover, 17 teachers indicated that the proposed environment allowed them to manually rearrange the teaching materials, thus enabling the materials to better meet their requirements. Finally, five teachers hoped that future versions of EPSO could consider the learning sequence of each learning object in order to further enhance the quality of the output.

As mentioned above, the results of this investigation show that EPSO can assist teachers in organizing teaching materials and further reduce the development time. Nevertheless, the wiki-based revision site will not always save time, because it may also need additional revision time, as well as discussions with their community members. However, this additional time can enable the teachers to produce higher quality teaching materials.

Conclusions

This paper describes a wiki-based teaching material development environment. By conducting a series of experiments, we show that the proposed approach can help instructors to develop and revise teaching materials in a collaborative manner. Although the users felt that the proposed environment could help them to form teaching materials, there are still some limitations to the proposed approach. First, more information should be provided to explain the development results, since EPSO cannot help users to automatically refine the developed teaching materials, in terms of the course sequence, content selection, and so on. Second, multimedia learning objects were ignored by the proposed approach, as they cannot easily be embedded and edited on the currently available wiki platforms.

Therefore, the future direction of this study is to continuously refine the proposed approach to support course sequence function. To address this problem, another element of LOM, namely semantic density, can be adopted to be a selection criterion of learning objects. Also, more participants with different background knowledge and teaching experiences will be invited to evaluate the proposed approach. We expect that the proposed approach can assist novice as well as experienced teachers to develop useful teaching materials rapidly and easily. Additionally, according to the zone of proximal development, the degree of difficulty of learning objects can be used as an attribute to develop an intelligent tutoring system based on the knowledge level of learners.

Acknowledgements

This work was supported in part by the National Science Council (NSC), Taiwan, ROC, under Grants NSC 100-2218-E-006-017, 100-2631-S-006-002, 100-2631-S-011-003, 100-2511-S-006-014-MY3, 100-2511-S-006-015-MY3, 101-2511-S-041-001-MY3, and 101-3113-P-006-023.

References

ADDIE (2004). ADDIE Design. Retrieved October 28, 2009, from http://ed.isu.edu/addie/design/design.html

Davis, F., Bagozzi, R., & Warshaw, R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science, 35*(8), 982–1003.

Ebersbach, A., Glaser, M., & Heigl, R. (2006). *Wiki: Web collaboration.* Berlin, Germany: Springer-Verlag.

García, F. J., & García, J. (2005). Educational hypermedia resources facilitator. *Computers & Education, 44*(3), 301–325.

Hofmann, J. (2004). *The synchronous trainer's survival guide: Facilitating successful live and online courses, meetings, and events.* San Francisco, CA: Pfeiffer.

Hwang, G. J. (2003). A conceptual map model for developing intelligent tutoring systems. *Computers & Education, 40*(3), 217–235.

IEEE Learning Technology Standards Committee (2002). *Draft standard for learning object metadata.* New York, NY: Institute of Electrical and Electronics Engineers, Inc.

Jeng, Y. L., Huang, Y. M., Kuo, Y. H., Chen, J. N., & Chu, W. C. (2005). ANTS: Agent-based navigational training system. *Lecture Notes in Computer Science, 3583*, 320–325.

Jong, B. S., Lin, T. W., Wu, Y. L., & Chan, T. Y. (2004). Diagnostic and remedial learning strategy based on conceptual graphs. *Journal of Computer Assisted Learning, 20*(5), 377–386.

Jou, M., & Liu, C. C. (2012). Application of semantic approaches and interactive virtual technology to improve teaching effectiveness. *Interactive Learning Environments, 20*(5), 441–449.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks* (pp. 1942–1948). Perth, Australia: IEEE Service Center. doi: 10.1109/ICNN.1995.488968

Kennedy, J. & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4104–4108). Piscataway, NJ: IEEE Service Center.

Kuo, Y. H., & Huang, Y. M. (2009). MEAT: An authoring tool for generating adaptable learning resources. *Journal of Educational Technology & Society, 12*(2), 51–68.

Leuf, B., & Cunningham, W. (2001). *The wiki way: Quick collaboration on the Web.* Upper Saddle River, NJ: Addison Wesley.

Lin, Y. T., Huang, Y. M., & Cheng, S. C. (2010). An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. *Computers & Education, 55*(4), 1483–1493.

Lin, Y. C., Lin, Y. T., & Huang, Y. M. (2011). Development of a diagnostic System using a testing-based approach for strengthening student prior knowledge. *Computers & Education, 57*(2), 1557–1570.

Lo, H. C. (2009). Utilizing computer-mediated communication tools for problem-based learning. *Journal of Educational Technology & Society, 12*(1), 205–213.

Meire, M., Ochoa, X., & Duval, E. (2007). SamgI: Automatic metadata generation v2.0. In C. Montgomerie & J. Seale (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* (pp. 1195–1204). Chesapeake, VA: AACE.

Motolet, O., & Baloian, N. (2007). Hybrid systems for generating learning object metadata. *Journal of Computers, 2*(3), 34–42.

Shih, W. C., Tseng, S. S., & Yang, C. T. (2008). Wiki-based rapid prototyping for teaching-material design in e-learning grids. *Computers & Education, 51*(3), 1037–1057.

Wang, T. H., Yen, N. Y., Du, Y. L, & Shih, T. K. (2007). A courseware authoring tool for achieving interoperability among various e-learning specifications based on Web 2.0 technologies. *Proceedings of the International Conference on Parallel Processing Workshops* (pp. 25–25). Washington, DC: IEEE Computer Society. doi: 10.1109/ICPPW.2007.32

Wheeler, S., Yeomans, P., & Wheeler, D. (2008). The good, the bad and the wiki: Evaluating student generated content as a collaborative learning tool. *British Journal of Educational Technology, 39*(6), 987–995.

Wikipedia (2004). *Wikipedia*. Retrieved October 28, 2009, from http://www.wikipedia.org/