*Journal of Universal Computer Science, vol. 16, no. 14 (2010), 1853-1881*
*submitted: 29/12/09, accepted: 25/07/10, appeared: 28/7/10 © J.UCS*

# A Treasure Hunt Model for Inquiry-Based Learning in the Development of a Web-based Learning Support System

**Dong Won Kim**
(University of Regina, Regina, Saskatchewan, Canada
kim263@uregina.ca)

**JingTao Yao**
(University of Regina, Regina, Saskatchewan, Canada
jtyao@cs.uregina.ca)

**Abstract:** One of the main problems of web-based learning is staying motivated at a sufficient level. Learning games offering challenges and entertainment may stimulate student motivation for learning and mitigate this problem. Web-based learning support systems combined with learning games may efficiently promote learning by encouraging student participation in learning. This study introduces a treasure hunt model, which represents the idea of inquiry-based learning using set theory. We demonstrate this via a prototype of a web-based learning support system called OTHI, which employs an online treasure hunt game as the learning game. We integrate the sound learning strategies of inquiry-based learning with the Web and online game technologies in this system. We expect that our learning support system will motivate students, and furnish an interactive student-centered learning environment.

**Key Words:** inquiry-based learning, web-based learning support systems, game-based learning, treasure hunt

**Category:** H.1 H.5 J.4 K.3.1

## 1 Introduction

Web-based learning is growing in popularity and complements traditional teaching methods. However, web-based learning does not always work effectively and sometimes fails to meet its educational goals [Xu and Wang 2006]. Students might passively take part in online classes which lack interaction with learning material [Khalifa and Lam 2002]. Moreover, in many online learning systems, instructors are not required to simultaneously participate in their students' learning process, and thus students cannot synchronize with instructors [Zhang and Nunamaker 2003]. Students normally study a subject in isolation. This passive, asynchronous web-based learning makes it difficult for students to become motivated towards learning [Jong et al. 2006].

Digital games may be applied to reduce the negative feedback of web-based learning when dealing with a lack of motivation for learning [Jong et al. 2006]. Digital games hold players' attention and stimulate their internal motivation [Hsiao et al. 2006]. The aim of a rational player is to win the game or to be one of the top players in the game. *Playing* implies active participation. That is, the game must be *player-centered*, and at every opportunity, players need to make their own decisions to win the game. Students require this type of motivation in their learning. Jong et al. [Jong et al. 2006] investigated student perception of a game-based learning method with traditional web-based learning. They found game-based learning was more capable of empowering students to learn with confidence and retain learned knowledge in their mind.

Learning support systems not only enhance student learning, but also motivate students to learn [Shareef et al. 2006]. The recent development of the Web has increased the need to move traditional computerized support systems to the Web platform and enhance support systems into further user-friendly systems [Yao and Yao 2003]. Web-based learning support systems (WLSS) are computerized learning support systems re-designed or modified to use Web technology to support student learning [Yao et al. 2007]. With the support of WLSS, students may not only eliminate study difficulties, but may also be encouraged to study learning concepts and topics.

Inquiry-based learning (IBL) is an educational approach driven more by student questions than by the instructor's lessons, and involves active, student-centered learning [Edelson et al. 1999, Fleissner et al. 2006]. Lim [Lim 2004] described web-based learning environments, designed with IBL, as providing students with cognitive tools and helping them form a learning community in which instructors and students interact to solve complex problems. IBL is compatible with constructivist learning strategies [Llewellyn 2004]. In constructivism, knowledge is defined as the cognitive structure of a person and learning is the active process of *constructing knowledge* rather than the process of knowledge acquisition [Kim 2005]. In IBL, students construct knowledge using an inquiry approach [Woolf et al. 2002].

In this paper, we propose a treasure hunt model, which embodies the idea of inquiry-based learning using set theory. The treasure hunt model is applied to the design of a web-based learning support system called OTHI - Online Treasure Hunt for Inquiry-based learning [Kim 2010]. OTHI employs an online treasure hunt game in which students conduct a *treasure hunt* on a certain topic and develop an answer to a given question, and includes a Web site to support instructors and students who utilize the treasure hunt game for student learning. With this system, students are able to experience IBL and study the course topics in an enjoyable and interesting way.

## 2 Web-based Learning Support Systems and Inquiry-Based Learning

### 2.1 Web-based Learning Support Systems

WLSS supports instructors and students to achieve better teaching and learning outcomes [Fan and Yao 2003]. *Web technology* refers to all technologies which implement, maintain, and use the Web. Advances in the Web technology remove any barriers of time and place for supporting various human activities and generate further momentum for the design and implementation of computerized support systems [Yao and Yao 2003].

Web-based support systems [Yao and Lingras 2003] take advantage of Web technology to provide further user-friendly support environments [Yao and Yao 2003]. The most popular and successful computerized support systems are decision support systems. If people develop decision support systems using Web technology, those systems become web-based decision support systems [Power and Kaparthi 2002]. Likewise, if people take into account learning, computerized learning support systems combined with Web technology can be intuitively called web-based learning support systems (WLSS) [Yao et al. 2007]. Any web-based system which contributes to students' learning can be viewed as a WLSS.

Fan and Yao [Fan and Yao 2003] described the main features of WLSS for the implementation of student-centered learning environments. The main features include: encouraging students to communicate with each other; delivering adapted learning content based on students' background knowledge; providing an interactive interface; and evaluating students' learning process. They also presented three views in terms of the design of WLSS, namely teacher, student and administrator views. Teachers consider WLSS to be convenient tools to create and modify learning content, while students regard WLSS as learning support tools containing a variety of learning contents. Administrators might use WLSS to maintain the systems.

### 2.2 Inquiry-Based Learning

Inquiry-based learning (IBL) is a student-centered instructional approach in which students seek truth, information, or knowledge by asking questions [Chan 2007, Fleissner et al. 2006]. IBL enables students to conduct self-directed investigations of problems and issues presented to them [Lim 2004]. Russell [Russell 1962] described inquiry as an investigation of a certain problem. Fleissner et al. [Fleissner et al. 2006] defined inquiry "as a seeking for truth, information, or knowledge; that is, seeking information by questioning." Chan [Chan 2007] described students constructing their perspectives of natural and human-designed worlds through inquiry activities.

The educational philosophy of IBL is founded on the ideals and principles of constructivism [Llewellyn 2004]. In constructivism, knowledge is defined as the cognitive structure of a student [Kim 2005]. The cognitive structure is changed, reconstructed and reorganized by the student's experience [Piaget 1976]. The constructivist believes students are the center of education, and learning is the product of self-organization and reorganization of the cognitive structure of the students [Llewellyn 2004, Yager 1991]. In other words, learning is an active, constructive process rather than process of knowledge acquisition, and teaching is the support of students' constructive processing of understanding rather than the delivery of information to the students [Kim 2005].

In addition, learning outcomes do not depend on what an instructor presents, but rather upon what information is encountered and how students process it, based on preconceived notions and existing personal knowledge [Yager 1991]. Inquiry, in constructivist teaching, offers an opportunity to better understand facts and formulas, and encourages student participation in learning [Lim 2004].

The process of IBL commonly described in the literature [Edelson 1998, Edelson et al. 1999, Lim 2004] involves the following four phases: 1) Presenting phase: presenting information and the background of the inquiry; 2) Retrieving phase: exploring and querying data and information for the inquiry; 3) Developing phase: developing and generalizing ideas or concepts by the evaluation and interpretation of data and information collected; and 4) Evaluating phase: evaluating the developed ideas and inquiry process, and giving constructive feedback.

Throughout the entire process of IBL, students may revise their ideas or decide to go forward with the intended direction by looking back at the question, process and direction of the inquiry. Students also communicate with others to fully understand given information, and clarify or solidify their findings.

### 2.3 A Web-enabled IBL Model: WebQuest

WebQuest (http://www.webquest.org) is a WLSS for IBL and is designed according to the guidelines proposed by Dodge [Dodge 1995]. Well-designed WebQuests promote learning practices by integrating the idea of IBL with Internet resources, open-ended questions and authentic tasks stimulating students' motivation [Chan 2007]. WebQuest is also a flexible model for Web-enabled IBL [Fleissner et al. 2006]. Instructors are able to very simply create their own WebQuests if they can create a document with hyperlinks.

A WebQuest should contain at least the following six components [Fleissner et al. 2006]: 1) Introduction: provides some background information allowing students to be ready to investigate a quest; 2) Task: specifies an interesting and doable task, duty, or assignment; 3) Resource: contains a collection of information sources - links and references - necessary to complete the task; 4) Process: describes the steps students should go through in accomplishing the task; 5) Evaluation: shows an assessment rubric informing how student performance will be evaluated; and 6) Conclusion: brings closure to the quest, reminds students of what they have learned, and encourages them to apply the experience to other domains.

An optional component, *credits*, denotes information about permissions to use and modify the WebQuest. Another optional component, *teacher page*, guides other teachers who want to implement the WebQuest.

## 3 Treasure Hunt Model for Inquiry-Based Learning

### 3.1 A Treasure Hunt for Inquiry-Based Learning

Treasure hunt was originally an outdoor activity and a game played by children and occasionally by adults. To play treasure hunt, an adult prepares a list of hidden objects for children to find. Each team of children receives a duplicate list of the hidden objects. The winner is the first team to find all the items on the list.

Di Blas et al. [Di Blas et al. 2004] introduced an online treasure hunt game in the SEE project providing students with a virtual learning environment. Students are included in online meetings and discuss previously studied themes under the active supervision of a guide, in a virtual museum. A treasure hunt in the SEE project helps students find a solution to cultural riddles provided by the museum, and enables them to review their learning in exciting ways. [Hamelin 2004] described an IBL experiment, in the form of a treasure hunt on the Web. It has been shown that a treasure hunt can be a very effective tool in developing searching abilities, using the Internet.

Chang et al. [Chang et al. 2006] implemented a system based on a treasure hunting learning model employing text-mode cell phones to communicate between students and the system. They applied the learning model to a traditional history and culture course in college. Once students send a position message to a positioning module of the system using their cell phones, its learning planner chooses a suitable quest and related guidance message on a basis of their learning records and physical position. Students do some kinds of treasure hunting, explore knowledge, and solve questions, based on the quest and guidance message.

A treasure hunt can be well matched to IBL. *Treasure* is considered to be information, truth, or knowledge, and *hunt* implies inquiry, which is a systematic investigation.

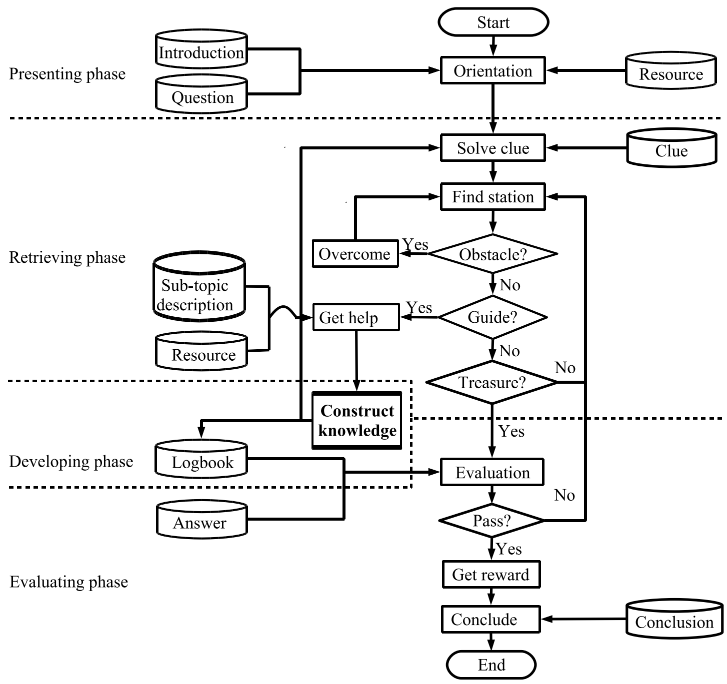


**Figure 1:** Flowchart of a treasure hunt process.

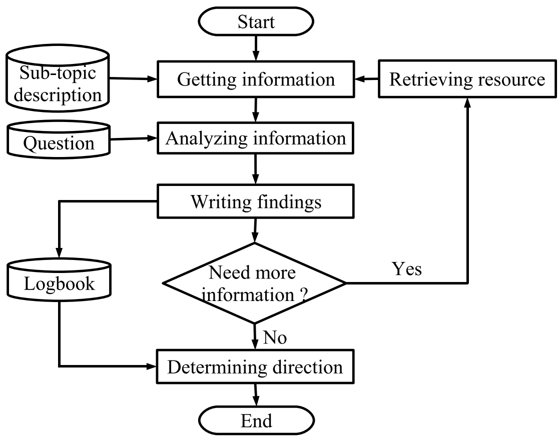


**Figure 2:** Flowchart of a knowledge construction.

*Treasure hunt* is an inquiry activity in which one systematically seeks knowledge with questions.

[Fig. 1] shows the process of the treasure hunt which we exploit in order to design a WLSS for IBL. The process has the following four phases:

1. **Presenting Phase**: In an orientation, a guide provides students with a question regarding a topic which they will investigate via a treasure hunt, as well as the outlines of the topic. The background information is available from a *resource* and helps students easily understand the objectives of the topic and the question.

2. **Retrieving Phase**: Once students receive their orientation from the presenting phase, they are required to explore every necessary *station* using the given *clues* and meet guides who offer a related sub-topic description called *help*. Students need to understand the sub-topic, using its *resource*, so as to construct a correct answer to the question. While exploring stations, students may meet and overcome several *obstacles*, and find *treasures*.

3. **Developing Phase**: When students obtain *help* at a *station*, they may develop ideas for the answer to the question through the "Construct knowledge" process, shown in [Fig. 1]. A more detailed flowchart is depicted in [Fig. 2]. Students analyze their collected information and write an idea into their *logbook*, with regard to the question. If they require more information to understand the given information, they may search a *resource*. By reviewing the ideas in the logbook, the students determine where to find more *helps* with the question, and thereby construct a correct answer to the question.

4. **Evaluating Phase**: Students are required to take a test when they find a treasure

at a *station*. In the evaluation, students must answer the question they received in the presenting phase, referring to ideas in their *logbook*. The answer is compared with the sample answer (prepared by the instructor) to determine whether they pass the test. If the students pass, they receive points and a *reward* which is useful in overcoming *obstacles* they will meet in future rounds, and they are also able to know available topics for the next round of treasure hunt. The winner is the student or team with the greatest number of points.

Each phase has its corresponding phase in the IBL process described in [Section 2.2]. The IBL process is also observed in this process. Therefore, it can be said that the treasure hunt does include the idea of IBL.

### 3.2 Structure of a Treasure Hunt Model

Before designing a WLSS, featuring treasure hunt, it is necessary to clearly define a treasure hunt by creating a *treasure hunt model* [Kim and Yao 2010]. In so doing, we come to understand what components and functions are needed, and how the functions might work for the system conducting the *treasure hunt* for IBL.

The treasure hunt model is a new Web-enabled IBL Model, formalizing the treasure hunt, using set theory, in order to efficiently integrate the learning strategies of IBL into WLSS. The treasure hunt model consists of seven components: *agents*, *treasures*, *information*, *events*, *actions*, *stations*, and *logbooks*. Its structure is defined as follows:

$$S = (AGT, TRE, INF, EVT, ACT, STA, LOG). \tag{1}$$

– $AGT$ is a finite nonempty set of *agents* representing students or guides in the system, and expressed as $AGT = SD \cup GI$, where $SD$ is the set of all students and $GI$ is the set of all guides. Guides help students find treasures by providing useful information about the treasures.

– $TRE$ is a finite nonempty set of *treasures*. A *treasure* is the conclusion of a topic and a reward for learning the topic. The *reward* is useful in helping students overcome obstacles to find other treasures.

– $INF$ is a finite nonempty set of *information* about the treasures. This is defined as $INF = \{TP, RS, QS, CL\}$, where $TP$ is a finite nonempty set of learning topics, $RS$ is a finite nonempty set of resources, $QS$ is a finite nonempty set of questions, and $CL$ is a family of nonempty sets of clues. A *learning topic* is a finite nonempty set of learning *objects*[1] necessary to master a topic. The *resource* is a collection of information sources related to a learning object. The *question* is what students must investigate while learning a topic. The *clue* is a rhymed hint about the next learning object to be found and learned.

[1] *Objects* to be dealt with in the course of this paper might be significant words, concepts, topics, educational materials, information units, documents, functions, commands, etc [Jones and Furnas 1987].

- $EVT$ is a finite nonempty set of *events* which invite student participation. There are four different types of events: orientation events, help events, obstacle events and evaluation events.
- $ACT$ is a finite nonempty set of *actions* students can perform in order to find the *treasure*. Some major actions are as follows: finding the information, analyzing the information, retrieving the resource, documenting the findings and determining the direction.
- $STA$ is a finite nonempty set of *stations* where the event occurs and students initiate the actions. In accordance with the events, the stations can be grouped into the following types: orientation stations, help stations, obstacle stations and evaluation stations.
- $LOG$ is a finite nonempty set of *logbooks* containing all records of the students' important achievements during the treasure hunt. Students record their findings in the logbooks, the contents of which assist them in constructing new knowledge.

Let $n$ be the number of topics for a course and let $m$ be the number of sub-topics of each topic. $TP$ of $INF$ in [Eq. 1] can be expressed as $TP = \{L_1, L_2, \cdots, L_n\}$, where $L_i$ (for $1 \leq i \leq n$) is the learning topic. Learning topic $L$ is described as $L = \{k_0, k_1, k_2, \cdots, k_m, k_{m+1}\}$, where $k_0$ is the *introduction* of the topic, $k_i$ (for $1 \leq i \leq m$) is a *sub-topic* of the topic, and $k_{m+1}$ is the *conclusion* of the topic. In the treasure hunt model, a set $T$ of all learning *objects* which students are required to understand in order to complete a *course* through the treasure hunt, is defined as:

$$T = \bigcup_{L \in TP} L. \tag{2}$$

Let $IT \subset T$ be a set of all the *introductions* of topics for a course, let $SU \subset T$ be a set of all the *sub-topics*, and let $CO \subset T$ be a set of all the *conclusions* of topics. Thus, the set $T$ of all the objects of the course is also defined as:

$$T = IT \cup SU \cup CO. \tag{3}$$

The learning objects are allocated to stations and guides using a learning object allocation function $f_{OA}$ given by:

$$f_{OA} : T \rightarrow STA \times GI. \tag{4}$$

If $TN$ is a finite nonempty set of names of topics for a course, an orientation function $f_{OT}$ representing the orientation event of $EVT$ in [Eq. 1] is described as:

$$f_{OT} : STA \times GI \times TN \rightarrow IT \times QS \times CL. \tag{5}$$

The orientation function $f_{OT}$ allows a guide to provide an introduction, a question and a clue for a given topic name at a station. The introduction is called an *orientation object*. A help function $f_{HL}$ for the help event of $EVT$ is expressed as:

$$f_{HL} : STA \times GI \times QS \rightarrow SU \times CL. \tag{6}$$

The definition of the help function $f_{HL}$ describes a guide furnishing a sub-topic description and a clue for a given question at a station. The sub-topic description is called a *help object*.

Let $f_{SM}(q,t,t^{'})$ be a similarity function to measure the similarity of a student answer $t^{'}$ to a sample answer $t$, provided by the instructor, for a given question $q \in QS$. If $S_q$ is a set of significant terms appearing in the question $q$, $S_t$ is a set of significant terms appearing in the sample answer $t$, and $S_{t'}$ is a set of significant terms appearing in the student answer $t^{'}$, then the similarity can be calculated by:

$$f_{SM}(q,t,t^{'}) = \frac{|S_{t'} \cap (S_q \cup S_t)|}{|S_q \cup S_t|}. \tag{7}$$

It means the similarity is measured by the ratio between the number of significant terms, included in both the student answer and the sample answer or question, and the number of significant terms, included in the sample answer or question.

If $RW$ is a set of rewards the student receives after passing a test at an evaluation event, the set of treasures $TRE$ in [Eq. 1] can be described by $TRE = CO \times RW$, where $CO$ is a set of all the topic *conclusions* as previously described. Therefore, an evaluation function $f_{EV}$ for the evaluation event of $EVT$ is defined as:

$$f_{EV} : STA \times QS \times R(f_{SM}) \rightarrow TRE, \tag{8}$$

where $R(f_{SM})$ is the range of $f_{SM}$. The evaluation function $f_{EV}$ expresses the fact a student is able to obtain a reward and a conclusion of the topic as a treasure, at a particular station, based on the similarity of their answer to a sample answer for a question. The conclusion is called an *evaluation object*.

The obstacle event gives students enjoyment, and hinders their ability to find the treasures. If $OB$ is the finite nonempty set of obstacles, an obstacle function $f_{OB}$ for the obstacle event of $EVT$ is described as:

$$f_{OB} : STA \times OB \times 2^{RW} \rightarrow \mathbb{N}, \tag{9}$$

which represents how well students overcome an obstacle at a station using rewards, which they obtained. Students acquire some points while struggling with the obstacle at a station, using the rewards they received.

### 3.3 Learning Order Relationship and Knowledge Spaces

A web-based learning support system needs to know each student's current learning state, and arrange the next learning objects for the student to learn, based on the learning state. The treasure hunt model should define the learning states and explain how to derive the next objects. To suggest the next objects to the students, we also need a mechanism to express which objects the students have learned and which objects might be next. Thus, we present the learning order relationship [Yao et al. 2007] and its knowledge space [Schrepp 1997].

*Learning order of objects* provides the correct order in which a student is supposed to learn the other objects which remain after learning one object [Yao et al. 2007].

**Definition 3.1** *We call a relationship between two objects a learning order relationship if the relationship satisfies the learning order of objects.*

In other words, the learning order relationship between two related objects can be clearly expressed as follows: *Learning one object is a prerequisite for learning another object*. It reflects the student's relative preference for the objects, based on his/her background knowledge. [Fig. 3] shows an example of the learning order of concepts in a textbook. The student may prefer "Concept 2 of chapter 3" to "Concept 3 of chapter 3" as their next learning concept.

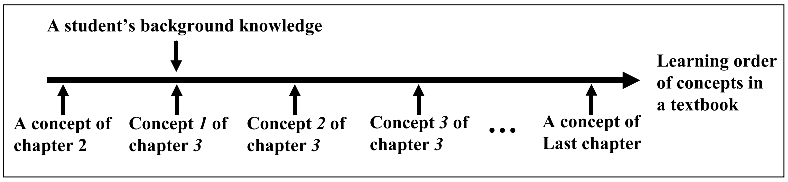


**Figure 3:** An Example of Learning Order of Concepts.

Let $T$ be a finite nonempty set of all objects in a collection of related documents. Then, a binary relation on $T$, with respect to Definition 3.1, is called a *learning order relation* $\preceq$. The learning order relation $\preceq$ on $T$ can be formally defined by: for $t_1, t_2 \in T$,

$$t_1 \preceq t_2 \Leftrightarrow \text{anyone who is able to learn } t_2 \text{ should also be able to learn } t_1. \tag{10}$$

$t_1$ is called a *prior* object of $t_2$, and $t_2$ is called a *posterior* object of $t_1$. The prior object $t_1$ of $t_2$ can be regarded as a prerequisite for mastering $t_2$. The learning order relation $\preceq$ on $T$ also shows the prerequisite relationship between two objects in the set $T$.

The subset $K$ of objects from $T$ is called the knowledge state of the student. A knowledge structure on $T$, containing all possible knowledge states from $T$, can be restricted by the *learning order relationships* between objects in $T$. For a learning order relation $\preceq$ on $T$, a knowledge structure, associated with the relation, is defined by:

$$\mathscr{K}_R = \{K | (\forall t, t' \in T, t' \preceq t, t \in K) \Rightarrow t' \in K\}. \tag{11}$$

It means a knowledge state $K \in \mathscr{K}_R$ contains all the prerequisites of an object $t$ if $t$ is an element of $K$, and the prerequisites are linearly ordered along with one single dimension, taking the learning order relation as deterministic. Hence, the object $t$ can only have one set of prerequisites. The knowledge structure $\mathscr{K}_R$ contains the empty set $\emptyset$ and the entire set $T$, and is closed under set union and intersection.

As previously described, in the knowledge structure $\mathscr{K}_R$, an object can have only one set of prerequisites, which is able to provide one *learning path* to the student. However, in practice, it is possible for an object to have more than one set of prerequisites. A mapping $\sigma : T \to 2^{2^T}$ is called a *surmise function* [Schrepp 1997] if it satisfies the following conditions: 1) $P \in \sigma(t) \Rightarrow t \in P$; 2) $(P \in \sigma(t), t' \in P) \Rightarrow (\exists P' \in \sigma(t'), P' \subseteq P)$; and 3) $P \in \sigma(t) \Rightarrow (\forall P' \in \sigma(t), P' \not\subseteq P)$. The interpretation of a surmise function $\sigma(t) = \{P_1, \cdots, P_n\}$ is that every student who is able to learn an object $t$ is also able to learn all objects from at least one element $P_i$ of $\sigma(t)$. The elements in $\sigma(t)$ are presented as sets of prerequisites for an object $t$.

The pair $(T, \sigma)$ represents a *surmise system* [Xu et al. 2008]. For the surmise system, a knowledge structure $\mathscr{K}_S$ on $T$, in which an object $t$ can have sets of its prerequisites $\sigma(t)$, is described as:

$$\mathscr{K}_S = \{K | (\forall t \in T, t \in K) \Rightarrow (\exists P \in \sigma(t), P \subseteq K)\}. \tag{12}$$

The knowledge structure $\mathscr{K}_S$ associated with the surmise system $(T, \sigma)$ is called a *knowledge space* on $T$, which is closed under union. The knowledge space can represent a collection of all possible *learning paths*, for all objects in $T$, with respect to the learning order relationship.

Let $T_K^O$ be the outer fringe [Albert and Hockemeyer 1997] of $K \in \mathscr{K}_S$. $T_K^O \subset T$ is a set of all objects $t$ such that adding $t$ to $K$ forms another knowledge state $K' \in \mathscr{K}_S$. Let $T_K^I \subset T$ be the inner fringe [Albert and Hockemeyer 1997] of $K$. $T_K^I$ is a set of all objects $t$ such that removing $t$ from $K$ forms another knowledge state $K'' \in \mathscr{K}_S$. Thus, objects in the outer fringe $T_K^O$ are the *next objects* to be suggested to the student whose current knowledge state is $K$.

### 3.4 Learning State of the Treasure Hunt Model

The concept of the knowledge space defined in [Eq. 12] allows us to discuss the learning state, which shows a student's learning progress. Furthermore, we are able to present how the current knowledge state of a student is changed by student's learning experience using the knowledge space.

If $LS$ is a learning state structure, $LS$ can be defined as a tuple:

$$LS = (SD, t_0, \lambda, LE, f_{LE}), \tag{13}$$

where $SD$ is a finite set of students, $t_0$ is a course introduction, $\lambda$ is the sequence of stations, $LE$ is the set of learning experiences a student has in order to complete a course, and $f_{LE}$ is an experience interpretation function.

Let $\mathscr{K}_S$ be a knowledge space on $T$, defined in [Eq. 2], and $\lambda$ be the sequence of stations a student visits in the treasure hunt. *Learning*, in the treasure hunt model, takes place at a station $s_\lambda \in STA$ as a student takes a series of actions $A_\lambda$ based on the student's current knowledge state $K_\lambda \in \mathscr{K}_S$, for a given object $t_\lambda \in T^O_{K_\lambda}$, the outer fringe of $K_\lambda$. The series of actions $A_\lambda$ is expressed as $A_\lambda = (a_1, \cdots, a_l)$, where $l$ is the number of actions which the student takes at the station, and for $1 \leq i \leq l$, $a_i \in ACT$ from [Eq. 1].

The set $LE$ of learning experiences can be expressed as:

$$LE = \bigcup_\lambda (s_\lambda \times K_\lambda \times t_\lambda \times A_\lambda). \tag{14}$$

If there is a function $f(\lambda) : \mathbb{N} \to LE$, the experience interpretation function $f_{LE}$ is defined as:

$$f_{LE} : t_0 \times \prod_\lambda f(\lambda) \to K_n, \tag{15}$$

where $t_0$ is the course introduction and $K_n \in \mathscr{K}_S$ is the new knowledge state of the student. $K_n$ will be used as the current knowledge state in the next station, replacing the old knowledge state of the student. As described in [Section 3.3], for the new knowledge state $K_n$, the next available objects for the student can be denoted by $T^O_{K_n}$, the outer fringe of $K_n$.

According to the learning state structure, the past learning states of a student are retrieved by changing the sequence $\lambda$. $LE$ in the structure may be a data storage, wherein the learning experiences of a student are kept. The system writes the learning experiences into the data storage. The past learning experiences can be the learning history of the student. The learning history is also retrieved from $LE$ by changing the sequence $\lambda$.

### 3.5 Knowledge Construction of the Treasure Hunt Model

It is important that the treasure hunt model defines the process of the *knowledge construction*, depicted in [Fig. 2]. [Fig. 4] shows a high-level view of the treasure hunt

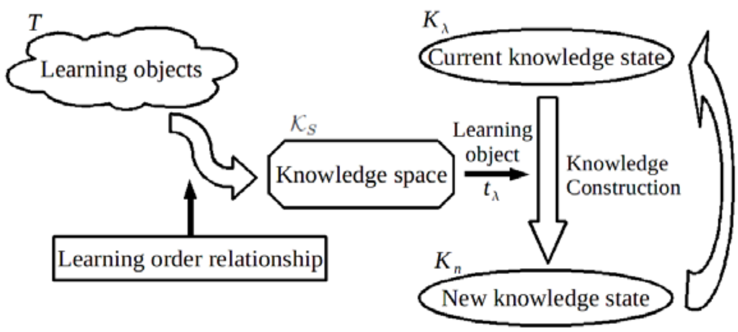


**Figure 4:** High-level view of the treasure hunt model

model, including the knowledge construction. Learning objects in $T$ make up a knowledge space $\mathscr{K}_S$ by applying learning order relationships between the objects. The student chooses a learning object $t_\lambda$ among the available objects derived from the knowledge space $\mathscr{K}_S$, based on his/her current knowledge state $K_\lambda$.

In fact, for [Eq. 14], all the essential *actions* for learning are involved in the knowledge construction process. The first information, given to the student at the station $s_\lambda \in STA$, is the object $t_\lambda$ the student chose before for this round of treasure hunt. The student is able to construct knowledge by writing an idea, from the analysis of given information, into the logbook, and attain a new knowledge state $K_n \in \mathscr{K}_S$. The student may retrieve the *resource* related to the object $t_\lambda$ to obtain more information and clearly understand it. The current knowledge state $K_\lambda$ is replaced with the new knowledge state $K_n$ for the next round.

In the system, the *blog* works as the logbook. If $IS$ is an idea structure describing the idea posted to the *blog*, $IS$ can be defined as a tuple:

$$IS = (QS, \lambda, SBJ, CMT, REL), \tag{16}$$

where $QS$ is a finite nonempty set of questions from $INF$ of [Eq. 1], $\lambda$ is the sequence of stations, $SBJ$ is a set of subjects, $CMT$ is a set of comments, and $REL$ is a set of relationships. The *subject* is the title of the *comment*, and the *relationship* represents the relationship between the *question* and the *comment*. A *writing idea* function $f_{WI}$, posting an idea to the *blog*, is defined as:

$$f_{WI} : QS \times \mathscr{K}_S \times T \times RS \to IS, \tag{17}$$

where $T$ is the set of all the objects as defined in [Eq. 2], $\mathscr{K}_S$ is the knowledge space on $T$, and $RS$ is a finite set of resources from $INF$ of [Eq. 1]. The function $f_{WI}$ is a mapping function specifying an idea developed from the given learning object and related resource, based on the current knowledge state of the student in order to find a correct answer to a question.

The idea structure $IS$ presents fields which may compose a blog entry form. The function $f_{WI}$ also describes the information available from the system, allowing the student to fill out the fields on the entry form.

## 4 OTHI - Online Treasure Hunt for Inquiry-Based Learning

### 4.1 Overall System Architecture

OTHI is a web-based learning support system for IBL and is designed on the basis of the treasure hunt model. The architecture of OTHI is depicted in [Fig. 5]. It basically follows the thin-client and server structure, and the three-layer architecture [Yao and Yao 2003]. OTHI has the following major subsystems: (1) a treasure hunt game, (2) a teaching support subsystem, (3) a learning support subsystem, and (4) a

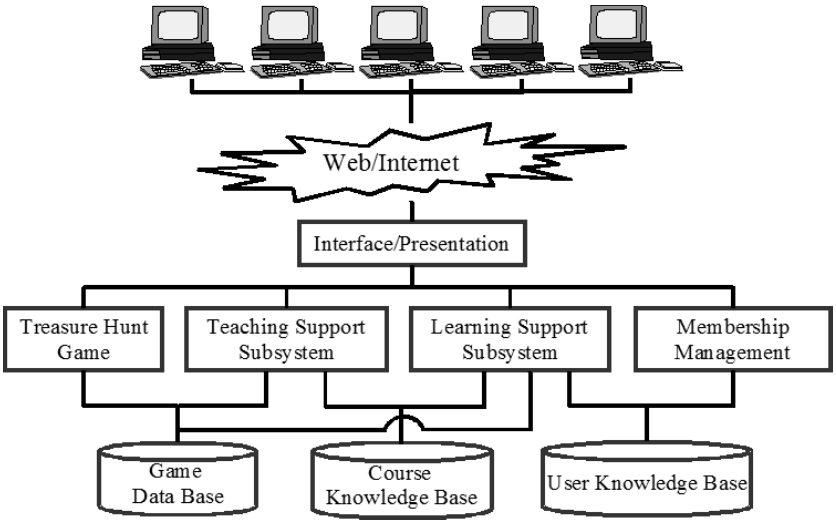


**Figure 5:** An Architecture of OTHI.

membership management subsystem. The teaching support subsystem implements the teacher's view of WLSS, while the learning support subsystem implements the student's view of WLSS.

The membership management subsystem manages various members such as the present, past and affiliate members, and supports three different types of user accounts: students, instructors and administrators. The members' profiles are managed on the user knowledge base. One of its most important features is the identification and authentication of each user. The user is able to access available services based on the authentication.

The teaching support subsystem enables instructors to design their courses as IBL. Instructors define course outlines, topics, questions, clues and related information using this subsystem. The system stores the information into the course knowledge base and generates IBL environments into the treasure hunt game. From this generation, treasure boxes containing questions, conclusions and rewards are placed at certain stations. The generation also allows non-player characters to have their own words to guide students as the *guides* of the treasure hunt model.

The learning support subsystem provides students with *resources* related to each topic as well as information about the treasure hunt. Students can search for more information about a topic using a Web search engine, if necessary. In addition, they are able to post their findings onto the *blog*, the data of which are stored in the user knowledge base. The system also automatically posts their achievements during the treasure hunt to the *blog*. The *blog* helps students develop their ideas and compose an answer to a question by tracing their findings.

Treasure hunt game provides a playground for the treasure hunt. All required information for the treasure hunt game is located in the game database. Students become motivated, and experience both the treasure hunt and IBL by playing the game. The learning support subsystem provides students with a virtual place, such as their home, where students complete homework and prepare for their next classes, while the treasure hunt game provides a different virtual place, such as a classroom where students learn from a teacher.

### 4.2 Treasure Hunting Algorithm

Treasure hunting algorithm (see Algorithm 1) describes a major function of the treasure hunt game. Its input data contain outer fringes of all knowledge states $\in \mathscr{K}_S$ and a set $STA$ of all stations in the system. Guides and treasure boxes, placed at the stations, already have the information and rewards to be delivered to students by the instructor using the teaching support subsystem.

Line **1** initializes variables. $n$ counts how many topics are mastered; $clue$ contains a clue; **t** is a learning object; **s** is a station; $points$ are experience points; $RW$ represents a reward inventory the student possesses; and $K$ is a knowledge state. Line **2** introduces the course to the student. Line **3** adds the course introduction object to the current knowledge state $K$, and gets an outer fringe $T_K^O$ of $K$. Lines **5-33** repeat until the student learns all topics of the course. Line **5** allows the student to choose one object $\mathbf{t} \in T_K^O$, which contains all the objects available to learn for the next round of treasure hunt. Line **6** invokes the function Explore_stations_and_overcome_obstacles() to find an appropriate station for the object **t**. The function allows the student to try to overcome obstacles and get some experience points while exploring stations.

Lines **8-13** are performed if the object **t** is an orientation object. Otherwise, if the object **t** is a help object, lines **16-22** are processed. If the object **t** is an evaluation object, lines **25-32** are executed. As defined in [Eq. 13], [Eq. 14] and [Eq. 15], line **8** calls the function Orientation() to give the student an orientation about a topic, line **16** invokes the function Help() to provide the sub-topic information, and line **26** invokes the function Evaluation() for the student to take a test if a treasure is found in a treasure box at the station **s**. Lines **10-12** are processed if the orientation was provided. Line **10** adds the object **t** to the current knowledge state $K$, and get an outer fringe $T_K^O$ of $K$. Line **11** displays a list of possible objects to learn and allows the student to choose an object **t** for the next round. Line **12** calls the function Get_clue_for_next_station() to have a clue, leading to the station where the next object **t** will be treated.

Lines **18-21** are processed if the sub-topic information was provided. Line **18** invokes the function Knowledge_construction() to construct an idea using the sub-topic information and get a new knowledge state $K$. Line **19** gets an outer fringe $T_K^O$ of $K$. Line **20** is identical to lines **11**, and line **21** is quite similar to line **12**. Lines **28-30** are

/* Refer to [Eq. 13], [Eq. 14], [Eq. 15] */

**Data**: Outer fringes of all knowledge states $\in \mathscr{K}_S$ and a set $STA$ of all stations

**Input**: A course introduction object $\mathbf{t_0}$

**Output**: A knowledge state $K$ containing all learning objects $\in T$

$n := 0$; $clue, \mathbf{t}, \mathbf{s} :=$ null; $points := 0$; $RW, K :=$ the empty set;
display $\mathbf{t_0}$.course_title, $\mathbf{t_0}$.course_objectives and $\mathbf{t_0}$.course_references;
add $\mathbf{t_0}$ to $K$; $T_K^O :=$ Outerfringe($K$);
**while** $n <$ *number of topics of the course* **do**
    **if** $\mathbf{t} = null$ **then** select an object $\mathbf{t} \in T_K^O$;
    $\mathbf{s} :=$ Explore_stations_and_overcome_obstacles($\mathbf{t}$, $clue$, $RW$, $points$);
    **if t** *is an orientation object* **then**
        $\mathbf{o} :=$ Orientation($\mathbf{s}$,$\mathbf{t}$);
        **if** $\mathbf{o} \neq null$ **then**
            add $\mathbf{t}$ to $K$; $T_K^O :=$ Outerfringe($K$);
            display a list of all objects $\in T_K^O$; select an object $\mathbf{t}$ in the list;
            $clue :=$ Get_clue_for_next_station($\mathbf{o}$.**guide**,$\mathbf{t}$);
        **end**
    **end**
    **else if t** *is a help object* **then**
        $\mathbf{h} :=$ Help($\mathbf{s}$,$\mathbf{t}$);
        **if** $\mathbf{h} \neq null$ **then**
            $K :=$ Knowledge_construction($K$,$\mathbf{o}$.question,$\mathbf{h}$,$\mathbf{t}$);
            $T_K^O :=$ Outerfringe($K$);
            display a list of all objects $\in T_K^O$; select an object $\mathbf{t}$ in the list;
            $clue :=$ Get_clue_for_next_station($\mathbf{h}$.**guide**,$\mathbf{t}$);
        **end**
    **end**
    **else if t** *is an evaluation object* **then**
        **if** *a treasure is found in a box* $\mathbf{b} \in \mathbf{s}.treasure\_boxes$ **then**
            $reward :=$ Evaluation($\mathbf{b}$);
            **if** $reward \neq null$ **then**
                add $reward$ to $RW$;
                add $\mathbf{t}$ to $K$; $T_K^O :=$ Outerfringe($K$);
                $\mathbf{t}, \mathbf{o}, clue :=$ null; $n := n + 1$;
            **end**
        **end**
    **end**
**end**

**Algorithm 1**: Treasure Hunting Algorithm performed if the reward is available. Line **28** adds the reward to the reward inventory $RW$. Line **29** is identical to line **10**. Line **30** initializes variables and increases $n$ by 1.

For a learning topic $L$ described in [Eq. 2], the last learning object is the evaluation object of $L$. It means a new topic will be introduced to the student at the next time, and the next object must be the orientation object of the new topic.

**Definition 4.1** *We say that every element of the outer fringe of an evaluation object should be an orientation object in the learning order of objects.*

Line **30** shows the orientation information **o** about the current topic is no longer necessary, since a new topic will be treated for the next round according to Definition 4.1.

### 4.3 A Demonstrative Example for a Data Communication and Network Course

According to the instructor's lecture notes on a topic, data link layer, the instructor may have the following learning objects: DLL introduction, DLL duties, error detection, flow control, parity checking, etc. In addition, the learning order relationships between the objects, are obtained using the learning order relation $\preceq$, given by [Eq. 10], on the set of the objects as follows:

$$\begin{aligned}&\text{DLL Introduction} \preceq \text{DLL Duties}, \text{DLL Duties} \preceq \text{Error Detection},\\&\text{DLL Duties} \preceq \text{Flow Control}, \text{Error Detection} \preceq \text{Parity Checking},\\&\text{Flow Control} \preceq \text{Parity Checking}.\end{aligned}$$

Since the object "DLL Duties" has two outer fringes – Error Detection and Flow Control – from the learning order relationships, the instructor can determine two possible learning paths: 1) DLL introduction $\rightarrow$ DLL Duties $\rightarrow$ Error Detection $\rightarrow$ Flow Control $\rightarrow$ Parity Checking; and 2) DLL introduction $\rightarrow$ DLL Duties $\rightarrow$ Flow Control $\rightarrow$ Error Detection $\rightarrow$ Parity Checking. The instructor may allow students to choose one of the two paths by defining the two objects as the outer fringes of "DLL Duties." Otherwise, the instructor fixes the learning path by choosing one. In this case, the first learning path is chosen.

Before defining a topic using Edit Topic pages of the teaching support subsystem (TSS), shown in [Fig. 6] and [Fig. 7], the instructor must select a course for a topic. Similarly, as represented in [Fig. 8], to define a sub-topic with an Add Sub-Topic page of TSS, the instructor is also required to select both a course and a topic, to which the sub-topic belongs. For the learning object "DLL introduction," the instructor enters the topic title with "Data Link Layer," and also its introduction and question on an Edit Topic Orientation page as depicted in [Fig. 6]. On the Edit Topic Evaluation page, the instructor enters a sample answer to the question presented on the Edit Topic Orientation page, a conclusion of the topic and a clue as to its evaluation station.

Other learning objects, except for "DLL introduction," become the sub-topics of the data link layer. [Fig. 8] shows the definition of a sub-topic "DLL Duties." The instructor enters its title and description, and also a clue as to its help station. Other sub-topic definitions can be performed in a similar manner.

Once the instructor finishes making up a topic with the Edit Topic Orientation, Edit Topic Evaluation and Add Sub-Topic pages, the instructor must build a learning envi-

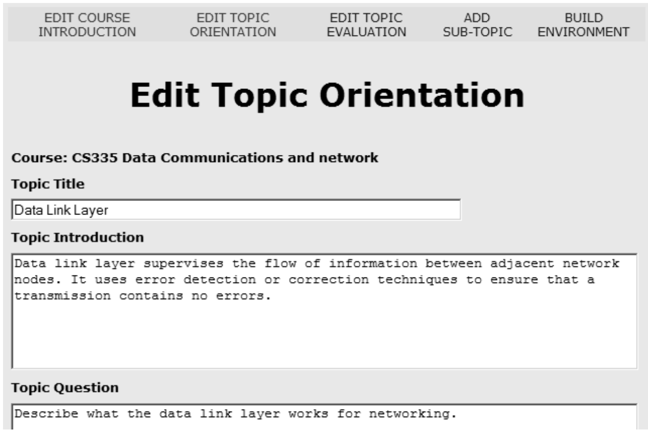


**Figure 6:** Edit Topic Orientation of teaching support subsystem.

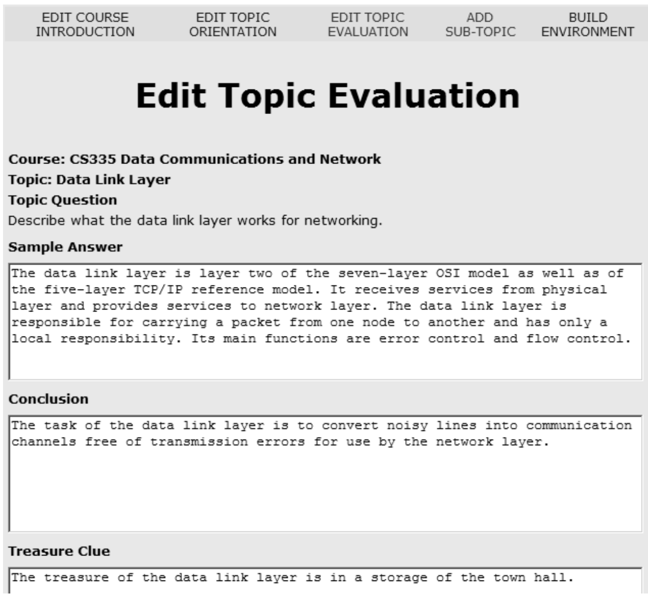


**Figure 7:** Edit Topic Evaluation of teaching support subsystem.

ronment into the treasure hunt game for the topic, using Building Learning Environment pages of TSS. As shown in [Fig. 9], for the topic "Data Link Layer," the instructor assigns a game map "Town Hall" as the orientation station and NPC "Mayor" as the guide, in accordance with the learning object allocation function, defined in [Eq. 4]. The in-

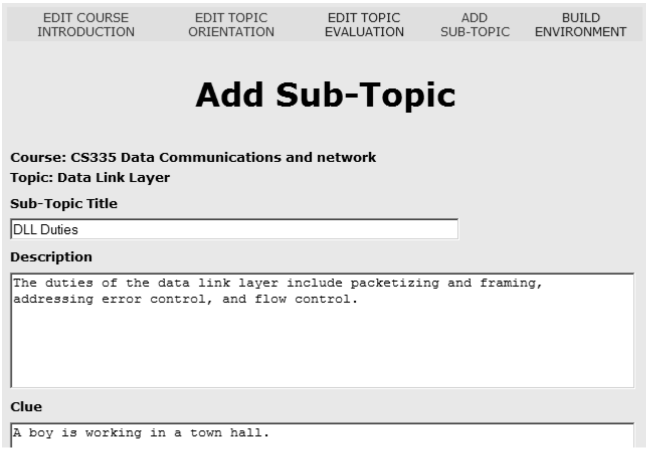


**Figure 8:** Add Sub-Topic for the sub-topic DLL Duties.

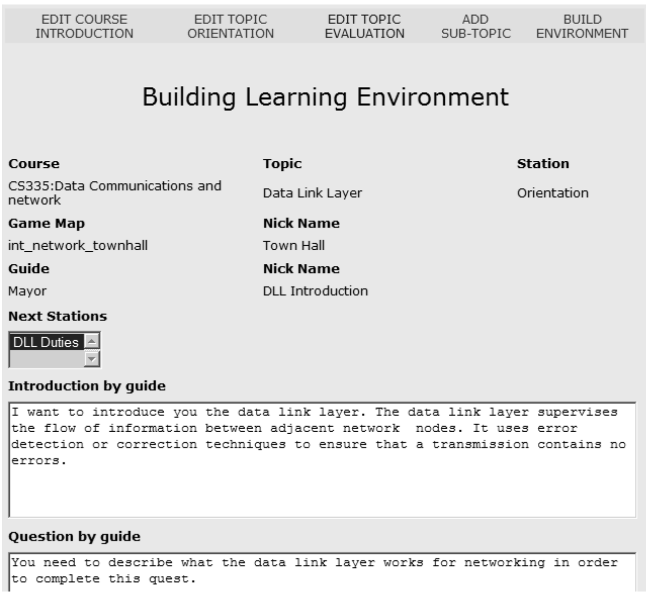


**Figure 9:** Building Learning Environment for orientation

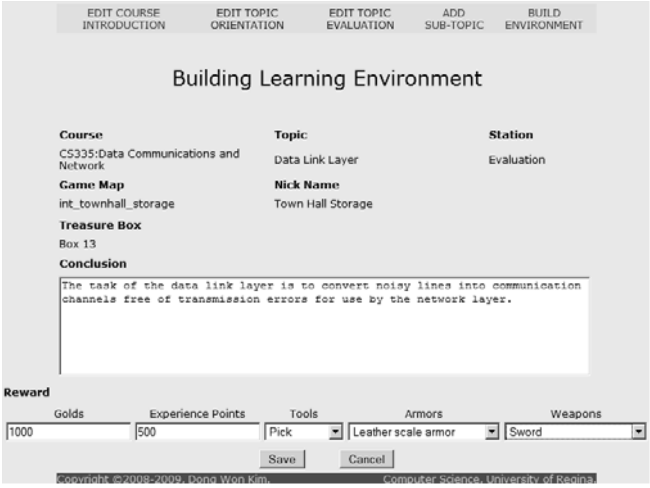


**Figure 10:** Building Learning Environment for evaluation.

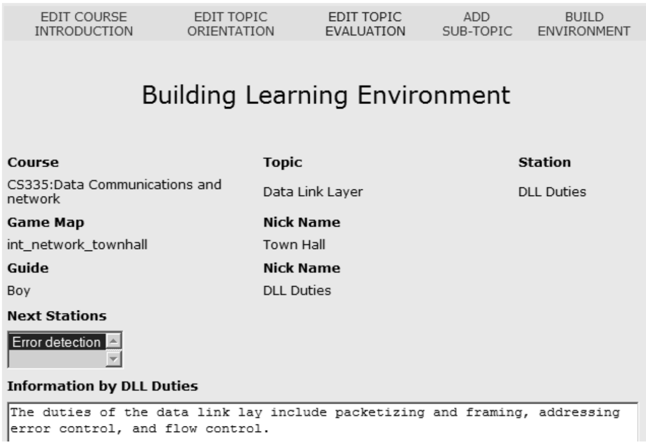


**Figure 11:** Building Learning Environment for sub-topic DLL Duties.

structor also modifies the introduction of the topic to simulate a conversation. For the evaluation station, the instructor chooses a treasure box, “Box 13,” in “Town Hall Storage,” and sets 1,000 gold, 500 game experience points and so on, as a reward for the quest, as depicted in [Fig. 10].

Meanwhile, according to the building definition of “DLL Duties,” shown in [Fig. 11], NPC “Boy” in “Town Hall” will deliver the description, shown in [Fig. 8], to the students. Sub-topic “Error Detection,” specified in the field “Next Stations,” is an outer

| Events | Stations | Game Objects | Learning Objects |
|---|---|---|---|
| Orientation | Town Hall | NPC Mayor | DLL Introduction |
| Help | Town Hall | NPC Boy | DLL Duties |
| Help | Network City | NPC Alice | Error Detection |
| Obstacle | Network Library Entrance | Rats | |
| Help | Network Library | NPC Monogenes | Flow Control |
| Obstacle | Town Hall Storage Entrance | Rats | |
| Evaluation | Town Hall Storage | Box 13 | Conclusion |

**Table 1:** Combination of game components and learning objects.

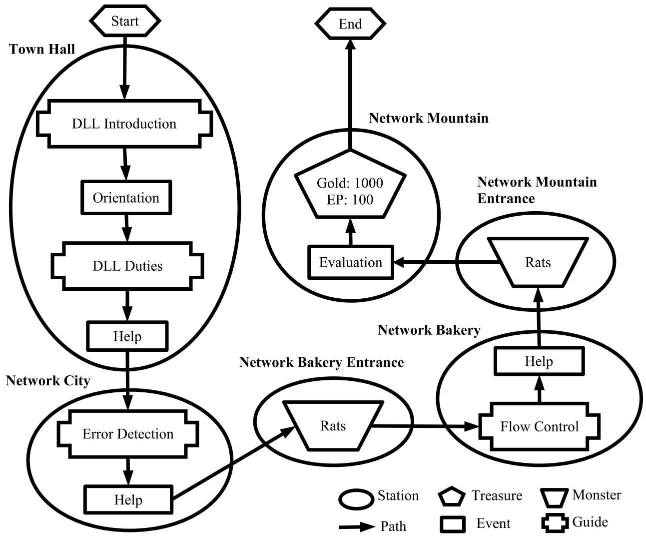


**Figure 12:** A Treasure Map.

fringe of the sub-topic. The combination of game objects and corresponding learning objects is described in [Tab. 1].

[Fig. 12] shows a journey to the treasure of the data link layer. In the game world, the name of the learning object each NPC will deal with is displayed as the name of the NPC. From the orientation function, defined in [Eq. 5], a student receives the quest about the data link layer from an NPC "DLL Introduction," including the background information and question. The student posts what (s)he has understood, with the orientation information, to the *blog* as the writing idea function, described in [Eq. 17]. To write an idea about it, the student may search for more information with key words "data link layer" or "introduction data link layer" using the *Google Search*.

After posting the idea, the learning object "DLL introduction," indicated in [Tab. 1], is added to the knowledge state of the student according to the experience interpretation function expressed in [Eq. 15]. For the next available sub-topics, there is only one sub-topic available, “DLL Duties,” in accordance with the building definition for the orientation shown in [Fig. 9]. Hence, NPC “DLL Introduction” tells the student the clue: “A boy is working in a town hall.” The student finds the boy called “DLL Duties” in a town hall.

When the student meets NPC “DLL Duties,” as described by [Eq. 6], (s)he may learn the basic duties of the data link layer from the NPC. To gain further understanding, the student searches for additional information, relevant to the sub-topic “DLL Duties,” using the search engine. Then, the idea regarding the sub-topic is posted to the *blog*, and the sub-topic becomes a member of the knowledge state of the student. For the next sub-topic “Error Detection”, NPC “DLL Duties” gives the student the clue: “A woman is waiting for you in a city.” The student moves to find the woman called “Error Detection” in a city.

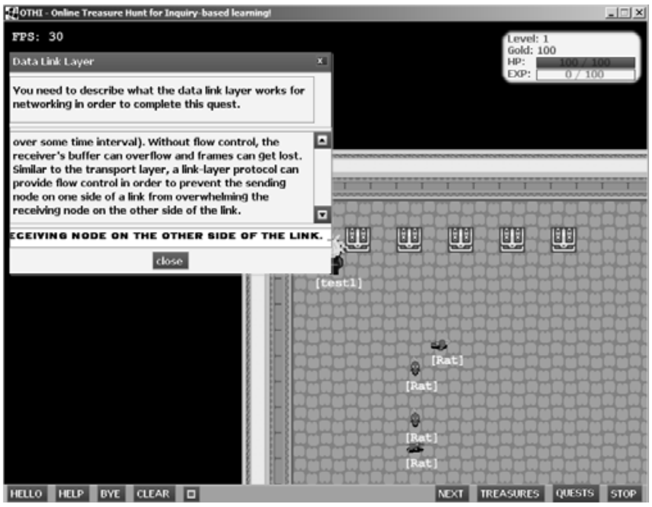


**Figure 13:** Ask for an answer in the treasure hunt game.

The woman can be found in “Network City.” The woman talks about the error detection of the data link layer and shows a clue for NPC “Flow Control” as the next object to learn. The student meets rats when (s)he tries to enter the “Network Library” on the way to find NPC “Flow Control.” The student is required to defeat the rats in order to check if the NPC is there. According to the obstacle function, defined in [Eq. 9], some experience points are obtained as the student expels the rats using his/her weapons.

After overcoming all the challenges, the student finds a treasure box in “Town Hall Storage.” When the student opens it, as shown in [Fig. 13], (s)he is asked to give an

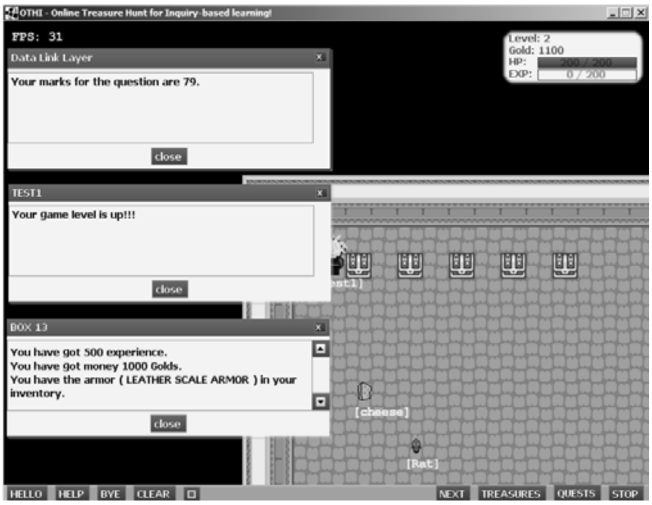


**Figure 14:** Complete a quest in the treasure hunt game.

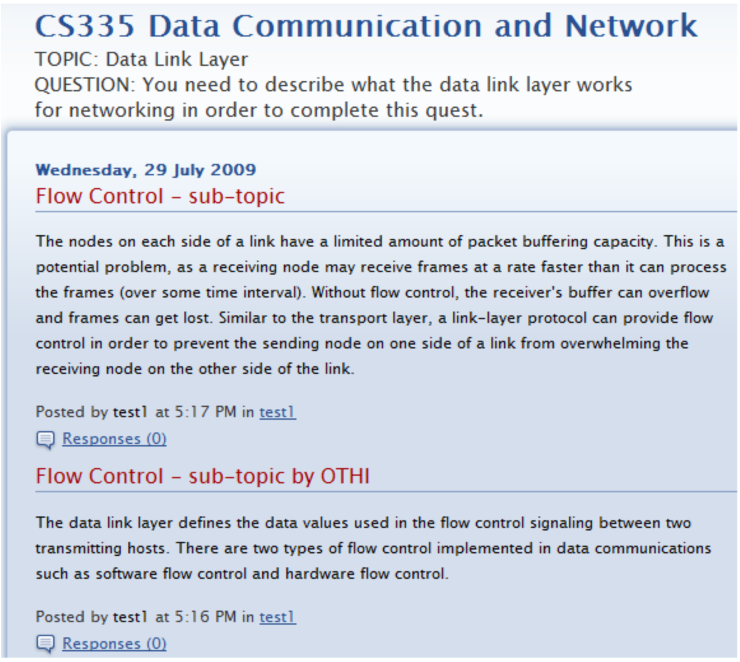


**Figure 15:** Blog posts about sub-topic Flow Control.

answer to the question to complete this quest in accordance with the evaluation function described in [Eq. 8. For the treasure, as defined in [Fig. 10], the student may gain 1,000 gold and 500 game experience points if (s)he passes. [Fig. 14] shows the student achieving 79 points for the question. Enough experience points enable the student to *level up*. The gold the student received will be spent on arms to help defeat the *obstacles* (s)he will face for the next round of the treasure hunt. The game level of the student is one of the measures indicating how well (s)he has performed learning.

As previously described, during the journey to the treasure hunt, the student is required to write his/her idea about each sub-topic in the *blog*. Each post to the *blog* forms the idea structure, defined in [Eq. 16]. [Fig. 15] shows the blog posts about the sub-topic "Flow Control," written by the system, as well as by the student. "Flow Control" is the subject and the "sub-topic" represents the relationship of the idea structure. The posts also contain comments below the subject portions. The question is shown on top of the *blog*. This mechanism can help the student organize ideas about the question and give the appropriate answer when the student finds the treasure and takes the test.

## 5 Usefulness of OTHI for Learning

The usefulness of fully implemented OTHI can be shown by means of contrastive evaluation of *OTHI supported learning* against other learning approaches: traditional face-to-face classroom learning (TFCL), and passive, asynchronous web-based learning (PAWL). The evaluation was performed by investigating the comparative advantages and disadvantages of each learning approach in previous studies related to web-based learning, inquiry-based learning, or learning games.

The following five attributes were developed to deal with the pros and cons of the learning approaches:1) Availability: Learning availability, including the accessibility and reuseability of learning services; 2) Flexibility: Flexibility and adaptability of learning content and process, according to students' individual needs; 3) Interactivity:

| **Evaluation attributes** | **Traditional learning (TFCL)** | **Web-based learning (PAWL)** | **TFCL with OTHI** | **PAWL with OTHI** | **Number of related research papers** |
|---|---|---|---|---|---|
| Availability | Low | High | Medium | High | 8 |
| Flexibility | High | Low | High | Medium | 9 |
| Interactivity | High | Low | High | Medium | 12 |
| Motivation | Medium | Low | High | High | 13 |
| Student-centricity | Low | Medium | High | High | 10 |
| Total | 10 | 8 | 14 | 13 | (Low=1, Medium=2, High=3) |

Table 2: Comparative and contrastive evaluation of OTHI and other learning approaches.

Real-time interactivity between students and instructors or learning content, and among students; 4) Motivation: Learning motivation and engagement, encouraging students to actively participate in learning; and 5) Student-centricity: Student-centricity in the process of teaching and learning.

[Tab. 2] presents the results found in an investigation of the evaluation. Each learning approach was evaluated and classified (low, medium, high) based on its comparative advantages and disadvantages as reported in the literature for the attributes and compared with the other learning approaches. The first column lists the evaluation attributes, and the values of the attributes for each learning approach are found in the second, third, fourth and fifth columns. The last column parades the number of related research papers from which the results are served for each attribute.

The availability of the traditional face-to-face classroom learning (TFCL) is relatively low when compared with other learning approaches, while the passive, asynchronous web-based learning (PAWL) shows a high availability of learning services. The OTHI supported learning approach may include either PAWL or TFCL. If OTHI supports students in PAWL, its availability is the same as the availability of PAWL. Otherwise, if OTHI supports students in TFCL, the overall availability of the learning services cannot attain the availability of the pure web-based learning due to the low availability of TFCL, in spite of the support of OTHI to improve its availability.

Previous studies reported TFCL provides students with higher flexibility and interactivity than PAWL. In a traditional classroom, if a student does not understand a particular section of a learning material, the student can raise questions and receive an immediate presentation about the material from the teacher who uses an example, a story, or just more detail. The teacher can adapt content and pace to the rate at which students understand the material. In PAWL, neither the instructor nor the delivery system can adapt the course presentation to different students, and online students perceive less interactivity compared to students in the traditional classroom.

On the other hand, the features of OTHI (such as the *learning state*, Web search engine and online treasure hunt game) are able to enhance the flexibility and interactivity of PAWL. A student who uses OTHI in learning is able to find and learn appropriate topics based on his/her *learning state*, and does not need to waste time learning irrelevant or already known material. In addition, the student cannot only find appropriate answers to questions about the topics on the Web using the search engine, but is also able to receive instant feedback regarding the questions from other students who have already investigated the questions in the treasure hunt game. The *guide*, an animated character [Sheth 2003], in the treasure hunt game provides a more interactive learning experience to the student. Nevertheless, PAWL with OTHI cannot attain the flexibility and interactivity of the traditional face-to-face classroom learning (TFCL), unless the animated character can compete with the human instructor.

As for the motivational aspect of the learning approaches, TFCL is more motivational than PAWL. Zhang et al. [Zhang et al. 2004] specified that one of the advantages of TFCL is “motivating students,” as compared to PAWL. Furthermore, previous studies on web-based learning reported PAWL makes it difficult for students to become motivated for learning.

OTHI supported learning provides students with higher motivation for learning than TFCL. Learning games can enhance students’ learning motivation when applied in classrooms or other learning scenarios. Particularly, online-learning games can arouse the motivation of the students by their enjoyment of playing the game and learning a topic at the same time. The online treasure hunt game of OTHI can be a valuable learning motivator and promote learning motivation. Furthermore, the inquiry activities that the game furnishes can create a motivation to learn. The *guide* works as an animated pedagogical character [Shaw et al. 1999, Conati and Zhao 2004] and also helps to increase the student’s motivation and engagement.

The traditional face-to-face classroom learning (TFCL) exhibits relatively lower student-centricity, representing a teacher-centered learning, when compared to other learning approaches. The high availability of the passive, asynchronous web-based learning (PAWL) increases the student-centricity of the learning approach, while the low availability of TFCL decreases its student-centricity, despite its high flexibility and interactivity, since the flexibility and interactivity are mostly controlled by the teacher, not by the student. Even though PAWL shows high availability compared to OTHI supported learning, its student-centricity is lower than OTHI supported learning due to its low flexibility, interactivity and motivation, on which student-centricity is also dependent. In addition, OTHI supported learning contains inquiry-based learning in which students are the center of learning and the role of the instructor is to assist the students in constructing their own knowledge.

The last row of [Tab. 2] indicates the total results of the evaluation, which may represent the learning effectiveness of the learning approaches. TFCL with OTHI shows the highest effectiveness, while PAWL shows the lowest effectiveness. In addition, the results reveal TFCL is still a better learning approach than PAWL, and the two OTHI supported learning approaches are more effective than the others.

However, the total results may be changed if each attribute is given some weight to reflect its relative importance under a certain learning situation. The evaluation results of [Tab. 2] can be gained only if every attribute has the same weight which is 1. Some courses may give a relatively high weight to the availability and low weight to other attributes so as to provide students with better learning environments for the courses. In real learning situations, the evaluation results may vary due to the consideration of additional evaluation attributes, as well as various attribute weights. For example, *collaborativity* may be added as one attribute to measure the effectiveness of the learning approaches for certain courses.

Although the results shown in [Tab. 2] do not provide any empirical evidence regarding the effectiveness of OTHI, at least it can be said that the results present the usefulness of OTHI in improving the learning effectiveness of other learning approaches.

## 6 Conclusion

Web-based learning support systems support the activities of instructors and students. Inquiry-based learning is a student-centered educational method in which students learn topics through a series of inquiry activities. Web-based learning support systems, combined with learning games, are able to support and encourage students to engage in learning. The effective integration of the learning games into web-based learning support systems can be achieved by applying the idea of inquiry-based learning to the systems.

In this research, we designed and developed a prototype of a web-based learning support system called OTHI, based on a treasure hunt model. The treasure hunt model represents the idea of inquiry-based learning using set theory. According to our model, inquiry activity is described as a treasure hunt. Students may learn about a topic by developing an answer to a question from the treasure hunt. Fully implemented OTHI is able to improve the learning effectiveness of other learning approaches by stimulating student motivation and providing an interactive student-centered learning environment.

## References

[Albert and Hockemeyer 1997] Albert, D. and Hockemeyer, C.; "Adaptive and dynamic hypertext tutoring systems based on knowledge space theory"; Proc. AI-ED'97, World Conf. on Artificial Intelligence in Education: Knowledge and Media in Learning Systems, Kobe, Japan (1997), 553-555.

[Chan 2007] Chan, Y. Y.; "Teaching queueing theory with an inquiry-based learning approach: A case for applying webquest in a course in simulation and statistical analysis"; Proc. FIE'07, 37th ASEE/IEEE Frontiers in Education Conf. (2007), F3C–(1–6).

[Chang et al. 2006] Chang, A., Chang, M., and Hsieh, A.; "A Treasure Hunting Learning Model for Students Studying History and Culture in the Field with Cellphone"; Proc. ICALT'06, Sixth IEEE Int. Conf. on Advanced Learning Technologies, Los Alamitos, CA, USA (2006), 106-109.

[Conati and Zhao 2004] Conati, C. and Zhao, X.; "Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game"; Proc. IUI'04, Ninth International Conf. on Intelligent User Interfaces, Funchal, Madeira, Portugal (2004), 6-13.

[Di Blas et al. 2004] Di Blas, N., Paolini, P., and Poggi, C.; "Learning by playing an edutainment 3d environment for schools"; Proc. ED-MEDIA'04, World Conf. on Educational Multimedia, Hypermedia and Telecommunications, Lugano, Switzerland (2004), 1313-1320.

[Dodge 1995] Dodge, B.; "Some thoughts about webquests"; (1995) `http://webquest.sdsu.edu/about_webquests.html`.

[Edelson 1998] Edelson, D. C.; "Matching the design of activities to the affordances of software to support inquiry-based learning"; Proc. ICLS'98, Third Int. Conf. on the Learning Sciences, Atlanta, GA, USA (1998), 77-83.

[Edelson et al. 1999] Edelson, D. C., Gordin, D. N., and Pea, R. D.; "Addressing the challenges of inquiry-based learning through technology and curriculum design"; Journal of the Learning Sciences, 8, 3 (1999), 391-450.

[Fan and Yao 2003] Fan, L. and Yao, Y. Y.; "Web-based learning support systems"; Proc. WSS'03, WI/IAT Workshop on Applications, Products and Services of Web-based Support Systems, Halifax, Canada (2003), 43-48.

[Fleissner et al. 2006] Fleissner, S., Chan, Y. Y., Yuen, T. H., and Ng, V.; "Webquest markup language (WQML) for sharable inquiry-based learning"; Computational Science and Its Applications - ICCSA 2006, Lect. Notes Comp. Sci. 3980, Springer, Berlin, Germany (2006), 383-392.

[Hamelin 2004] Hamelin, D.; "Searching the web to develop inquiry and collaborative skills"; Proc. ITiCSE-WGR'04, Working group reports from ITiCSE on Innovation and Technology in Computer Science Education, Leeds, UK (2004), 76-79.

[Hsiao et al. 2006] Hsiao, H. S., Wong, K. H., Wang, M. J., Yu, K. C., Chang, K. E., and Sung, Y. T.; "Using cognitive affective interaction model to construct on-line game for creativity"; Technologies for E-Learning and Digital Entertainment, Lect. Notes Comp. Sci. 3942, Springer, Berlin, Germany (2006), 409-418.

[Jones and Furnas 1987] Jones, W. P. and Furnas, G. W.; "Pictures of relevance: a geometric analysis of similarity measures"; Journal of the American Society for Information Science, 38, 6 (1987), 420-442.

[Jong et al. 2006] Jong, M. S. Y., Shang, J. J., Lee, F. L., Lee, J. H. M., and Law, H. Y.; "Learning online: A comparative study of a situated game-based approach and a traditional web-based approach"; Technologies for E-Learning and Digital Entertainment (2006), 541-551.

[Ke 2008] Ke, F.; "A case study of computer gaming for math: Engaged learning from gameplay?"; Computers & Education, 51, 4 (2008), 1609-1620.

[Khalifa and Lam 2002] Khalifa, M. and Lam, R.; "Web-based learning: effects on learning process and outcome"; IEEE Transactions on Education, 45, 4 (2002), 350-356.

[Kim 2010] Kim, D. W.; "Web-based Learning Support System for Inquiry-Based Learning using Treasure Hunt"; Masters thesis, University of Regina (2010).

[Kim and Yao 2010] Kim, D. W. and Yao, J. T.; "A Web-Based Learning Support System for Inquiry-Based Learning"; Web-Based Support Systems, Springer, London (2010), 125-143.

[Kim 2005] Kim, J. S.; "The effects of a constructivist teaching approach on student academic achievement, self-concept, and learning strategies"; Asia Pacific Education Review, 6, 1 (2005), 7-19.

[Lim 2004] Lim, B. R.; "Challenges and issues in designing inquiry on the web"; British Journal of Educational Technology, 35, 5 (2004), 627-643.

[Llewellyn 2004] Llewellyn, D.; "Teaching High School Science Through Inquiry A Case Study Approach"; Corwin Press (2004).

[Piaget 1976] Piaget, J.; "To understand is to invent: The future of the education"; Penguin Books, New York, NY, USA (1976).

[Power and Kaparthi 2002] Power, D. J. and Kaparthi, S.; "Building web-based decision support systems"; Studies in Informatics and Control, 11, 4 (2002), 291-302.

[Prensky 2003] Prensky, M.; "Digital game-based learning"; ACM Computers in Entertainment, 1, 1 (2003), 21-21.

[Russell 1962] Russell, B.; "An Inquiry into Meaning and Truth"; Penguin Books, Harmondsworth, Middlesex, UK (1962).

[Schrepp 1997] Schrepp, M.; "A generalization of knowledge space theory to problems with more than two answer alternatives"; Journal of Mathematical Psychology, 41, 3 (1997), 237-243.

[Shareef et al. 2006] Shareef, A. F., Kinshuk, K., and Petersen, S. A.; "Distance education in the maldives: Learner support for students in island communities"; Proc. TEDC'06, Fourth IEEE International Workshop on Technology for Education in Developing Countries, Washington, DC, USA (2006), 24-25.

[Shaw et al. 1999] Shaw, E., Johnson, W.L., and Ganeshan, R.; "Pedagogical agents on the web"; Proc. AGENTS'99, Third annual conf. on Autonomous Agents, Seattle, Washington, USA (1999), 283-290.

[Sheth 2003] Sheth, R.; "Avatar technology: Giving a face to the e-learning interface"; The eLearning Developers' Journal (2003), 1-10.

[Spronken-Smith et al. 2008] Spronken-Smith, R. A., Bullard, J., Ray, W., Roberts, C., and Keiffer, A.; "Where might sand dunes be on mars? engaging students through inquiry-based learning in geography"; Journal of Geography in Higher Education, 32, 1 (2008), 71-86.

[Woolf et al. 2002] Woolf, B. P., Reid, J., Stillings, N., Bruno, M., Murray, D., Reese, P., Peterfreund, A., and Rath, K.; "A general platform for inquiry learning"; Proc. the 6th Int. Conf. on Intelligent Tutoring Systems, Springer-Verlag, London, UK (2002), 681-697.

[Xu and Wang 2006] Xu, D. and Wang, H.; "Intelligent agent supported personalization for virtual learning environments"; Decision Support Systems, 42, 2 (2006), 825-843.

[Xu et al. 2008] Xu, F., Yao, Y. Y., and Miao, D.; "Rough set approximations in formal concept analysis and knowledge spaces"; Foundations of Intelligent Systems (2008), 319-328.

[Yager 1991] Yager, R. E.; "The constructivist learning model"; The Science Teacher, 58, 6 (1991), 52-57.

[Yao et al. 2007] Yao, J. T., Kim, D. W., and Herbert, J. P.; "Supporting online learning with games"; Proc. SPIE Symposium on, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2007, Orlando, FL, USA (2007), 65700G-(1-11).

[Yao and Lingras 2003] Yao, J. T. and Lingras, P. (eds.); "Proceedings of WSS'03, WI/IAT Workshop on Applications, Products and Services of Web-based Support Systems"; Saint Marys University, Halifax, Canada (2003).

[Yao and Yao 2003] Yao, J. T. and Yao, Y. Y.; "Web-based support systems"; Proc. WSS'03, WI/IAT Workshop on Applications, Products and Services of Web-based Support Systems, Halifax, Canada (2003), 1-5.

[Zhang and Nunamaker 2003] Zhang, D. and Nunamaker, J. F.; "Powering e-learning in the new millennium: An overview of e-learning and enabling technology"; Information Systems Frontiers, 5, 2 (2003), 207-218.

[Zhang et al. 2004] Zhang, D., Zhao, J. L., Zhou, L., and Nunamaker, J. F.; "Can e-learning replace classroom learning?"; Communications of the ACM, 47, 5 (2004), 75-79.