

## **Item Classification Algorithm for Computer Adaptive Testing Based on Web Services**

**Manop Phankokkrud, Kuntpong Woraratpanya**

Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Thailand

### **ABSTRACT**

**A learning management system (LMS) plays an important role in e-learning. It provides educational services to manage learner profiles and learning contents. One of the core processes of an LMS is the computer adaptive testing (CAT), which helps teachers evaluate student's knowledge capability. Although many CAT systems have been introduced to LMSs, they do not provide the efficient item classification algorithm that supports interoperability. Such interoperability makes systems feasible to efficiently interact with each other. Thus, this paper proposes a CAT framework supporting interoperability by using web service. In this framework, the service of the CAT system is to generate an optimal item set for an item selection module of a CAT web application. For this reason, an item classification module is redesigned and implemented by a web service. Inside this module, there are two significant components for cooperation. The former is an item classification algorithm proposed by means of a triangle-decision tree (TDT) collaborating with genetic algorithms (GAs) for generating the optimal item set, whereas the latter is the web service for making the optimal item set interoperable. The simulation results show that the item classification module can classify accurately item sets and the web services can work properly.**

*Keywords* - computer adaptive testing, web services, service-orientated architecture, learning management systems, genetic algorithms.

### **I. INTRODUCTION**

E-Learning is the convergence of the Internet and learning, and currently becomes an alternative learning method in education. E-learning systems provide processes of delivering the learning contents to learners who have different backgrounds, interests, and locations away from a classroom in order to

maximize the effectiveness of learning. The core of the e-learning systems is learning management systems (LMSs) comprised of a set of functional and educational services to manage learner profiles and learning contents. The principal parts of the LMS are course management and assessment management. The course management handles learning contents such as text, picture, sound, and others, while the assessment.

Inside an LMS, one of the core processes is the student assessment which facilitates teachers to evaluate student's knowledge level; furthermore, the precise measurement helps the student's knowledge development reach their full potential. In student ability estimation, the multiple-choice test is widely used, especially in adaptive learning that items are adapted to approach to the ability level of individual examinees. Therefore, the learning assessment requires the higher precision level for identifying the appropriate item to the examinee. In the past, the adaptive testing approach has been applied to the learning assessment in computer-based learning. It is recognized as "computer adaptive testing" or "computerized adaptive tests" (CAT). In general, CAT is a combination of a measurement theory and computer technology to provide a precise measurement in computer-based learning. A CAT system commonly requires the standard items with item response theory (IRT) parameters stored in the item bank. The item selection, which is a heart procedure of the CAT system, retrieves the suitable items from the item bank and assigns to examinees by assessing their knowledge capabilities. However, CAT will benefit if it can be administered not only on a single platform but also on a cross platform. Unfortunately, current LMSs are developed as web applications on the web server. Both course and assessment databases are installed on the same repository, making it unfeasible to distribute or share

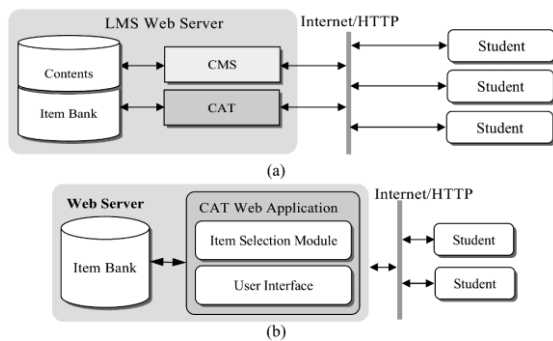


Figure 1: Problem formulation schemes (a) principal components in an LMS and (b) a conventional CAT system.

this information to the others. In addition, existing LMSs do not support the interoperability for information interchanges on across different platforms. For this reason, this paper proposes web service strategies to extend LMS capability in an assessment module to support platform independence.

For developing web services for CAT systems, the following factors are essential; an item-set requirement for adaptive testing and a decision-making technique for an item selection. In item selection module, the recognized decision-making technique is a binary tree [1][2], which is implemented in CAT systems for retrieving suitable items to examinees. However, the item-set requirement of the binary tree is  $2^n - 1$ , where  $n$  is a number of items. Suppose that, in adaptive testing, the requirement of a number of items is 10, so the total item-set requirement of the binary tree is 1023. It is evident that the binary tree is impractical when a number of items are increasing. Furthermore, the binary tree cannot represent transitive relationships of the adjacent nodes of the items in the decision tree. Thus, generating an optimal-decision tree is difficult. Such drawbacks can be eliminated by implementing the decision-making procedure with a suited tree structure. One of the practical solutions is a triangle decision tree (TDT) applying to the CAT systems [3]. In TDT structure, each item is classified into the right location node by determining decision parameters in order to generate an optimal tree. Nevertheless, each node consists of multiple decision parameters. This

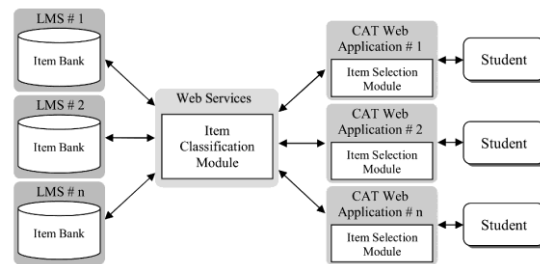


Figure 2: Web service strategy to provide the interoperability in CAT systems.

leads to the difficulty of an item classification. Hence, Genetic Algorithms (GAs) [8], which is an efficient method to solve complex optimization problems, is used so that the optimal-item tree is automatically generated. Consequently, the collaboration of TDT and GAs makes the item classification accurate and supports the interoperability.

The remainder of this paper is organized as follows. Section 2 describes the problem formulation. Computer adaptive testing based on web services is proposed in section 3. In section 4, the simulation system is implemented. The simulation results are discussed in section 5. Finally, the conclusions and future works are presented in section 6.

## II. PROBLEM FORMULATION

An LMS mainly consists of a course management system (CMS) and a computer adaptive testing (CAT) system as shown in Fig. 1(a). CMS is made use of managing learning contents while CAT is made use of managing the learner testing. However, with rapid development of the Internet and technology, the capabilities of CAT and LMS should be enhanced to support the change of technology. Thus this paper identifies the research problems oriented toward improving the existing CAT and LMS systems.

CAT is a combination of a measurement theory and computer technology to provide a precise measurement in computer-based learning. A CAT system requires the standard items with IRT parameters stored in the item bank and the decision-making procedure to assign the suitable item to each

examinee. Nevertheless, a conventional CAT system and an item bank are installed in the single LMS server which does not provide interoperability as shown in Fig. 1(b). This makes it difficult to distribute, exchange, and share information. Fortunately, with technology development, web services can fulfill such a requirement as depicted in Fig. 2. In order to achieve this scheme, the principal components of CAT systems are investigated so that they can perform on across different platforms of LMSs and servers. The item bank and item selection module commonly cooperates in the same place. Based on the interoperability, however, they are decoupled, but exchanged and shared information via item classification module implemented as a web service.

The heart of conventional CAT systems is an item selection procedure, which is usually represented in different forms of decision trees. However, when the item bank and item selection procedure are decoupled, it needs an intermediate module so as to transfer the suitable item set from the item bank to the remote item selection module. Such an intermediate module requires the appropriate decision tree structure and supports the interoperability. From preliminary studies, many CAT systems have been introduced [5][6][7], but there are no appropriate decision trees to support such a requirement. Hence, this paper proposes an item classification module made use of a triangle decision tree (TDT) collaborating with Genetic Algorithms (GAs) to support the interoperability using web service strategies. Based on the TDT structure, tree characteristics and decision rules are defined. Then each item is evaluated and classified into the suited position node in order to generate an optimal decision tree. However, each node consists of statistically complicated information and decision parameters. This leads to the trouble of classifying the IRT parameters, which are multiple variable problems, to generate the optimal item tree. In addition, the complication of a node classification depends on the number of items in a tree structure. That is, the more items, the more complicated classification occurs. This difficulty is one of the optimization problems. The effective method is GAs which has been

extensively used to solve complex optimization problems, especially decision rule problems [8][9]. Therefore, this paper applies GAs to resolve this problem so that an optimal-item tree is automatically generated. Nevertheless, since the number of nodes in the TDT structure is directly proportional to the number of tree levels, the larger number of tree levels unavoidably requires heavy computational time. This drawback is overcome by dividing a set of items into the smaller trees and using the multi-step decision method [10].

Current LMSs do not support the interoperability for content sharing between LMSs and across different platforms. This becomes a limitation of LMSs. Some solutions [11] have been introduced to eliminate this drawback by identifying the essential services and defining all functionalities needed for the interactions between the services. These services provided a common interface between various components to the delivery of learning resources to learners. Nevertheless, the difficulty is still remained, because of the incompatibility of the different vendor components. The interconnection between heterogeneous systems does not make it easy, since existing systems are heavily dependent on some certain operating systems and are built in the homogeneous network. In addition, there exist some difficulties to develop the platform independence for web applications, which need to support some specific features in LMSs. Also, the solution is not flexible in adjusting a data structure and does not provide interoperability.

In order to overcome this trouble, a web service strategy is applied to existing standard LMSs to extend them with the new functionality. Based on a web service concept, an interchange of learning information between LMSs and other applications requires a well-suited service-orientated architecture (SOA). Thus, the SOA needs to design, identify, and create the essential services to support LMSs and CAT web applications for communicating with each other. This paper identifies the functionality needed to enhance interoperability in the subsection 3.2.

### III. METHODOLOGY



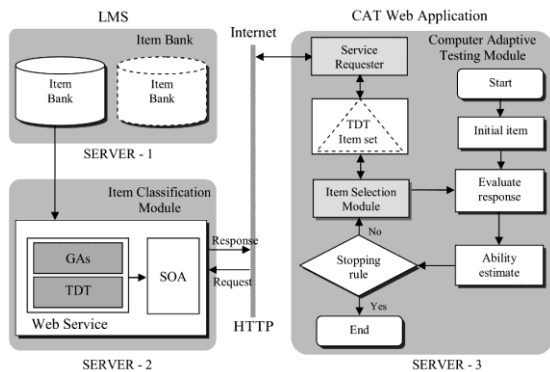


Figure 3: A logical flowchart of computer adaptive testing based on web services.

This section describes the components of the computer adaptive testing based on web services, including TDT and GAs procedure, web service, and SOA. Each component of the proposed method is explained in the following subsections.

### 3.1 Computer Adaptive Testing Framework

A logical scheme of computer adaptive testing based on web service is graphically depicted the principal components in Fig. 3. Such a scheme consists of the item bank, item classification module, and computer adaptive testing module. The proposed scheme differs from the traditional CATs in that the item bank and item selection procedure are decoupled and installed in different servers.

The item bank on the SERVER-1 is a database that collects and stores items. Each item consists of several testing parameters; question, multiple-choice, answer, and IRT-parameter. Since the efficiency of an adaptive testing system depends on the quality of items, all items have to be calibrated and evaluated the IRT parameters by experts. Moreover, the item bank has to provide sufficient items to supply during a test and cover topics and difficulty levels. Such an item bank also supports testing information to an item classification module, which is a core component in generating a TDT item set for item selection procedure at a remote server, SERVER-3. On the SERVER-2, the item classification module is developed by means of web service technology. It consists of TDT, GAs, and SOA. TDT is a symmetrical decision structure for classifying items to each decision node. It requires the more effective

method to generate the optimal-decision tree. Here, the suitable approach is GAs applied to TDT. In addition, SOA is designed as the data structure of the TDT item set for delivering to a connected CAT system.

On the SERVER-3, a computer adaptive testing module is the online assessment software to work with the adaptive testing. For student assessments, the operation starts with assigning a moderate difficulty item to the examinees. Then the response from such an examinee is evaluated. Then the item score is immediately updated. If the answer is correct, the test statistically estimates the student's ability with the higher scores than the previous estimation and the item selection procedure retrieves a question that matches to the examinee's ability with the higher difficulty. Conversely, a question with the lower difficulty is presented. Predicting and updating the student ability is based on all prior answers. This operation is repeated until a maximum number of items are reached or the ability estimation is done.

#### 3.1.1 Triangle Decision Tree

In general, the significant requirement of a CAT system is providing an effective decision tree for item bank retrievals. However, in order to make the CAT system available to perform on across different LMS platforms, the CAT service is evidently identified. In this paper the CAT service is generating an optimal decision tree for an item selection procedure, since the CAT web application is isolated from the item bank as shown in Fig. 3. This makes the decision tree structure and its property important to support the service. Therefore, the structure and property of TDT is firstly analyzed. Unlike the classical binary tree shown in Fig. 4(a), TDT is a graphical decision tree model as depicted in Fig. 4(b). Its structure is composed of nodes and branches. A node represents testing parameters, whereas a branch indicates the value of target parameters. Each node has only two branches; left and right branches, for child nodes. The left branch denotes a wrong outcome while the right branch denotes a correct outcome. For instance, at the current node (root node), the next item of item number 1 is item number 3, if the answer is correct. On the other hand, the next is item number 2, if the

answer is wrong. Each node represents three IRT parameters; discrimination, difficulty, and pseudo-guess, respectively, as shown in Fig. 4(b). For decision paths, the root node is the starting point of TDT and moves down through the tree branch corresponding to the value of the parameters. With this structure, the items are classified and an optimal-decision tree is generated. Moreover, only one solution is obtained.

In practice, IRT parameters of each item, including discrimination, difficulty, and pseudo-gauss, are determined and located at the suited node. However, the IRT parameter mapping is an essential process to automatically generate the optimal-decision tree. To do this, in the first phase the characteristics of the TDT structure are analyzed in the form of a mathematical model. With symmetrical structure of TDT, the number of items,  $M$ , can be calculated by equation (1).

$$M = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} \quad (1)$$

where  $n$  is the number of levels.

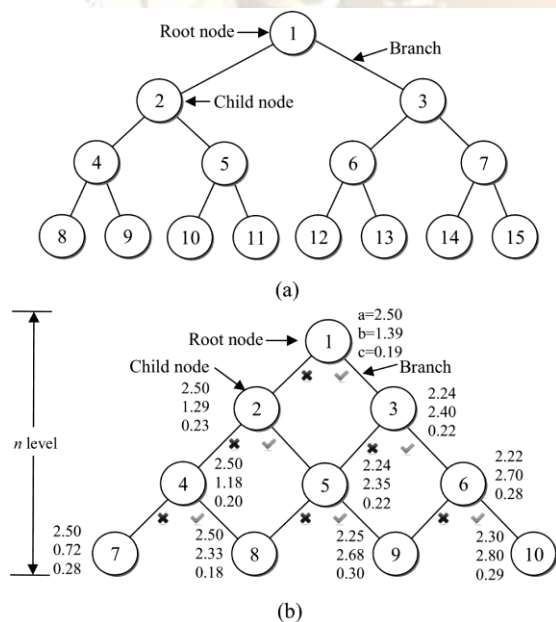


Figure 4: Decision tree structures (a) a classical binary tree, and (b) a TDT structure with IRT parameters:  $a$ ,  $b$ , and  $c$ , respectively.

In general, the number of items,  $M$ , in the TDT structure denotes the minimum requirement of the number of items for the item bank. Simultaneously, the number of levels,  $n$ , represents actually the number of items for the test. Basically,  $M > n$ . For instance, if TDT has four levels ( $n = 4$ ), the number of items for this tree is 10. Compared with the binary tree structure, the minimum requirement of items is 15 as illustrated in Fig. 4(a). This shows that increasing the number of levels in the binary tree leads to exponentially grow the minimum requirement of items in the item bank.

In the second phase, the testing parameters have to be obtained their relationships in order to generate the TDT structure in an optimal form. In Fig. 4(b), the difficulty of IRT-parameters is set up to the highest priority of the decision, while discrimination and pseudo-guessing are the minor decision parameters as recommended in [12][13]. The relationship between nodes and branches in TDT can be arranged as shown in Fig. 5(a). A relationship between one parent and two child nodes is determined by equation (2), whereas a relationship between nodes in the same level is obtained by equation (3).

$$b_{i+1,j} < b_{ij} < b_{i+1,j+1} \quad (2)$$

$$b_{i,1} < b_{i,2} < \dots < b_{i,j} \quad (3)$$

where  $b_{ij}$  is the item difficulty of  $i^{\text{th}}$  level in  $j^{\text{th}}$  order. This relationship called “transitivity” is very important rules to maintain the TDT structure to be optimal. Furthermore, the optimal TDT structure can be expressed ability patterns of examinees in the form of route tracking as demonstrated in Fig. 5 (b). If a route tracking trend falls into the right-hand side of TDT, it implies that the examinees’ ability is higher. Otherwise, the examinees’ ability is lower.

Finally, based on the structure illustrated in Fig. 4 and Fig. 5, a comparison between a binary tree and TDT can be summarized in Table 1. The advantage of TDT can be described as follows. TDT has exactly a number of nodes; this implies that teachers can accurately assign a number of items for each part of testing. TDT has the transitive relationships between items in the same item set and the same level as illustrated by equation (2) and equation (3). Such

Table 1  
A comparison of classical binary tree and TDT structures

	Binary Tree	TDT
1. A number of nodes	<ul style="list-style-type: none"> <li>• <math>M=2^n-1</math> (<math>n</math>: a number of levels in a tree)</li> </ul>	<ul style="list-style-type: none"> <li>• <math>M = (n/2)(n+1)</math></li> </ul>
2. Parent's child nodes	<ul style="list-style-type: none"> <li>• Zero, one, or two nodes</li> </ul>	<ul style="list-style-type: none"> <li>• Exactly two nodes</li> </ul>
3. Node relationship in the same level	<ul style="list-style-type: none"> <li>• No relationship</li> </ul>	<ul style="list-style-type: none"> <li>• Equation (2)</li> </ul>
4. Node relationship in a tree	<ul style="list-style-type: none"> <li>• No relationship</li> </ul>	<ul style="list-style-type: none"> <li>• Equation (3)</li> </ul>
5. Ability pattern tracking	<ul style="list-style-type: none"> <li>• Implicit pattern tracking</li> </ul>	<ul style="list-style-type: none"> <li>• Explicit pattern tracking</li> </ul>
6. Node arrangement	<ul style="list-style-type: none"> <li>• Multi-solution</li> </ul>	<ul style="list-style-type: none"> <li>• Only one solution</li> </ul>

relationships lead to generating the optimal tree. Moreover, TDT expresses the route of ability pattern tracking helping teachers analyze and evaluate students' ability. Finally, TDT has only one optimal solution in classifying the items to each node. It is summarized that TDT is able to overcome some disadvantages and provide most appropriate structure for decision trees. For these reasons, TDT is appropriate for item classification module in the CAT service.

Unfortunately, when the number of nodes in the TDT structure is increasing, the item classification takes more computational time. Thus this paper overcomes this drawback by dividing the larger TDT into the smaller TDTs based on the multi-step decision [10]. As shown in Fig. 6, the larger triangle representing a TDT item set is divided into the smaller triangles with  $s$  levels as demonstrated in the right-hand side of the figure, such that  $l$  is a number of levels in the larger TDT and  $s$  is a number of portions. So, there are  $s$  portions. Each portion contains  $l/s$  sublevels. For instance, a test provides sixty items for students. In this case, such items can be divided into ten portions. Hence, each portion consists of six items in six sublevels. That is, the larger TDT item set is separated into the smaller TDT item sets. In this case, the minimum requirement of a number of items is reduced from 1830  $((60/2)(60+1))$  to 210  $(6(6/2)(6+1))$ .

Normally, an item bank contains many item sets which are grouped by the item difficulty criterion as depicted in Fig. 7. Reducing a size of the larger TDT to a set of smaller TDTs also provides a variety of item sets. In Fig. 7, at the top of the sublevel, the item

classification module can randomly retrieve one of the item sets (such as A, B, C, or D) with the same difficulty level to construct completed TDT. Such random retrieval process repeatedly fulfills from the decision step 1 to the decision step  $l$ . In each step,  $l/s$  items will be appended in the larger TDT set until the minimum number of items is fulfilled. Then, the completed item set is delivered to the CAT web application at the remote site. Not only does this improvement decrease the computational time of the larger number of TDT levels, but also retain the item-classification accuracy. In addition, it makes the system available to generate the various patterns of item sets while the difficulty level of the item set is unchanged. In this case, the different patterns of item sets are generated by equation (4).

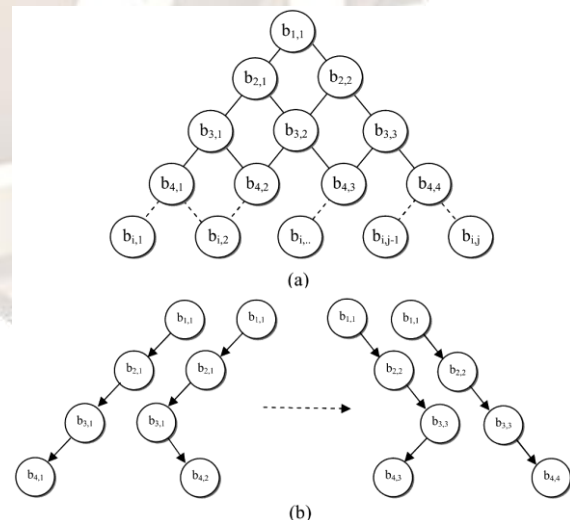


Figure 5: (a) A relationship between nodes and branches in the form of difficulty parameters and (b) ability patterns.



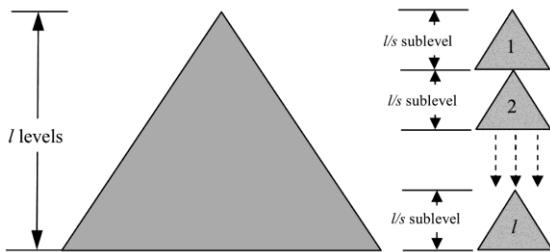


Figure 6: A larger TDT structure, represented with a triangle shape, divided into the smaller TDTs.

$$P = T_1 \times T_2 \times T_3 \times \dots \times T_m \quad (4)$$

where  $T_1, T_2, T_3, \dots, T_m$  are the number of item sets.

### 3.1.2 GAs procedure

The role of a GAs procedure in the proposed scheme is an intelligent item-classification for automatically gene-rating an optimal-decision tree. The GAs procedure transforms the classification problems to a mathematical model and represents a solution in a various size of trees. The standard GAs operation consists of chromosome encoding, fitness evaluation, selection, crossover, and mutation. The overall operation flowchart of the GAs is shown in Fig. 8.

In general, the GAs procedure starts with an initial condition. The problem is identified and transformed into a chromosome structure. Such a transformation is called “process encoding”. There are four methods for process encoding; permutation, binary, tree, and value encoding [14]. In this research, the permutation encoding is a suitable method for directly encoding item IDs into a chromosome structure. The outcomes of the permutation encoding are called “integer chromosomes”. Each chromosome is encoded with item IDs for constructing a set of items as shown in Fig. 9(a). These item IDs are randomly generated a set of chromosomes. Such chromosomes are called “population”. Then, a pair of chromosomes in population is regenerated the new offspring by a crossover operator. In this paper, a two-point crossover operator is used, since it can provide more disruptive than a one-point crossover as illustrated in Fig. 9(b). This operation yields a new population that inherits the best feature from the previous population. The mutation operator is applied to create random changes in various chromosomes. A simple way to

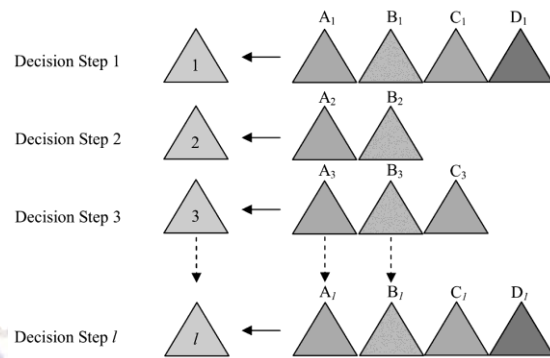


Figure 7: Various item-set generations.

achieve a mutation is alternating one or more chromosomes. Here, the swap mutation method is used as depicted in Fig. 9(c). A swap mutation is used for interchanging the item IDs in the chromosomes designed for an integer chromosome structure [15]. Finally, the selection operator is applied to the mutation chromosomes. The chromosome with a high fitness value is selected to be alive in the next generation. Otherwise, it is eliminated. A Roulette wheel selection is an approach to generate a chance of selecting a chromosome to be the parent in the next generation [16]. This chance is directly proportional to the parent’s fitness. An example of this operation is shown in Fig. 9(d). The termination of GAs is defined by the lowest value of the cumulative fitness function. That is, if the difference of cumulative fitness value in equation (6) is lower than a threshold, the GAs is terminated. The stop-ping condition is based on the theoretical tendency of the cumulative fitness value as shown in Fig. 10.

The fitness function is important for a GAs termination. A score of each chromosome is determined by equation (4). This score indicates the probability of chromosomes to be alive. Thus, a Hamiltonian cycle method [17] is applied to a pair of nodes to obtain the fitness value,  $f_i$ . Then the cumulative value of fitness function,  $F$ , is calculated by equation (6).

$$f_i = \sqrt{W_a(a_i - a_i)^2 + W_b(b_i - b_i)^2 + W_c(c_i - c_i)^2} \quad (5)$$

where  $a$ ,  $b$ , and  $c$  denote the discrimination, difficulty, pseudo-guessing parameters of items, and

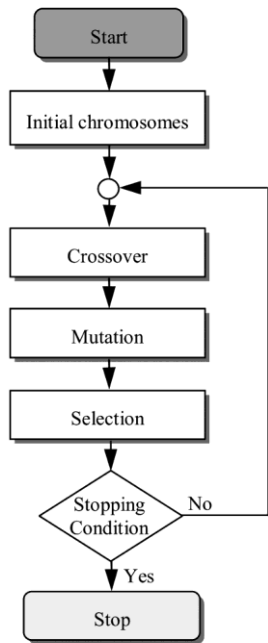


Figure 8: A flowchart of a GAs procedure.

$w_a$ ,  $w_b$ , and  $w_c$  are weighting factors of item parameters,  $a$ ,  $b$ , and  $c$ , respectively.

$$F = f_1 + f_2 + f_3 + \dots + f_{n-1} \quad (6)$$

where  $f_1, f_2, f_3, \dots, f_{n-1}$  are the fitness value.

$$\varepsilon = F_{m-1} - F_m \quad (7)$$

where  $F_m$  is the cumulative fitness value of chromosome at  $m^{\text{th}}$  generation and  $\varepsilon$  is the difference of cumulative fitness value at  $m^{\text{th}}$  and  $(m-1)^{\text{th}}$  generations. In equation (5), the weighting factors of discrimination, difficulty, and pseudo-guessing parameters are assigned by the priority order. In adaptive testing, the difficulty of items directly relates with the examinee ability, thus a weighting value of the difficulty parameter is set to greater than the others. The discrimination and pseudo-guessing parameters are lower priorities for the decision making, thereby defining them to be an equal weighting but less than that of the difficulty parameter.

The outcome of the terminated operation is a solution chromosome consisting of an item ID sequence of an item set. The chromosome length is equal to the number of nodes of TDT. Then the solution

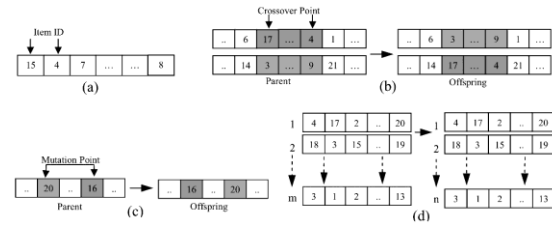


Figure 9: Operations of a GA procedure (a) genetic permutation encoding, (b) two-point crossover, (c) mutation, and (d) selection.

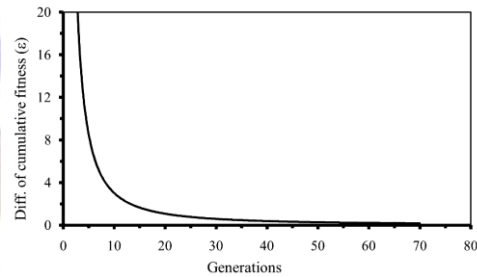


Figure 10: A theoretical tendency line.

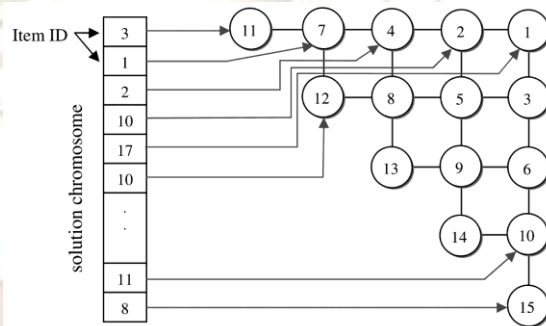


Figure 11: Mapping of a chromosome solution to a TDT structure.

chromosome is mapped to each node as depicted in Fig.11. The top item ID of the solution chromosome is mapped to the top-left node of TDT, and the next item IDs are consecutively matched to the next nodes. This result is the optimal TDT generation. Although the TDT rules and GAs procedure are successfully developed to classify the items, a method to deliver the TDT item set to a CAT web application is required. Thus the most important phase is the designing of service-oriented architecture (SOA) for web services. All designing phases are described in next subsection.

### 3.2 Service-oriented architecture design

An SOA design is a process that transforms the item's data structure to services. In this paper, such



services provide some essential data for CAT web applications and for student's assessments. Such services also use to transfer the items from the item bank to the CAT web application [18]. Here, the item data structure is expressed in an XML schema developed according to the standards [19]. The designing process starts with defining the data structure of an item bank database. This database uses a basic data structure for testing parameters, including questions, multiple choices, answers, and IRT parameters. These testing parameters are defined as XML elements, where CA, CB, CC, and CD are the multiple-choice, QUE is the question, and QANS is the answer, as shown in Table 2, where IA, IB, and IC denote discrimination, difficulty, and pseudo-guess of the IRT parameters, respectively.

An SOAP message is encoded in an XML format that is defined as a specification for an XML-based distributed computing infrastructure. The SOAP message is described a framework and endpoint. In order to test the service and exchange messages between servers, the SOAP messages and responses are created.

In order for CAT web application to be able to access the services, it is necessary to provide the Web Services Description Language (WSDL) description, which is used to syntactically describe the functionality of web services. In addition, WSDL uses XML elements to describe a network service. An XML schema, SOAP message, and WSDL of Web service were described in [18].

#### IV. SIMULATION SYSTEM

In order to evaluate the proposed scheme, the simulation system is conducted using MySQL Server database, PHP, and conventional SOAP web service. The MySQL server is set up as the database, since it is the world's popular database system designed for speed, power, precision, fully compatible with PHP, and free. The web service and CAT web application are developed by using a general software architecture principle based on the instructional system design (ISD) model, which is an industry standard for instructional design of educational applications [20]. Additionally, items, test data from the Computer Networking Test, are calibrated by experts and are stored in the item bank with IRT parameters.

In Fig. 12, the simulation system is set up as follows. First, the SOAP web service is developed by PHP and hosted on the web server operating on Linux operating system and Apache web server. The item bank is managed and stored in the MySQL server database. Second, the web server operates on Windows 7 and IIS web server for supporting the CAT web application developed by PHP. It differs

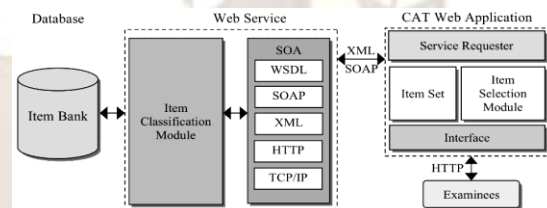


Figure 12: Overall architecture of a simulation system.

Table 2  
A data structure of an item set

Field	Description	Data type
Index	Item index	Number
QUE	Question	String
CA	Choice A	String
CB	Choice B	String
CC	Choice C	String
CD	Choice D	String
QANS	Answer	String
IA	IRT Discrimination	Single
IB	IRT Difficulty	Single
IC	IRT Pseudo-guess	Single

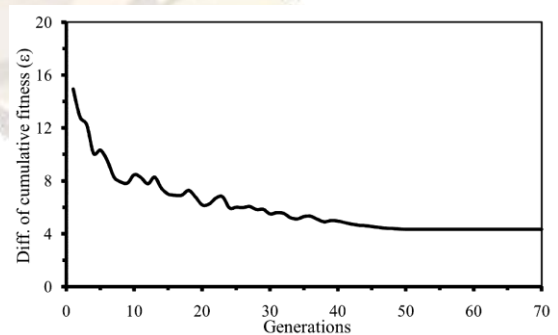


Figure 13: An experimental tendency line.

Table 3  
The GA parameters

Parameters	Value(s)
Population size	5000
Chromosome length	21
Cross over rate	0.8
Mutation rate	0.001
Weighting Factors	$W_a = 0.25; W_b = 0.5; W_c = 0.25$

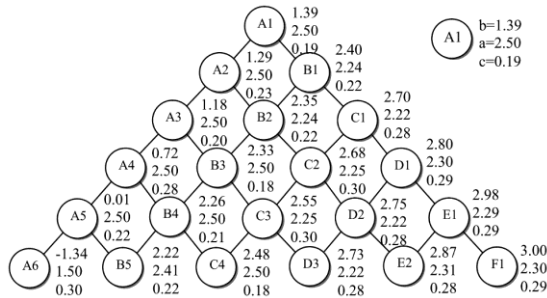


Figure 14: A six-level standard pattern of a TDT diagram.

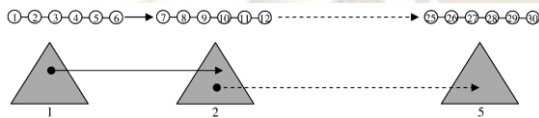


Figure 15: A multi-step decision diagram for 30 items of testing.

from the first web server in that there is no database on this web server.

Fig. 12 shows overall architecture of the simulation system. The principal components are composed of data storage, a web service, and a CAT web application. The data storage is a database that stores items with IRT parameters; it is called "Item Bank". The web service is developed by using the standard SOAP. It contains an item classification module and supplement components, i.e., TDT and GAs procedure, which are developed by PHP. The items are classified into item sets for the request of connected applications. The web service also comprises of the supplement components; WSDL, XML, TCP/IP, HTTP, SOAP, and UDDI. These components are used to transfer the data from a web service to other applications. Both data storage and web services are hosted on the same server. Finally, A CAT web application, an online assessment, is developed by PHP and HTML. It has two main functions; adaptive testing module and user interface. The adaptive testing module controls the testing

Table 4  
Experimental results of transitive relationships of difficulty parameters

Level	Difficulty parameter in the same level
1	1.39
2	$1.29 < 2.40$
3	$1.18 < 2.35 < 2.70$
4	$0.72 < 2.33 < 2.68 < 2.80$
5	$0.01 < 2.26 < 2.55 < 2.75 < 2.98$
6	$-1.34 < 2.22 < 2.48 < 2.73 < 2.87 < 3.00$

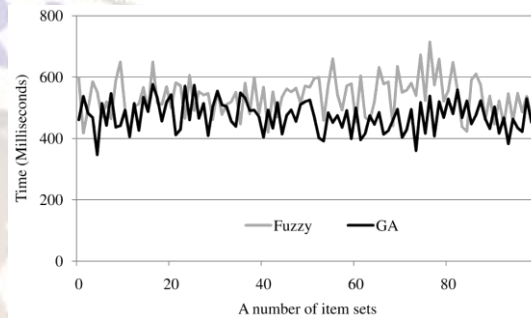


Figure 16: A comparison of the computational time of GAs and Fuzzy inference.

process whereas the user interface interacts with the students.

## V. RESULTS AND DISCUSSION

In this section, the simulation system is evaluated its performance both an item classification module and a web service. The item classification module is tested in terms of classification accuracy and computational time, whereas the web service is evaluated in two cases, the interoperability and efficiency. The interoperability is tested the accuracy of the interconnection data, while the efficiency is tested the speed of communication by measuring a response time. All parameters are defined as the following subsections in order to evaluate the proposed system.

### 5.1 Item classification module testing

In order to test the item classification module, the TDT classification rule and GAs procedure are precisely verified step-by-step. In this experiment, all parameters of GAs procedure are set up as recommended in [17] and exhibited in Table 3. The GAs procedure is carried out by three parameters; population size, crossover rate, and mutation rate. The efficiency in solving the item classification problem depends on IRT-parameters. Hence, the weighting factors of IRT-parameters are set up as

Table 5  
A performance comparison between GAs and fuzzy inference

Method	Accuracy	Average Computational Time (msec)	Flexibility
GAs	100%	473	Variant
Fuzzy	100%	537	Invariant

Table 6  
A part of the first five service response times (milliseconds) and message size (bytes) of string and integer

Number	Message Size (bytes)		Time (msec)	
	String	Integer	String	Integer
1	916	3	677	473
2	929	3	684	464
3	886	3	679	459
4	874	2	692	496
5	948	3	710	469

shown in Table 3. Based on 34 item sets for test-set data, the GAs gives the results as it terminates. It is found that fitness values carry on the theoretical tendency line as illustrated in Fig. 13. The total number of evolved generations is 70. The initial fitness value is 14.95 and the optimal fitness value is 4.34. The result of cumulative fitness difference is consistent with the theoretical tendency line.

From this experiment, when compared with the 6-level standard pattern in Fig. 14, an every node is located at the right position. Furthermore, the accuracy of the item classification is proven by TDT rules in equation (3). That is, the transitive relationship of the difficulty parameter between nodes in the same level. Table 4 shows the transitive relationship of the difficulty parameters in each level of TDT.

In addition, the efficiency of GAs procedure is evaluated by comparing with the fuzzy inference method [20]. The accuracy is tested by inspecting the classified items set. GAs and fuzzy methods give the same results as shown in Table 5; however, by varying the number of items, which directly relates with the number of tree levels, the fuzzy inference is invariant to the change of the TDT level, whereas GAs is absolutely adjustable. In case of computational time, the larger TDT item set is equally divided into the smaller sets. This split makes a number of items in TDT's structure reducing, thus the computational time is decreased. For instance, a

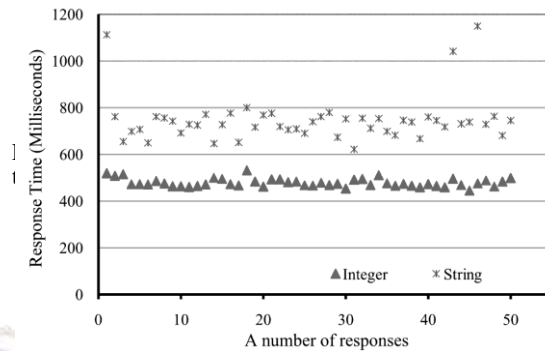


Figure 17: Response time of string and integer for testing.

number of items in 30 levels of TDT are 465 items. It is reduced to 105 items in case of a 5-step decision as shown in Fig. 15. This performance is demonstrated in Fig. 16. Moreover, the multi-step decision method makes the system can randomly select another item set. This improvement not only can solve the increasing computational time in case of the larger number of TDT levels, but also can randomly provide the various item sets.

## 5.2 Web service performance testing

In case of web services, the simulation evaluation includes the interoperability and efficiency. For interoperability evaluation, it starts with the CAT web application sending the item request message to the web service via the Internet. When the web service receives the request message, it retrieves the items from the item bank database. Then, these items are classified by GAs procedure based on the TDT classification rules. The classified items are encapsulated in a XML format and sent back to the CAT web application. Finally, the CAT web application is used the received items in the adaptive testing process. This experiment is illustrated that the web service and CAT web application can interact with each other via the Internet and SOA works properly. The interoperability makes the items transferable from the item bank to the CAT web application. Furthermore, it leads to the items sharable and reusable in other applications.

For web service efficiency, the simulation evaluation focuses on the speed of network communication and the computational time of item classification module. Therefore, the critical factors of benchmarking are two different data types: integer data type and string data type, as described in [21]. The integer data type is used for testing the communication network speed,



whereas the string data type is used for testing a service that produces a message from the item bank. The communication network speed is easily measured with a response time of sending an integer data type as a service. However, in this experiment, the measurement focuses on the impact on GAs operation time processed by varying a number of items in a message. Hence, measuring the response time of the service request implies the summation of the communication network time, the transmission time of the item from database, and GAs operation time.

Table 6 shows the results of integers and strings for testing the communication network speed and the service response time. The evaluation measures the messages size in bytes and the message's response time in milliseconds. As shown in Fig. 17, the response time of integer is less than the response time of string. The gap between these response times is the GAs operation time. The less response time implies that the faster and higher performance of web services.

## VI. CONCLUSION AND FUTURE WORKS

This paper proposes a noble CAT framework that supports platform independence. The significant contribution of such a framework is developing a CAT system as a service. The proposed framework provides item banks and item selection modules of CAT systems to operate across different platforms of LMSs and servers. The item classification module is an important component that makes the service for CAT web application feasible. It connects to the item bank and manages the suitable item sets for the item selection module of the CAT web application. Based on this framework, the item classification module is designed and implemented by means of the TDT structure collaborating with GAs. It is found that the item classification module can provide important features for a service such as reducing the minimum requirements of item sets, supporting transitive relationships of adjacent nodes, explicating ability patterns of students, and providing only one solution of decision trees. Furthermore, in order to fulfill the requirement of the noble CAT framework, the interoperability is essential. The web service that is the best solution for the interoperability applies to the item classification module. Finally, the performance of the simulation system based on the noble CAT framework is evaluated in both the item classification

module and web service. The simulation results show that the item classification module can correctly classify the item sets based on TDT rules and GAs procedure, and the web service can make the item sets interoperable for delivering them to the CAT web application.

In future works, the item classification module will be improved for automatically adjusting the tree size in order to reduce a number of items, while the completed web services for LMSs will be extended to the large scale deployments by connecting to the multiple LMS servers in distributed locations and different kinds of LMSs.

## REFERENCES

- [1] T. Hsieh, and B. Kuo, "Information-based Item Selection with Blocking Strategy based on a Bayesian network," *Journal WSEAS Transactions on Information Science and Applications* archive, 7(9), 2010, 1160-1169.
- [2] J.A. Collins, J.E. Geer, and S.X. Huang, "Adaptive Assessment Using Granularity Hierarchies and Bayesian Nets," *ITS '96 Proceedings of the Third International Conference on Intelligent Tutoring Systems*, 1996, 569-557.
- [3] M. Phankokkrud, and K. Woraratpanya, "An automated decision system for computer adaptive testing using genetic algorithms," *The 9th ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD2008)*, 2008, 655-660.
- [4] J. H. Holland, "Adaptation in natural and artificial systems", MI: University of Michigan Press, 1975.
- [5] E. Millán, M. Trella, J. L. Pérez-de-la-Cruz, and R. Conejo, "Using Bayesian Networks in Computerized Adaptive Tests," *Computers and Education in the 21st Century*, 2000, 217-228.
- [6] P. Salcedo, M. Pinninghoff, and R. Contreras, "Computerized Adaptive Test and Item Response Theory on a Distance Education Platform," *Lecture Notes in Computer Science*, 3562, 2005, 613-621.
- [7] M.C. Desmarais, and X. Pu, "Computer adaptive testing: Comparison of a probabilistic network approach with item response theory," *Proceedings of the 10th International Conference on User Modeling (UM'2005)*, 2005, 392-396.
- [8] T. C. Chen, and T.C. Hsu, "A GAs based approach for mining breast cancer pattern," *Expert Systems with Applications*, 30(4), 2006, 674-681.
- [9] D. R. Carvalho, A. A. Freitas, "A Hybrid Decision Tree/Genetic Algorithm Method for Data Mining", *Information Science*, 163(1-3), 2004, 13-35.

- [10] Y. Lai, and S. Sastry, "Edge-Valued Binary Decision Diagrams for Multi-Level Hierarchical Verification", 29th ACM/IEEE Design Automation Conference, pp. 608-613, 1992.
- [11] A. Grewal, S. Rai, R. Phillips, and C. C. Fung, "The e-learning lifecycle and its services: The web services approach," Proceedings of the second international conference on e-learning for knowledge-based society, 2005, 4.1-4.8.
- [12] C. M. Chen, and L. J. Duh, "Personalized web-based tutoring system based on fuzzy item response theory," Expert Systems with Applications, 34, 2008, 2298-2315.
- [13] J. Q. Tian, D. M. Miao, X. Zhu, and J. J. Gong, "An Introduction to the Computerized Adaptive Testing", US-China Education Review, 4(1), 2007, 72-81.
- [14] B. Wang, Y. Q. Chen, "A genetic algorithm with chromosome-repairing technique for polygonal approximation of digital curves". Lecture Notes in Computer Science, 3612, 2005, 822-831.
- [15] E. Ryan, R. M. A. Azad, and C. Ryan, "On the performance of genetic operators and the random key representation," Lecture Notes in Computer Science, 3003, 2004, 162-173.
- [16] G.M. Megson, and I.M. Blandy, "Synthesis of a systolic array genetic algorithm," 11th international parallel and distributed processing symposium, 1998, 316-310.
- [17] S. Baluja, and R. Caruana, "Removing the genetics from the standard genetic algorithm," Proceedings of the twelfth international conference on machine learning, 1995, 38-46.
- [18] M. Phankokkruad, K. Woraratpanya, "Web service architecture for computer-adaptive testing on e-learning," International Journal of Behavioral, Cognitive, Educational and Psychological Sciences, 1(1), 2009, 43-47.
- [19] IEEE LTSC, "Draft standard for learning technology – learning technology systems architecture (LTSA)", Learning Technology Standards Committee of the IEEE computer society, 2003.
- [20] X. Liu, T. Shaohua, S.H. Ong, and V. Srinivasan, "Map recognition using an adaptive fuzzy pyramid decision approach", IEEE region 10 ninth annual international conference, 1994, 918-922.
- [21] H. Hamad, M. Saad, R. Abed, "Performance Evaluation of RESTful Web Services for Mobile Devices," International Arab Journal of e-Technology, 1(3), 2010, 72-78.