AN EFFECTIVE MULTIMEDIA ITEM SHELL DESIGN FOR INDIVIDUALIZED EDUCATION[1]: THE CROME PROJECT

Irene Cheng [2, 3] and Anup Basu [2]

[2]Dept. of Computing Science, University of Alberta, Canada
[3]Dept. of Computer and Information Science, University of Pennsylvania, USA
Email: chenglin@seas.upenn.edu, anup@cs.ualberta.ca

ABSTRACT

There are several advantages to creating multimedia item types and applying computer-based adaptive testing in education. First is the capability to motivate learning by making the learners feel more engaged and in an interactive environment. Second is a better concept representation, which is not possible in conventional multiple-choice tests. Third is the advantage of individualized curriculum design, rather than a curriculum designed for an average student. Fourth is a good choice of the next question, associated with the appropriate difficulty level based on a student's response to the current question. However, many issues need to be addressed when achieving these goals, including: (a) the large number of item types required to represent the current multiple choice questions in multimedia formats; (b) the criterion used to determine the difficulty level of a multimedia question item; and (c) the methodology applied to the question selection process for individual students. In this paper, we propose a multimedia item shell design that not only reduces the number of item types required, but also computes difficulty level of an item automatically. The concept of question seed is introduced to make content creation more cost-effective. The proposed item shell framework facilitates efficient communication between user responses at the client, and the scoring agents integrated with a student ability assessor at the server. We also describe approaches for automatically estimating difficulty level of questions, and discuss preliminary evaluation of multimedia item types by students.

1. INTRODUCTION

Along with content creation, multimedia has potential for use in both *knowledge acquisition* and *innovative testing*. Rather than traditional paper-and-pencil formats, audio, video, graphics and animation are being conceived as alternative means for more effective learning and testing in the future [1, 3, 5-13]. One advantage of computer-based learning and testing is to advance education to *individualization*; instead of handing out the same set of learning or testing material to the entire class, the next tutorial or exam question can be adaptively selected based on the current performance of an individual student. Since the learning curve of every student is different, individualization is an effective approach for educators and teachers, who can monitor a student's progress better than before using informative and summative computer generated information, and can provide immediate assistance if necessary. The challenge of adopting such an approach is the large collection of tutorials and questions in the database, which needs to cater to all learning levels, and has to be available on demand. Differing from traditional multiple-choice item formats that can easily accommodate different questions, multimedia items require specific screen layouts depending on a question's contents and the type of media used. The programming and development cost may outweigh the benefits of individualization if each question requires a new design and implementation. In a distance-learning environment, the question content can easily overload the server-client capability if the transmitted data is not properly designed and regulated. Another advantage of computer-based learning is to make designed and organized curriculum centrally available, through wired or wireless

[1] This project is funded by grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada; Informatics Circle of Research Excellence (iCORE), Alberta; and Castle Rock Research Corp., Canada. Parts of this paper were presented at a special session in IEEE Int. Conf. on Multimedia and Expo, Beijing, 2007.

networks, to remote locations where resources are insufficient to prepare educational materials locally.

In this paper we propose a novel approach for designing multimedia item shells to address the issues of scalability, reusability and portability. Our design takes into consideration: (a) determining question difficulty, (b) assigning scores, and (c) selecting questions automatically. The rest of the paper is organized as follows: terminology used in this work is described in Section 2. Section 3 discusses the requirements of an effective multimedia item shell. Section 4 introduces our design strategy, where some of the examples implemented are also described. Section 5 discusses the testing and scoring strategy. Section 6 describes a brief student evaluation of some of our multimedia item types. Finally, Section 7 gives the conclusion and discusses future work.

2. TERMINOLOGY

We use the following terminology in this work:

Item − is used in the same context as question or question item.

Multiple Choice (MC) Item Shell − is a template used to generate MC items. Multiple questions can be generated using the same shell by inputting different text to describe the question and the choices.

Multimedia Item Shell − is a template used to generate items with similar characteristics. While a MC item shell has a fixed screen layout, a multimedia item shell can be mapped onto different screen layouts. We use a category code to control such mapping. The item shells implemented in our framework are Multimedia Item Shells.

Parameters − are control variables. By assigning different values to a parameter, multiple items with different screen layouts can be generated.

3. ITEM SHELL REQUIREMENTS

Requiring special software installation prior to executing an application often turns learners away. In order to motivate and engage learners, runtime support of common browsers such as Internet Explorer and Fire Fox should be considered. Furthermore, in an adaptive testing environment, online communication with the server is required; therefore, running a testing session as an offline application is not an option. In order to design effective multimedia item shells, we also need to consider the following:

1. Understanding the current pen-and-paper question contents and screen layout, and determining what type of multimedia content is a good match for each question.
2. Designing a format conversion system by incorporating multimedia content so that the concepts associated with a question are better presented. The idea is to make a student feel more engaged and motivated. It is important that the format conversion does not alter the difficulty of a question; otherwise, the curriculum needs to be reorganized. For example, if a Grade 6 multiple-choice question requires a student to select the organ not located on the body, and after conversion, the new format requires the student to drag and drop the organs to the correct location of the body (Figure 1), then the question difficulty is increased because the student needs to have additional knowledge of an organ's positions.
3. Enhancing the new format to a generic item shell with the capability of enabling or disabling embedded multimedia types by changing some control values. For example, the drag and drop item shell shown in Figure 1 can be designed for dragging 2D images and 3D objects, as well as text. An item shell can also be designed to embed audio, video, animation, still picture and text description, so that question items of similar screen layouts requiring one or a combination of these multimedia content can share the same item shell.

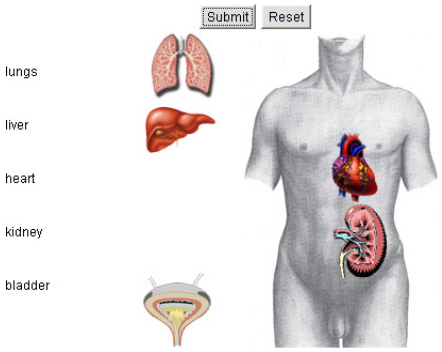

Figure 1: Changing a multiple-choice question to a drag and drop question may increase the difficulty level.

4. Extending an item shell to accommodate multiple screen layouts. Completing an electric circuit may sound very different from picking groceries for a shopping cart (Figure 2 (a) and (b)), but both can be handled by the same item shell in our design. Note that the same item shell can also be used to present the following layouts:

(a) Ordering components in a specified sequence;
(b) Choosing the correct operator to complete an equation;
(c) Dragging the correct descriptions to different locations of an image;
(d) Spelling a word by rearranging the alphabets;
(e) Mapping an image with the correct description; and,
(f) Classifying components into their correct categories.

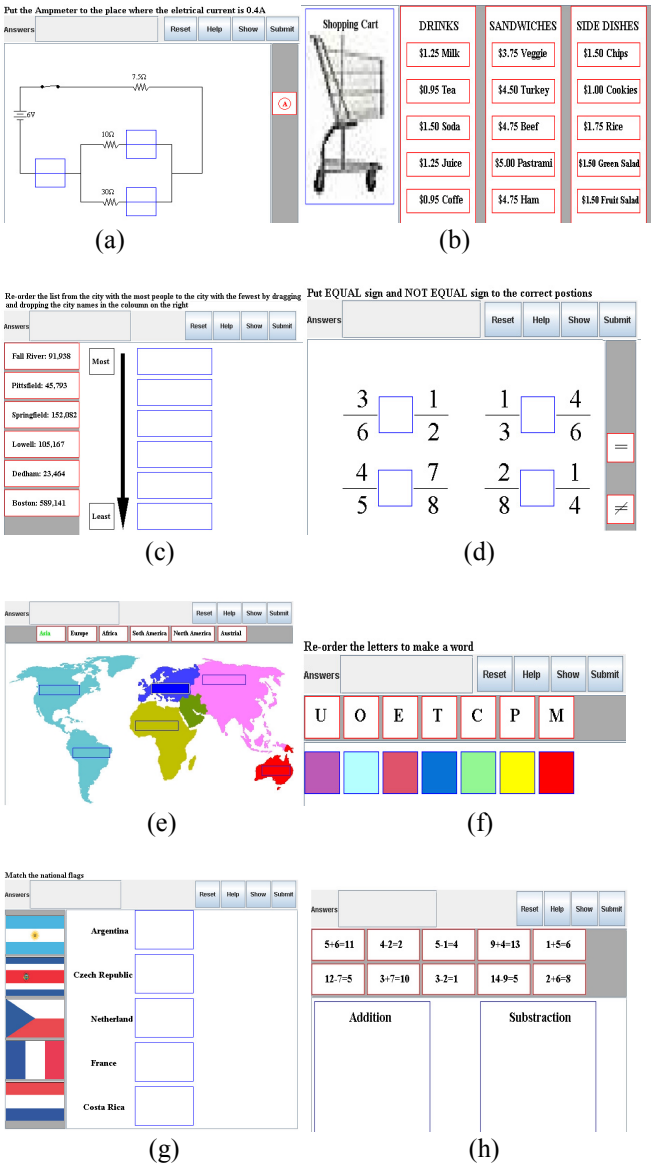

Figure 2: *The capability of displaying different multimedia contents in the same item shell reduces the number of item shells to be implemented.*

5. Many abstract concepts that cannot be captured using pen-and-paper format can easily be demonstrated using multimedia content, such as, three-dimensional display, audio and video. Therefore, computer-based multimedia items should be more diversified than traditional items.

A cost effective item shell should be portable, scalable and reusable so that not only the number of item shells is minimized, but also the number of questions is minimized. Our design strategies are described in the next section.

4. ITEM SHELL DESIGN STRATEGIES

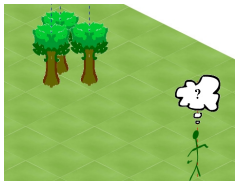

Figure 3: *An example of a Flash item: numerical values and operators will appear forming an algebraic expression as the stick man moves along. The student has to compute the result of the expression.*

Many learning objects available on the web are implemented using Flash. One reason is the runtime support from web browsers; another reason is its appealing graphics, animations and sound effects. However, Flash-based questions are not reusable; specific design and implementation is required for a question and it is inefficient if not impossible to implement a Flash item shell to satisfy requirements (3) and (4) above. Furthermore, although Flash can present 2D graphics and animations nicely, its 3D capability is not comparable with Java3D, at least in its current version. We chose Java to implement item shells because of it 2D and 3D capability, as well as the flexibility to satisfy requirements (3) and (4). To animate an object in Flash, multiple frames of 2D orientations have to be defined similar to the production of traditional cartoons. Although drawing a sequence of vector-based objects (Figure 3) is fast in Flash, creating an elaborate and photorealistic object for animation is time consuming. Besides, a student is not able to alter the sequence of predefined actions and navigate around the object interactively. By contrast, only a single Java3D object needs to be created which can be manipulated in any orientation. 3D mesh formats like OBJ and OFF can be reusable by different question items, and a large number of off-the-shelf Java3D objects are available in the public domain. Java runtime is also fully supported by web browsers, and more importantly, it is platform independent and thus portable. Effective 3D rendering can also be achieved in an OpenGL environment. However, it has to run locally as a stand-alone application, and cannot be used as an online network application for adaptive testing. Compared to earlier versions, the current Javascript version is more powerful in capturing graphics content including drag-and-drop components, as well as highlighting text. Although they can be used as learning items and take a

shorter time to render than applet items, the readable script at a client machine leads to security and copyright issues.

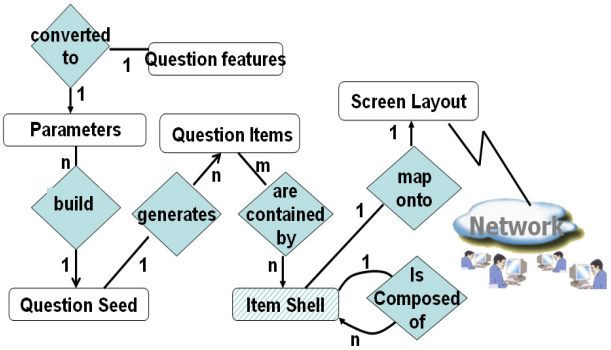

Figure 4: *A relational chart for the item shells.*

In order to achieve reusability and scalability, we propose using (Figure 4):

1. Screen layout and item shell mapping;
2. Item and item shell independence;
3. Composite item shells in addition to unit item shells;
4. Question seed association; and,
5. Parameter-based difficulty level estimation.

While question content, e.g., text description, is unique per question, there are meta-data (e.g., number of dragged objects in Figure 2(g) above) that do not need to change. By analyzing a question's content in the current curriculum, features independent of question content are extracted as parameters for designing multiple item shells. Groups of related parameters are then identified to form question seeds. Multiple question items can be generated, by assigning appropriate values to these parameters. Two fundamental parameters are the *question* and the *answer* parameter strings, which are composed of tokens. Question description and control values are extracted by parsing these strings. The question parameter controls how a question is displayed to a student, while the answer parameter guides the scoring. Although an item shell defines a group of parameters, not every parameter appears in a question. For example, one question may require 3D navigation while the other may need video clips to be displayed. We apply a standard screen layout to mask an item shell in order to maintain a uniform and consistent appearance. Masking means activating only screen components used by the current question. For example, the 3D animation component between the image and the audio panels is masked in Figure 5, because no animation is required in this question. Our design also supports composite questions composed of sub-questions. Figure 5 shows an example of a composite question. Note that a student can switch from one question to another by clicking on the tags displayed at the bottom. Composite questions are generated by item shells with multiple components. The advantage of using a composite shell is to let multiple questions share common descriptions, or to group related questions together into a logical unit.

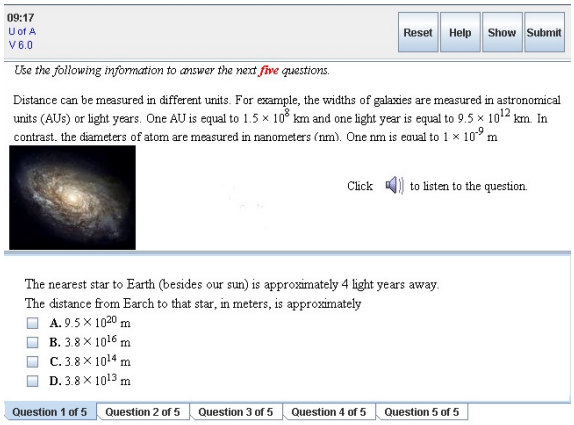

Figure 5: *A standardized and consistent screen layout is used by every question item. An item shell can be composite or standalone. In this example, a composite shell is shown where a student can switch between questions by clicking on the tags at the bottom.*

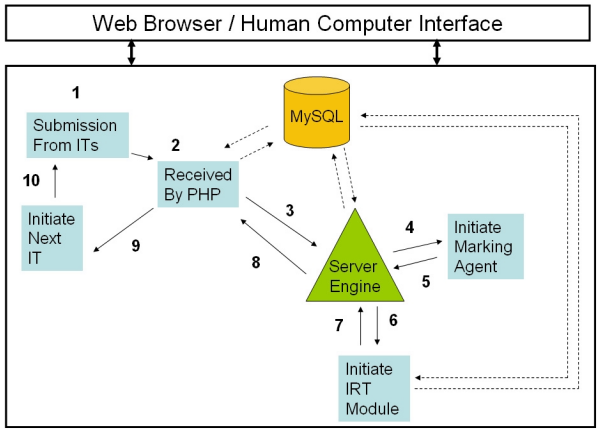

Figure 6: *Item shell environment.*

In the current implementation, PHP is used as the web language to communicate between a database (MySQL) and Java applet based item shells (Figure 6). Note that our item shells are portable and can be integrated with other web languages. Each student has his or her own logon session; allowing statistics, such as time stamp and IP address, to be recorded for every student. PHP is also used to communicate with the integrating engine at the server. After receiving the response string from a client applet, instead of passing the entire string to the server engine for further processing, the string is first divided into logical tokens by PHP and used to

update the database. An index is then created for the database record. Only the index is passed to the engine, which uses the index to retrieve a reduced data set that is required by the marking agent. This approach ensures that the server engine is not flooded with the large amount of data generated by multiple student sessions running concurrently. Since each item shell has its unique design and parameter composition, it is associated with its own marking agent. A student response is scored in the range [0, 1] where 0 means wrong and 1 means correct. Partial marks, between 0 and 1, can also be given.

An important process in our framework is to record the performance history and scoring statistics allowing abnormal learning patterns of students to be detected when they occur. This enables educators to take corrective actions on time. In order to assess individual student abilities, it is necessary to select a sequence of appropriate items dictated by a student's responses, allowing scores to converge to the student's ability level. In order to monitor a student's response curve and compute the student's ability based on the convergence of this curve, we adopt Item Response Theory (IRT) [4, 14], which is commonly used by most computer adaptive testing systems.

IRT is a family of mathematical models that describe how students' abilities relate to their item responses [14]. Figure 7 shows the s-shaped curve of an Item Response Function for students with different ability (θ) levels. The upper asymptote is at 100% and the lower asymptote is for students with very low ability. The x-axis represents a student's ability and the y-axis represents the probability of a correct response to test items. Note that the range [-3, 3] is normally used to assess a group of students at a particular grade level but it is possible that a student's ability may fall outside this range. The characteristic of an IRT curve is dictated by three parameters:

a_i – is the item discrimination parameter and defines the slope of the curve at its inflection point. When the value is 0, the IRT curve is flat showing no difference between students with high and low abilities. Items with low a_i values are eliminated from the item bank because they provide little information about individual student abilities. The value has to be quite large, say 2, before the curve becomes steep. The steeper the curve, the better is the item at discriminating students with slightly different abilities.

b_i – defines the item difficulty. A lower value will shift the curve to the left and a higher value will shift the curve to the right. When the value is less than 0, more than half of the students will get the correct answer (easy item). When the value is greater than 0, less than half of the students will get the correct answer (hard item). Note that if a student has ability (θ) = b_i, the probability of a correct response is 50%.

c_i – defines the probability of getting a correct response based on guessing. Changing its value will affect the lower asymptote. If the value is 0, students with limited ability have a low probability of getting a correct answer.

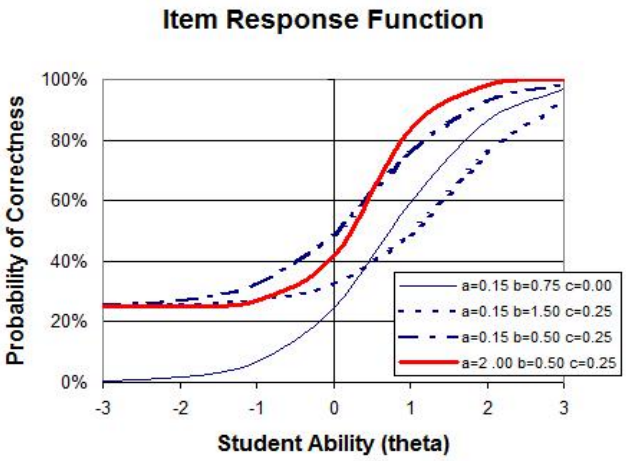

Figure 7: *An illustration of the Item Response Function using different parameter values.*

An IRT model can be associated with one, (1PL) two (2PL) or three (3PL) parameters. Only a_i and b_i are considered in 2PL. In 1PL, only b_i is considered. A general representation of the Item Response Function is given below:

$$P_i(\theta_j) = P(u_i = 1 \mid \theta_j, a_i, b_i, c_i) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta_j - b_i)}}$$

2PL is obtained by setting c_i = 0, and 1PL (Rasch model) is obtained by setting c_i = 0 and a_i = 1.

The 3PL model was first proposed in [15]. Given an item *I* and a student *j*, θ_j denotes the ability of student *j*, and $u_i (0 \le u_i \le 1)$ denotes the correctness of the student's response on item *i*. The 3PL model defines the probability of a correct answer to item *i* for student *j* in terms of $P_i(\theta_j)$. Traditional "True/False" and "Multiple Choice" (1 out of 4) test items have a probability of 50% and 25%, respectively, on guessing correctly. In contrast, the probability of guessing correctly in most of the multimedia items designed in our framework is extremely low, if not impossible (c_i approaches 0). For example, in a drag and drop item with *M* objects to place (Figures 2 & 12), the probability of guessing correctly is $1/M!$. Nevertheless, we adopt the 3PL

IRT model in our system in order to cover the possibility of correct guessing for some items.

In our item shell framework, we are interested in the following:

1. How to map the scores output from the marking agents to the IRT input?

2. How to define the terminating condition in order to detect the converged ability level?

Given a group of students at a particular grade, we start by assuming that every student has an average ability $\theta = 0$ in the range [-3, 3] and is given the first question with average difficulty $b_i = 0$. It is possible that a value outside the [-3, 3] range is generated resulting from the responses of a student. If such pattern persists, it is an indication that the ability level of the student is either below or above the current grade being considered. In our framework, difficulties are assigned in the range [0, 1]. Let β be the set of b_i defined in the IRT model and $\Re$ be the set of difficulty values in the range of [0, 1] used by our marking agents. To address Question 1 above, we establish a 1-to-1 mapping between $\Re$ and β. To address Question 2, we use the Item Information Function for item *i*:

$$I_i(\theta_j) = \frac{P_i'(\theta_j)^2}{P_i(\theta_j)(1 - P_i(\theta_j))}$$

and Standard Error Function in IRT:

$$SE(\theta_j) = \frac{1}{\sqrt{I_i(\theta_j)}}$$

where $P_i'(\theta_j)$ is the first derivative of $P_i(\theta_j)$.

The adaptive testing process terminates and the response curve is considered to have converged when $SE(\theta_j)$ is less than a predefined threshold. Regardless of the starting difficulty level given to a student, his or her ability can be assessed in a limited number of items as illustrated by the convergence rate of the curve shown in Figure 8. Readers interested in the mathematical analysis of IRT can refer to [4, 14, 15] for details. During the adaptive testing process, the server engine returns the newly estimated ability based on the current response of a student. Depending on the value of the updated ability, PHP retrieves the appropriate item and passes it on to the client applet for student testing, until $SE(\theta_j)$ reaches the predefined threshold.

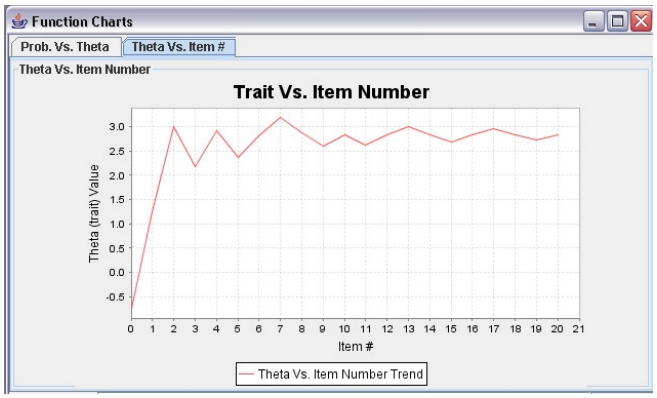

Figure 8: *A snapshot of our interface showing a student response based on IRT.*

5. TESTING AND SCORING STRATEGY

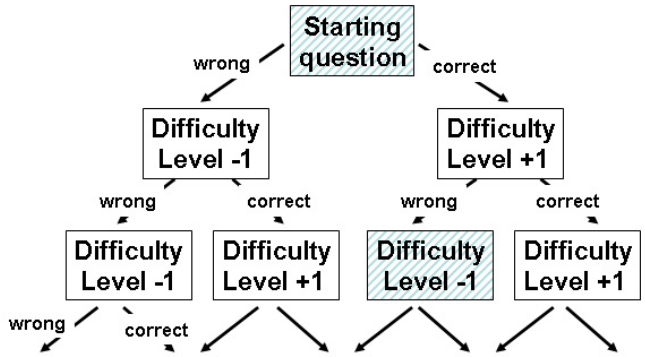

Figure 9: *Choosing the next question following a binary tree structure.*

Conventional design for classroom or group testing caters to an average difficulty level. This tends to make good students bored and weaker students drop out of courses [16, 17]. Adaptive testing on the other hand starts with a question of average difficulty, and then follows a path of questions depending on an individual student's response to the current question. One suggested implementation is to use a tree structure (Figure 9). The left child is less difficult than the parent and the right child is more difficult than the parent. In other words, when the current question is answered correctly, the next question is chosen from the right bin. If the current question is answered wrong, the left bin is chosen. This strategy allows the number of questions in a test to be cut down by as much as 50% while maintaining a level of difficulty in questions that makes tests remain interesting to all students. However, such implementation raises some issues:

1. Suppose there are P items of difficulty level d, which need to be distributed in multiple bins. For example, the shaded boxes in Figure 9 contain items of same difficulties. Duplicating the items in bins is not efficient in term of storage and update. Dividing P items into groups and putting into Q bins will result in P/Q items being placed in a bin. Note that sufficient number of items needs to be available in order not to repeat the same item frequently.

2. Putting different items into bins also means that students are not able to access the same item pool because each student has his or her individualized path traversing the tree structure.

3. The tree structure limits the increment and decrement of difficulty by one unit of difficulty. If the change of difficulty level is adaptively chosen, as shown in Figure 8, the next item can be many units away. In this case, the tree structure is not suitable.

These issues can be addressed by ordering the questions of N difficulties into N bins, which can be implemented using a doubly linked list structure (Figure 10). In this case there is no need to distribute questions of the same difficulty into multiple bins, and the ordered list facilitates searching items of the required difficulty.

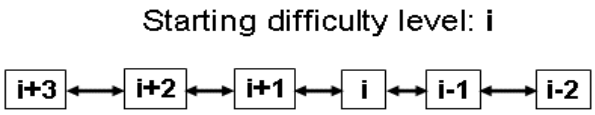

Figure 10: *Adaptive testing with a doubly-linked list structure.*

5.1 Response Curve

The True/False type of question is vulnerable to random guessing. If a student does not know the answer, (s)he can guess and the probability of guessing correctly is 50%. In this case, it is not possible to determine the student's true knowledge level because whichever initial difficulty level is assigned, the student's response curve stays in equilibrium. In normal circumstances, the response curve should follow one of the following three trends [2] (an example of a response curve is given in Figure 8):

- If the initial knowledge level is higher than that of the student, the curve should decrease and then converge at his or her level.

- If the initial level is lower than that of the student, the curve should increase and then converge.

- If the initial level matches that of the student, the curve should stay more or less horizontal.

The probability of guessing correctly in multiple choice questions with 4 choices is 25%. If the number of correct answers is used to assess a student's performance, random guessing can award a student 25% even if the student's ability is lower. Note that the point of inflection in Figure 7 divides the probability of getting or not getting a correct response into 50%. Assigning a zero value to c_i will affect the probability $P_i(\theta_j)$ given a student's ability, and thus will affect $I_i(\theta_j)$ and the next item to be selected. In order to assess a student's ability correctly, the 3PL IRT model taking the guessing parameter c_i into account is therefore more accurate. While the guessing probabilities for True/False and Multiple Choice items are obvious, it may not be possible to precisely compute the guessing parameter value for all the multimedia items due to their different screen presentations. It is therefore important to design the items in a way that minimizes the guessing probability as much as possible. At the same time, it is necessary to monitor the adaptive testing response curve to detect possible guessing patterns.

5.2 Parameters based Estimation of Difficulty Level for Math Item Types

Figure 11: *(Left) an interactive math question, and (right) a student answer.*

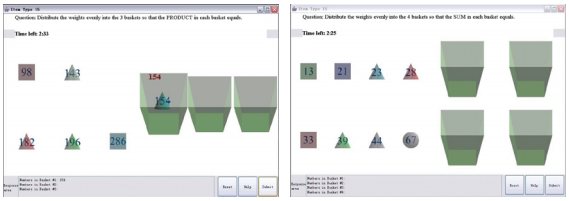

Figure 12: *Interface used in the evaluation experiment.*

Parameter based strategy is a more general approach for assigning initial difficulties to items. We use Math questions as examples to illustrate the concept. Figure 11 shows an item requiring a student to distribute the numbers into four bins so that the sum in each bin is the same. We define parameters n_{bkt} (number of bins) and n_{nbr} (number of objects to distribute) to control the generation of multiple items, as well as the difficulty levels of the questions

generated. For example, when solving the question "distribute the numbers so that the sum in each bin is equal" (Figure 11), the difficulty level of a question is defined by the function $f(n_{bkt}, n_{nbr})$. The difficulty level increases as n_{bkt} or n_{nbr} increases. Additional difficulty can be introduced by using decimal instead of integer numbers. We verified the feasibility of our approach by conducting evaluation experiments.

5.3 Evaluation of the parameter based strategy

Methodology

We extended the concept of IRT and used 2PL, coupled with measurement of average time taken to solve problems, to fit a linear regression model and examine the correlation between the difficulty levels generated by our strategy with the predefined difficulty levels. The calibration was done by seven students to rate the difficulty of each item based on the percentage of correct responses. 2PL was used, since it was almost impossible to guess the correct answer for the given question format; the value of parameter *c* was close to zero. Mathematical details will not be discussed here for brevity. However, we will describe the design of the evaluation experiment and discuss results. The user interface of the evaluation program is shown in Figure 12. The questions used for evaluating the automatic difficulty estimation algorithm are described in Table 1. The higher the question ID, the greater is the difficulty level. Questions 1 to 6 involve summation and Questions 7 through 12 involve multiplication.

Question ID	**1**	**2**	**3**	**4**	**5**	**6**
n_{bkt}	2	4	3	4	6	4
n_{nbr}	5	8	10	10	13	13
Max. time units	9	15	18	18	21	30

Question ID	**7**	**8**	**9**	**10**	**11**	**12**
n_{bkt}	2	2	3	3	4	4
n_{nbr}	7	6	6	9	12	12
Max. time units	15	15	18	21	30	33

Table 1: *The set of questions to test SUM (top), and to test PRODUCT (bottom), with increasing difficulties.*

The order in the table is the same as the one used in the evaluation. Maximum time (in units of 10 seconds) assigned to each question was based on a roughly predicted difficulty during the calibration process, *i.e.*, more difficult questions are allocated more time. Participants' familiarities with the questions were not taken into account during the assignment of maximum times. A participant's answer, time needed and mark for each question was recorded. The mark for an answer was not based on a simple correct or wrong criterion, and partial mark was awarded. For example, if a participant got the numbers in only one of baskets correct, whereas all together 4 baskets were present, (s)he could still get a mark of 0.25 (the full mark for an answer being 1.0).

Procedure

Seven participants were chosen, who were high-school students in Grade 10 to Grade 12 and understood basic arithmetic including factorization. Two sets of questions were given to the students:

1. Distribute the weights evenly into M baskets so that the SUM of the numbers in each of the basket is the same.

2. Distribute the weights evenly into M baskets so that the PRODUCT of the numbers in each of the basket is the same.

A procedure to solve a SUM question is:

(a) Add up the numbers and divide the sum by the number of baskets.

(b) Move the appropriate numbers into each basket based on the average computed in (a).

A procedure to solve a PRODUCT question is:

(a) Factorize each number into prime numbers.

(b) Place the appropriate numbers into different baskets so that each basket has the same count of different prime numbers.

Participants were allowed to use assisting tools such as a calculator, but not allowed to discuss with one another. The entire experiment follows a zero-feedback procedure:

1. A participant is introduced to the graphical user interface and the method for each type of question.

2. The participant answers two warm-up SUM questions in order to get used to the user interface and the method for solving the question.

3. The participant answers the set of SUM questions sequentially.

4. The participant answers two warm-up PRODUCT questions.

5. The participant answers the set of PRODUCT questions sequentially.

Results and Analysis

Each participant's ability was considered as his or her total mark scaled in the range between [-3, 3]. Depending on the estimated abilities, each question's difficulty parameter *b* is calculated using IRT. Based on the experimental data (not shown here), the linear regression equation for estimating the difficulty of the SUM questions is:

$$b = -6.44 + 0.47 n_{bkt} + 2.77(n_{nbr}/n_{bkt}) - 0.74\ ID$$

where *ID* varies between 1 and 6 (Table 1) depending on the calibrated difficulties. The correlation between the calibrated and experimental values was found to be R^2 = 0.95.

The linear regression equation for estimating the difficulty of the PRODUCT questions (*ID* between 7 to 12) is:

$$b = -14.74 + 3.52\ n_{bkt} + 2.77(n_{nbr}/n_{bkt})\ \ -1.08\ ID$$

with $R^2 = 0.99$. The high R^2 values (close to 1.0) indicate that the difficulty parameter *b* estimated by our algorithm has very high correlation with the *b* obtained from the calibrated values. Hence, the proposed parameter based strategy for estimating difficulty level is validated.

6. STUDENT FEEDBACK ON MULTIMEDIA ITEM TYPES

We have received positive feedback from K-12 students groups visiting our research centre regarding the appeal of multimedia item types to students. Extensive user studies with some students were conducted during August 2007. Some of the findings on four students are summarized in Table 2. The numbers in Table 2 are counts of multimedia items used in the evaluations. Grade 11 students evaluated 33 items in total, whereas the Grade 7 student evaluated 36 items. For example, out of 33 multimedia items, Student1 was satisfied with 31, neutral on 1, and dissatisfied with 1. Student2 took less (more) time to work with 14 (11) of the 33 items, compared to their corresponding pen-and-paper versions; and took about the same time with both forms of tests for 8 items.

Note that the four students in Table 2 had somewhat different backgrounds. Three were in Grade 11 and one was in Grade 7. Among the Grade 11 students, Students 2 and 3 had taken computer-programming courses while Student 1 did not have any programming knowledge. It can be seen that these students in general were both satisfied with the multimedia item types and also preferred computer based testing. However, there were some differences in the evaluations: (a) Students 2 and 3 had very similar evaluation and timing results since they were both from the same grade with good programming and user-interface knowledge, these skills may have given them an edge in performing the computer-based tests quite fast; (b) Student 1, though very interested in computer based multimedia item types, was relatively slower in working with the computer test interfaces, and in most cases performed the pen-and-paper tests faster; (c) Student 4, though satisfied and interested in the multimedia item types, was unable to record precise time data properly. This may be a result of the slight immaturity of a Grade 7 student compared to Grade 11 students. In future evaluations with junior students, it is necessary to find appropriate means of accurately recording the time taken on pen-and-paper tests without involving a costly monitoring process.

Feedback	**Student1 Grade 11**	**Student2 Grade 11**	**Student3 Grade 11**	**Student4 Grade 7**
Satisfaction				
Satisfied	31	29	25	24
Neutral	1	4	7	8
Dissatisfied	1	0	1	4
Preference				
Computer-based	30	21	23	26
Pen&paper	3	12	10	10
Time taken				
Less computer	9	14	14	Not
Less pen&paper	18	11	9	recorded
About the same	6	8	10	properly.

Table 2: *Summary of detailed evaluations by some students.*

7. CONCLUSIONS AND FUTURE WORK

In this paper the important prerequisites for effective item shell design for individualized education were discussed, along with implemented examples. We addressed the reusability, portability and scalability issues by introducing the concept of item seeds and parameters, which are used to control automatic generation of difficulty levels and scoring.

Experiments on using IRT for estimating difficulty level of multimedia math questions were described. Student feedback on some of our multimedia item types was also summarized.

In future work, we will look into the effectiveness of educational games and machine learning techniques in

studying learning behavior and enhancing performance using individualized education.

8. ACKNOWLEDGMENT

The implementation support of the CROME project team is gratefully acknowledged.

REFERENCES

[1] R. Allen, "The Web: Interactive and Multimedia Education," Computer Networks and ISDN Systems, vol. 30, pp. 1717-1727, 1998.

[2] I. Cheng and A. Basu, "Improving Multimedia Innovative Item Types for Computer Based Testing," IEEE *International Symposium on Multimedia,* pp. 557-566, 2006.

[3] R. Gonzalez, G. Cranitch and J. Jo, "Academic Directions of Multimedia Education," Comm. of the ACM, Vol. 43, No. 1, January 2000.

[4] R.K. Hambleton, H. Swaminathan and H.J. Rogers, "Fundamentals of Item Response theory," Sage Publications Inc., 1991.

[5] K. Ivers and A. Barron, "Multimedia Projects in Education: Designing, Producing and Assessing," 2nd Edition, Libraries Unlimited, 2002.

[6] E. ORhun, "Web-Based Educational Resources for Learning and Online Teaching in Higher Education: The MERLOT Project," Proc. of Int. Association of Technological University Liberies (IATUL), Vol. 13, 2003.

[7] C.G. Parshall, T. Davey and P.J. Pashley, "Innovative item types for computerized testing," in Computerized Adaptive Testing: Theory and Practice. W. van der Linden & C. Glas (Editors), pp. 129-148, 2000.

[8] O. Parlangeli, E. Marchigiani and S. Bagnara, "Multimedia Systems in Distance Education: Effects of Usability on Learning," Journal Interacting with Computers, Vol. 12, No. 1, pp. 37-49, 1999.

[9] S. Rabinowitz and T. Brandt, "Computer-based Assessment: Can it deliver on its promise?" an article from WestEd.org, 2001 website:http://www.wested.org/cs/we/view/rs/568

[10] J. Tuovinen, "Multimedia Distance Education Interactions," Education Media International, International Council for Education Media, vol. 37(1), pp16-24, 2000.

[11] T. Volery and D. Lord, "Critical Success Factors in Online Education," Int. Journal of Educational Management, vol. 14, No. 5, pp. 216-223, 2000.

[12] J. Yau and M. Joy, "Adaptive Learning and Testing with Learning Objects," International Conference on Computers in Education, 2004.

[13] A.L. Zenisky and S.G. Sireci, "Technological innovations in large-scale testing," Applied Measurement in Education, 15(4), 337-362, 2002.

[14] W. van der Linden and R. Hambleton, "Handbook of Modern Item Response Theory," London, Springer Verlag 1997.

[15] A. Birnbaum, "Some latent Trait Models and Their Use in Infering an Examinee's Ability," Statistical Theories of Mental Test Scores, 1968.

[16] S.D. Craig, A.G. Graesser, J. Sullins and B. Gholson, "Affect and learning: An exploratory look into the role of affect in learning with AutoTutor," Journal of Educational Media, vol. 29, no. 3, Oct. 2004.

[17] M. Gierl, Personal communication, Dept. of Educational Psychology, University of Alberta, 2006.