

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/282007839>

Massive Development of E-Testing Questions

Article in *International Journal of Emerging Technologies in Learning (IJET)* · September 2015

DOI: 10.3991/ijet.v10i4.4688

CITATIONS

6

READS

341

3 authors:



Sashko Ristov

University of Innsbruck

209 PUBLICATIONS 1,311 CITATIONS

[SEE PROFILE](#)



Marjan Gusev

Ss. Cyril and Methodius University in Skopje

487 PUBLICATIONS 2,063 CITATIONS

[SEE PROFILE](#)



Goce Armenski

Ss. Cyril and Methodius University in Skopje

50 PUBLICATIONS 192 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PeCoTradeinCloud [View project](#)



ECCGalert [View project](#)

Massive Development of E-Testing Questions

<http://dx.doi.org/10.3991/ijet.v10i4.4688>

Sasko Ristov, Marjan Gusev, Goce Armenski
Ss. Cyril and Methodius University, Skopje, Macedonia

Abstract—This paper presents several techniques and strategies for massive e-testing questions generation. These techniques can be used to develop a large set of questions and assessment content. A pilot project is presented in achieved speedup of questing development for single- and multiple-choice questions, as well as questions with two options (true/false).

Index Terms—eAssessment, eTesting system, question database, knowledge evaluation.

I. INTRODUCTION

e-Assessment allows teachers to determine the students' learning outcome of some part or the whole course. Many systems specify a possible application of e-Assessment in the learning process, even in various e-Learning systems [1]. New cloud-based architectures offer even scalable and elastic e-Assessment systems [2], cost effective compared to on-premise [3], which can support even MOOCs.

Developing and automatically generation of questions are important functions of modern e-Learning, which includes the existence of intelligent tutoring systems [4] and e-Assessment systems. Automatically generating high-quality questions is the long-term goal of the emerging Question Generation (QG) research community [5]. Graesser et al. [6] discuss limitations of human question asking that motivates the automated question asking generation. Motivation to build tools for automated question asking has raised a lot of attention, in order to create a relevant question database.

Teachers have a huge problem to build a knowledge base that can be used to assess the students' learning outcomes properly. This is a very difficult and time consuming task, which requests a huge amount of resources [7]. Powers has reported that 10 hours per week is a minimal requirement to create a small set of questions [8]. This leads towards developing new strategies to build and update regularly the knowledge database. In addition to this, the knowledge database must be updated due to continual development in the engineering courses.

There are other approaches to generate a bulk set of questions, such as natural language processing, machine learning, semantic and other approaches. For example, Jovanov and Gusev's approach is based on ontology building of a big knowledge database, which achieves the question generation speedup by 10 times [9].

Gusev et al. [10] proposed a novel technology for e-Testing question development. The approach uses objects, their hierarchical organization and associations with certain properties. To realize this approach, we have developed several strategies to address different question type definitions. This paper presents an intelligent tool, which can construct questions based on establishing

relations among objects and associations with their properties. The generation of questions using several strategies is evaluated in order to prove the huge speedup and the greater number of generated questions. The additional benefit is the similarity in the generated questions, which reduces the possibility for cheating and guessing.

Section II discusses related work about e-Testing question types and question generation for e-Testing systems. Overview of e-Testing technologies is given in Section III. Section IV describes the question generation process and specifies various e-Testing types for bulk upload of questions. Strategies for question generation are elaborated in Section V. The evaluation of some techniques and strategies is presented in Section VI, which proves the successfulness of our approach. Section VII shares our experiences and discusses the strategies and technologies. Finally Section VIII provides conclusions and recommendations for further work.

II. RELATED WORK

This section covers two areas about categorization of the question types and question generation approaches.

A. e-Testing question types

Scalise and Gilford [11] construct intermediate constraint taxonomy for e-Testing questions and tasks by identifying 28 different question types. Their methodology is based on definition of item/task response format, having multiple choice questions as a fully selected category and presentation/portfolio as fully constructed response format. The second ordering scale concerns the complexity of provided answers. In the example of multiple choice questions, true/false types are less complex, and more complex are alternate choice (with only one correct answer for more than 2 options) and standard multiple choice questions.

The more construction is needed in the response format in the intermediate constraint item types them more they are incompatible for automated e-Testing, where the computer can automatically evaluate and grade the answer. Examples that might be used in the e-Testing systems include multiple selection and identification, reordering and rearrangement, and some forms of substitution/correction and completion question types. More complex forms of the last two categories and construction and presentation / portfolio question types can be organized as a kind of essay question and are not convenient for automated testing with current development of machine learning tools and understanding of natural languages.

Possible ways to reduce the effect of guessing in the overall assessment, such as constraints in developing the questions, are at least the following: at least one correct

answer and at most 50% correct answer candidates, introducing negative grading scheme, extending the domain of option answers, etc [12]. For example, let there be k correct answers in a multiple choice question with n possible answer options. It means that if there is no negative grading scale, the guessing method can have k/n probability of a correct answer. Their method in setting constraints in developing questions is based on setting at least one correct and one wrong answer options. The negative scoring scheme they implement evaluates each answer option, whether is chosen by the student in such a way that if it is correct it scores the appropriate portion of the possible grading, and if it is wrong it scores half of the negative points.

Extension of the domain of possible answers is another issue, that has been also analyzed by other researchers. Parshall et al. propose innovative question types that improve resistance to guessing [11]. To reduce the potential effect of guessing as an assessment measure they suggest several new innovative types that increase the domain of possible answer options. Examples include selection of a given topic in a sentence, or by a use of constructed response item (including a formula that calculates a mathematical expression instead of a list of preselected numerical values. Inclusion of sound and images are another suggestion to expand the content areas.

Gusev et al. [13] have implemented an e-Testing system with interactive images that include several features about interactive images, such as zooming, navigation, clicking, marking up a region, etc. This enables the pixels in the whole image to be possible candidates for answer and eliminates the guessing as a measure in the assessment evaluation.

Lehnert [14] has defined 16 question categories: verification (yes/no), disjunctive (multiple-choice single answer), concept completion (who, what, when, where), example, feature specification (properties of X), quantification (how much, how many), definition (what does X mean), comparison (how is X similar to Y), interpretation (what is the significance of X), causal antecedent (why/how did X occur), causal consequence (what next, what if), good orientation (why did an agent do X), instrumental/procedural (how did an agent do X), enablement (what enabled X), expectation (why didn't X occur)? And judgement (what do you think of X). Some of the questions can appear in two categories. Most of the e-Testing questions that have automated evaluation as a part of the e-Assessment build only few of these categories, such as verification, disjunctive, concept completion, feature specification, and quantification. The other question types belong to questions that need more sophisticated tools for evaluation of provided essay answers.

B. Question generation for e-Testing systems

Several research papers report various question generation techniques.

Most of the published research concerns natural language techniques to realize completion or substitution/correction question types.

Stahl [15] proposes a model with three levels effective vocabulary instruction: 1) association processing (as a kind of matching question type) where the word meaning is associated with other familiar concepts, 2)

comprehension processing (as a kind of completion question type), by placing a word in a particular context and 3) generation processing (as a kind of completion question type), that places the word in a novel context with a deeper multidimensional understanding of its meaning. Existence of vocabulary dictionaries are the essence of these programs.

Brown et al. [16] use WordNet to generate 6 types of vocabulary questions: definition, synonym, antonym, hypernym, hyponym, and close questions. Their implementation uses selection of words that belong to a certain set of synonyms, antonyms, hypernyms or hyponyms found in WordNet. It can be classified as matching structure in the question answer taxonomy.

Automatic question generation was analyzed by Wolfe on examples of a given text. This is analyzed in scope of the natural language processing (NLP), such as those simulated psychotherapist programs, including the well-known Eliza, film Her etc. Most of these approaches use pattern matching keyed to certain words and aim not to evaluate the student's answers to given questions, but to follow a certain procedure towards different goals, such as psychotherapy, diagnosis etc. Another example, includes IBM's computer system Watson, that won the USA's national Jeopardy quiz and recently used for diagnosis of cancer in medical practice.

Kalady et al. [17] use phrase-specific question generation, such as yes/no questions over a sentence, and various "wh" (what, where, who, how) questions from the subject, object, adverbials and prepositional phrases in the sentences. They have introduced the factoid question and definitional question generation strategies.

Question generation was analyzed by Rus et al. [18] as shared task evaluation challenge (STEC) in two directions: question generation from paragraphs and from sentences. They have developed an evaluation criteria based on relevance, question type, syntactic correctness and fluency; ambiguity and variety.

Another approach is based on community based question answering (QA) services, implemented by several search engines and similar sites, which analyzes the answer statistically on a huge sample of texts [19]. The approach is based on multidocument summarization, of answers according to the question type. However, these approaches for question answering are beyond the techniques described in this paper with goal to create a huge set of questions in e-Testing system.

Question generation can also be analyzed by a semantic approach. Yao et al. [20] suggest a transformation from natural language understanding to natural language generation via symbolic text representation. To realize this transformation they use sentence simplification, and question ranking in addition to the transformation process. Their method is based on term extraction, construction of a feature structure chart in an XML format, parsing, decomposition, generation and output selection.

III. OVERVIEW OF E-TESTING TECHNOLOGIES

This section provides an overview of the features and architecture of an e-testing system.

A. Features of an e-testing system

Classical Multiple choice e-Testing technology is a standardized computer-based testing exploited by a lot of

solution providers. Here, we understand that classical e-Testing approach is far beyond the conventional paper based testing, where a static test is generated with predefined answer options, so the student has to select one or more candidates as answer options. In this context, a classical e-Testing technology uses graphical user interface and presentation module that realize at least the following features:

- The question database is organized in several knowledge items, that is, learning objectives (LO).
- Each LO consists of several questions.
- Each test is defined by a strategy that defines the LOs and number of questions from each LO.
- A test is generated by randomly choosing a question from the LOs.
- The answer options are presented in randomized order.
- All provided answers are sent for automatic evaluation.
- As soon the test is finished, the results appear on the screen, so the students learns what are the correct answers and what is the obtained score.
- Negative scoring scheme can be defined to prevent from recognition, guessing and other related cheating methods.

B. Architecture and organization of an e-testing system

Most of these features can be realized by several modules (management, assessment and reporting) [2]. The presented system is realized as a scalable cloud solution capable to cope with increased processing demand.

The management module deals with administration issues. It contains the admin agent responsible for authentication, authorization and accounting, communication with the learning management system to exchange relevant data, and other systems. The infrastructure agent controls the cloud resources and activates necessary virtual machines or deactivates them according to specific eco parameters for efficient work. The final goal of this infrastructure agent is to enable elasticity in the system. Assessment generation agent is responsible for entry of questions in the question database, while the assessment commit agent for definition of the test strategies, activation and termination of tests and evaluation of results.

The assessment module is responsible for generation of the test, its delivery via the graphical user interface, execution of the answering session, collecting the results, evaluation of the test, and presentation of correct answer after test termination etc.

These activities are executed by the student module, which corresponds with the infrastructure agent and the reporting agent. Besides textual presentation, the graphical user interface can present figures, sounds, or related multimedia that can be presented by a modern web browser.

The reporting module creates all necessary reports and statistics. The teacher gets all relevant success parameters for executed tests, and the student gets only those personalize results and statistical indicators.

IV. QUESTION GENERATION FOR E-TESTING

Questions differ in the nature, when enhanced media is used. A very old proverb says that an image is like 1000

words. In this case the image really can present a lot of details and offer a huge number of answer options, in contrast to the classical multiple choice questions. The specifics of the both approaches are analyzed next.

A. Specifics of various e-Testing types

The goal of question generation in multiple choice questions is to ask a question and offer several answer options. There are certain limitations for offering correct answer options, such as at least one correct answer, at most 50% correct answers among all answer options, negative grading scale, which usually evaluates wrong answer by half of possible points that can be scored for correct answer. These constraints prevent cheating realized by guessing, or selecting all answer options. Instead, it increases the possibility to evaluate the provided answer as a measure of knowledge.

The question generation is realized in a traditional way, for example, by asking about which properties belong to a certain object and selecting from a given set of options. A lot of implementation variants may exist, such as realization of yes/no questions, single-answer question types, or multiple correct options.

Matching questions aim at matching a set of possible answer options to a set of questions parts in the whole question. They can be considered as a variant of a multiple choice question, since for each question part, a set of multiple choice options is offered.

In this category we can also analyze the questions that are based on completion, where the correct answer has to be provided by (typing) specifying a number or a string.

All questions in the process of generation has to identify the LO it belongs to. Gusev and Armenski [21] specify an organization that uses course, lessons, section, subsection, chapter, subchapter and LO in a hierarchical tree organization, defined by a recursive parent child relation, meaning that the structure can go deeper in the hierarchy.

A LO is the basic topic to be analyzed and it contains of several questions, each with determined complexity. Note that, a LO can be shared and used by different sections or chapters, enabled by realization of a corresponding structure.

The complexity is usually measured in three levels, easy with 1 point, medium with 2 points, hard with 4 points, although the teacher has freedom to set new complexity weights, for example, very hard question with 6 or even 8 points. Time allowed for question answering is mostly associated to the complexity, the number of points is usually allowed time in minutes, although the teacher has freedom to define a new value.

B. E-Testing question generation process

Fig. 1 explains the question generation process realized by setting the question text and setting answer options. This is followed by specifying the correct answer options and specification of question attributes. The question attributes are specifying:

- allowed time for providing answer in seconds (minutes),
- LO it belongs,
- grading scheme (selection among various negative marking for provision of wrong answers),

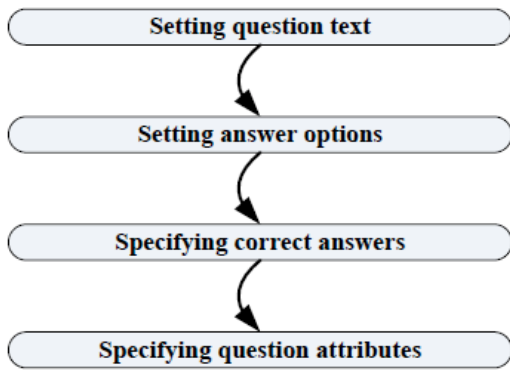


Figure 1. Question generation of classical multiple choice questions

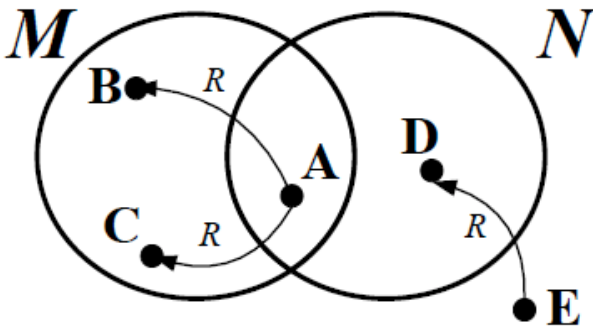


Figure 2. Objects and their properties as explanation of LO

- question weight complexity in score (points),
- other related attributes, such as if it allowed for self testing, exam, on-line tutoring and learning, etc.

The main idea is to enable a tool that will be used for various text processing software applications and after editing, the question upload to be finalized by a click on the application, instead of having complex graphical user interface where all question generation parts are to be entered one by one in separate window objects and fields, mixing a lot of mouse movements, clicks and keyboard typing.

For example, the teacher will prepare a word document that uses predefined styles, save the document and then upload it as a whole. The word processing is much easier and faster than entering the required structured information, and in the same time it allows structuring of different textual parts, so the application can easily understand the question attributes, and store it in corresponding database fields.

Once the questions are uploaded in the database, the teacher can validate its correctness and set additional attributes, not specified during the typing process.

V. QUESTION GENERATION STRATEGIES FOR E-TESTING

The strategies will be explained by the objects and their properties, as presented in Fig. 2. Each object presents an identification of a knowledge item or related term.

Properties are presented as relations from objects to a certain property. Objects can be organized in various sets, which can have a common intersection. In this sense, set is a presenting the relation "is element of a set". Fig. 2 contains two sets M and N, defined by a relation "is element of a set" and denoted as relations P and T

correspondingly. These relations are not drawn on the Figure for better clarity, but the sets are presented with their boundaries. Note that the relation P establishes relations between the pairs (A,M), (B,M) and (C,M), while the relation T between pairs (A,N) and (D,N). Note that the object E does not belong to any set.

Objects also can be organized in a hierarchical structure, which is another relation among objects, like the relation "parent to child" or "child to parent". The example in Fig. 2 presents the relation R as the "parent to child" relation between pairs (A,B), (A,C) and (E,D).

The following strategies for question generation of classical multiple choice questions are defined among objects and their properties:

- Strategy O2T (Originator to Target)
- Strategy T2O (Target to Originator)
- Strategy Rel (Setting Relations)
- Strategy CA (Complementary Asking)

One might use two strategies to generate a set of questions, especially the use of the complementary asking strategy is recommended as a combination of all other strategies.

The O2T strategy addresses the "originator to target" relation among objects. Both relations "parent to child" (relation R) and "is element of" (relations P and T) can be illustrated by the given example in Fig. 2. Possible questions within the presented example are the specified in Table I.

The T2O strategy is reverse of the O2T strategy. It addresses the relation "target obtained by the originator" relation among object. The relation "child to parent" is reverse of the "parent to child" relation and the relation "contains the element" is reverse of the "is element of" relation. Both can be illustrated by this strategy in the example presented in Fig. 2. Some illustrations of possible questions are presented in Table II.

The Rel Strategy aims to set relations among various question items. This strategy actually mixes several facts with possible answers into one combined question. Also in this case we will use the example presented in Fig. 2 and illustrate possible questions in Table III.

TABLE I.
 QUESTION GENERATION USING O2T STRATEGY

question text	answer options	Correct
Which objects are related by the object A for the relation P?	A, B, C, D, E, M, N	M
Which objects are related by the object A for the relation R?	A, B, C, D, E, M, N	B, C
Which objects are related by the object A for the relation T?	A, B, C, D, E, M, N	N

TABLE II.
 QUESTION GENERATION USING T2O STRATEGY

question text	answer options	Correct
Which objects with relation P target the object M?	A, B, C, D, E, M, N	A, B, C
Which objects with relation R target the object B?	A, B, C, D, E, M, N	A
Which objects with relation T target the object N?	A, B, C, D, E, M, N	A, D

TABLE III.
 QUESTION GENERATION USING REL STRATEGY

question text	answer options	Correct
Which properties (relations) originate from the object A?	P, R, T	P, R, T
Which properties (relations) originate from the object B?	P, R, T	P
Which properties (relations) originate from the object C?	P, R, T	P
Which properties (relations) originate from the object D?	P, R, T	T
Which properties (relations) originate from the object E?	P, R, T	R
Which objects originate the property (relation) P?	A, B, C, D, E, M, N	A, B, C
Which objects originate the property (relation) R?	A, B, C, D, E	A, E
Which objects originate the property (relation) T?	A, B, C, D, E	A, D

TABLE IV.
 QUESTION GENERATION USING COMPLEMENTARY ASKING STRATEGY IN COMBINATION WITH O2T AND T2O

question text	answer options	Correct
Which objects are not related by object A for the relation P?	B, C, D, E, M, N	B, C, D, E, N
Which objects are not related by object A for the relation R?	B, C, D, E, M, N	D, E, M, N
Which objects are not related by object A for the relation T?	B, C, D, E, M, N	B, C, D, E, M
Which objects with relation P do not target the object M?	A, B, C, D, E, N	D, E, N
Which objects with relation R do not target the object B?	A, C, D, E, M, N	C, D, E, M, N
Which objects with relation T do not target the object N?	A, B, C, D, E, M	A, B, C, M

The strategy for complementary asking is usually used in combination with the other strategies. Table IV illustrates usage of this strategy in combination with the O2T and T2O strategies for the example presented in Fig. 2.

VI. EVALUATION

This section presents the pilot system for massive e-testing questions development. It evaluates the generation of questions by using the O2T and T2O strategies. Additionally, negative version of questions is generated, which additionally increases (doubles) the knowledge database.

A. The pilot system

The pilot project is an extension of the existing e-assessment and online learning engine. The existing database is upgraded to support the relations between the objects and the properties. The teacher first must define several sets (classes) within a LO and several objects within a single set.

After definition of the relations, the teacher can proceed to question generation, which is shown in Fig. 3. The teacher selects the course (Computer Networking), learning objective where the questions will be generated (Introduction and layers), class of the question (set), type of the question (strategy), and language (Macedonian / English). The last two text boxes are intended to define the total number of possible answers (for example, three

choices) and the number of correct answers of them (for example, two correct).

Here is an example of generated questions. We defined 7 sets for each layer of OSI model, and three objects (hub, switch and router), which are on the lowest layers, correspondingly (the first, the second and the third layer). The question templates with possible answers are:

- “Which layer object <object> belongs?” with all sets as possible answers. Note that these generated questions are the single choice, but the system can generate multiple choice questions, as well.
- “Which object belongs to layer <layer>?” with all objects (network devices) as possible answers. Note that only the sets that have objects within will be included in the generation (the lowest three layers in this case).
- “Is the object <object> from the layer <layer>?” with the possible answers *True* and *False*.

By starting the generation of the questions, the questions of all three types will be generated. Even more, the negative version of the questions can be generated, by generating the negative relations.

B. The question generation evaluation

Two types of question generation will be evaluated: the traditional one by one question with its choices, and with the engine for massive question development. The evaluation is based on the comparison of the time spent for the preparation and generation, and the number of generated questions for that time, by using the same inputs.

Table V presents the average time for actions that are required for classical question generation process, that is, generating the question text and the possible choices with selecting the correct answers. Here, we must note that other two activities (specifying of the correct answers and specifying the question attributes) are included, that is, generating the question text includes the specifying the question attributes, while the creating one choice includes the specifying of the correct answers. For example, for question with 3 choices, a teacher will need average of $(30+3*30) = 120$ seconds. For three questions of this type, a teacher will spend 360 seconds, that is, the total time for question generation is linear, or the average time for a question is constant.

Генерирање прашања

Предмет: Податоци и компјутерски комуникации ▾

Област: Вовед и словенитост ▾

Класа: Вовед во компјутерски мрежи и Интернет ▾

Тип на прашање: Генерирање прашања од објекти ▾

Јазик: Македонски ▾

Број на одговори: 3

Број на точни одговори: 2

Генерирај

Figure 3. Form for massive generation of questions

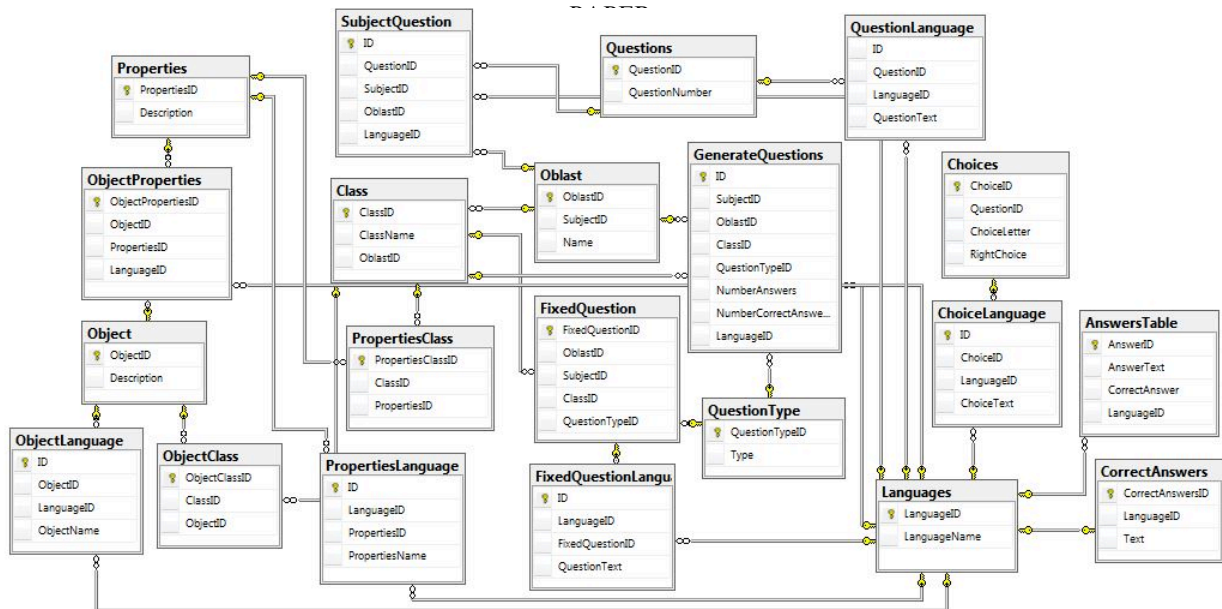


Figure 2. Question generation process including interactive images

TABLE V.
AVERAGE TIME FOR CLASICAL QUESTION GENERATION

Activity	AVG Time
Generating the question text	30 sec
Generating one choice text	30 sec

TABLE VI.
AVERAGE TIME FOR MASSIVE QUESTION GENERATION

Activity	AVG Time
Adding an object	30 sec
Adding a property	30 sec
Question type and relation	30 sec

Table VI presents the average time for actions that are required for massive question generation process. Here, a teacher spends more time for creating the relation of a given class of a given LO, rather than writing the question text and choices. For example, for three objects and three properties, the total average time will be $3 \times 30 + 3 \times 30 = 180$ seconds, and together with $3 \times 30 = 90$ seconds for definition of questions of the three types, a teacher will need 270 seconds. Additionally, the automatic question generation is ignored because it is less than a second. Similarly as the activities of the classical question generation process, here the activities for specifying of the correct answers and specifying the question attributes are assumed in the last activity (specifying the question type and relation).

The automatic process will generate $9 + 3 + 3 = 15$ questions for 270 seconds, compared to the three questions for 360 seconds. We can conclude that more questions are generated in less time.

Let us analyze now for generating questions which can be modeled as more objects in a set. Database is organized as shown in Fig. 4, which allows generating the combination of questions, objects and properties. For example, 10 questions with 10 properties. The total average time for the traditional generation will be 1200 seconds. But, the massive generation will be reduced much more, that is, by 300 seconds for objects and properties, 90 seconds for questions, or total average time of 690 seconds plus 10 seconds for generation. As a conclusion, the standard generation will provide 10

questions for 1200 seconds, while the massive generation will provide $100 + 10 + 10 = 120$ questions for 700 seconds.

Similarly, generating 30 questions with three answers will require 60 minutes (3600 seconds) for traditional generation, while generating by 30 objects and properties requires 32 minutes (1920 seconds). The former will generate a total of 30 questions, while the latter will generate a total of $900 + 30 + 30 = 960$ questions.

The better analysis offers Fig. 5, which presents the average time per question generation as a function of the number of generated questions, for both generations. We observe that the average time per question generation is decreasing when the number of generated questions is increased. That is, generating more questions will increase the number of generated questions with the same efforts and it will decrease the average time per question generation.

Additionally, generating questions with more objects and properties will increase the number of generated questions and decrease the average time per question generation even more.

VII. DISCUSSION

In this paper we have specified several question generation strategies that can be efficiently used for e-Testing capable for automatic generation by appropriate tools. The main goal is to specify a technique and a strategy that can generate a bulk set of questions.

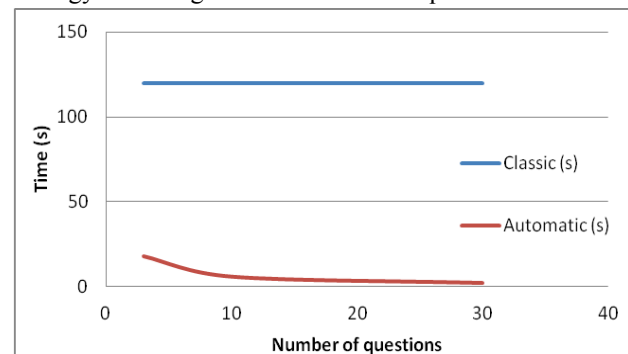


Figure 5. Comparison of average time per question for traditional and massive question generation

A. Application domain

Let us analyze the LO as a single organizational unit in the structure of the course or lesson, or in even deeper hierarchical levels, such as sections and chapters. Each LO is supposed to present a given definition or fact. The idea is to generate a set of questions with equal complexity in the corresponding LO. The strategies are giving hints how to generate questions that will fit in the same knowledge item or similar items, or how to generate several questions with the same complexity.

The presentation we are using looks like building an ontology, although it is not presented by a semantic approach. More details on semantic approach and use of ontologies for question generation are specified by Jovanov et al. [9].

When analyzing the strategies for generation of multiple choice questions, we can conclude that the Rel strategy generates questions with more complexity than the strategies O2T and T2O. It uses definition of several properties among objects (several facts) and combines them making a possibility for connections of related terms and LOs. Therefore, the student is asked to understand and make difference among similar objects and properties.

B. Experience with real implementations

We have implemented the classic e-Testing system [21], which has been in use from 1999. In the meantime we have faced a lot of problems with bulk upload and question generation. The strategies discussed are result of experience realized for a lot of courses and generation of more than 50.000 questions for more than 100 available courses. For example, there are more than 3.000 questions for the Computer Architecture and Organization course and more than 2.500 questions for the Computer Networks course.

According to our experience, the process starts with a given LO and then one group of questions is associated by using the O2T strategy and another with the P2O strategy. All these questions are evaluated with the same weight. Our experience shows that we can generate several question sets (groups) associated with the same LO. We found that the questions generated with the Rel and CA strategies are more difficult for the students, since they have to think and associate (apply the knowledge) and find the answer, instead of just following the memorization principle of a parent object and its children (memorization is connections between first words of questions and possible answers).

C. Additional benefits

Generating massive number of questions in the knowledge database provides additional benefits, apart of the huge knowledge database that is generated.

The system generates huge number of the same / similar questions, with similar answers. These similar questions and choices are very hard to be guessed without being learned. Also, the possibility for cheating (seeing the correct answer from the colleague monitor or telling the colleague the answer) and remembering is reduced. Finally, the students will be discouraged to take the exams offhand or learn the question answers by rote.

The pilot system allows generating the same questions for other languages with minimal additional efforts, that is, by defining only the terms of objects, relations and

fixed part of the questions on other languages, without creating the relations between objects again for each language. Additionally, a teacher can define many questions with the same objects and properties for a given class and LO, with different semantic in order to reduce the memorizing of the questions and correct answers.

Another benefit is the quality of the questions. That is, our experience showed that about 5% of the generated questions had some errors in the question or choices text, or the wrong choice was selected as a correct answer, or correct answer was not selected as a correct one. Also, by copying some parts of the text for similar questions or choices, a lot of questions had two same answers. These errors frustrate the students during and after the exam. It also spent additional teacher's time for reevaluating the exams. This pilot system on the other side, create automatically the questions with combinatorial logic, which cannot miss or duplicate some part of the questions or choices. Even if the teacher misses something in defining the objects and properties, he or she can recreate all questions of a given class by a single click and then reevaluate the exam.

VIII. CONCLUSION AND FUTURE WORK

In this paper we present several technologies and strategies that lighten and speed up the question generation process for e-testing system. They can be efficiently used in question generation of a set of questions or bulk upload of questions to the database that was our research target in this paper.

Our experience in realization of these kinds of systems shows that the systematic approach using these strategies reduces the overall question development time and results in a good and sufficient quality of questions used in the e-Testing system. Even more, many additional benefits are obtained with the realization of the pilot system (without or with minimal additional efforts):

- Reducing the cheating, memorizing and guessing
- Discouraging the students for exam offhand or learning by rote
- Generating questions in several languages
- Improving the questions' quality
- Reducing the errors in questions and answers
- Improving the reevaluation of the exams.

Comparison of different strategies within the same system is set to be our next experimental task. We plan to evaluate results of applying tests with questions developed with different strategies, and evaluate their complexity by analyzing the student answers in several disciplines.

We will conduct a survey among students and teachers to determine the satisfaction of using this new pilot system.

REFERENCES

- [1] Y. B. Gonzalez and M. L. Nistal, "Assessment functionalities on elearning platforms: An analysis," in *Tecnologias Aplicadas a la Ensenanza de la Electronica (Technologies Applied to Electronics Teaching) (TAEE)*, 2014 XI, June 2014, pp. 1–8.
- [2] S. Ristov, M. Gusev, G. Armenski, K. Bozinoski, and G. Velkoski, "Architecture and organization of e-assessment cloud solution," in *IEEE Global Engineering Education Conference (EDUCON)*, 2013 IEEE. Germany: 2013, pp. 736–743, best paper award. <http://dx.doi.org/10.1109/EduCon.2013.6530189>

PAPER
MASSIVE DEVELOPMENT OF E-TESTING QUESTIONS

- [3] A. Caminero, A. Robles-Gomez, S. Ros, R. Hernandez, R. Pastor, N. Oliva, and M. Castro, "Harnessing clouds for e-learning: New directions followed by UNED," in *Global Engineering Education Conference (EDUCON), 2011 IEEE*, April 2011, pp. 412–416. <http://dx.doi.org/10.1109/educon.2011.5773169>
- [4] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney, "Autotutor: An intelligent tutoring system with mixed-initiative dialogue," *Education, IEEE Transactions on*, vol. 48, no. 4, pp. 612–618, 2005. <http://dx.doi.org/10.1109/TE.2005.856149>
- [5] Question generation main website. [Online]. Available: <http://www.questiongeneration.org/>
- [6] A. Graesser, J. Otero, A. Corbett, D. Flickinger, A. Joshi, and L. Vanderwende, "Guidelines for question generation shared task evaluation campaigns," in *The Question Generation Shared Task & Evaluation Challenge Workshop Report, University of Memphis, 2008*. [Online]. Available: <http://www.questiongeneration.org/TheQuestionGenerationSharedTaskAndEvaluationChallenge.pdf>
- [7] D. Pain and J. L. Heron, "Webct and online assessment: The best thing since soap?" *Educational Technology & Society*, vol. 6, no. 2, pp. 62–71, April 2003. [Online]. Available: http://www.ifets.info/journals/6_2/ets-6-2.pdf
- [8] E. Powers, "In support of the e-test," Inside Higher ED, online news, Aug 2007. [Online]. Available: http://www.insidehighered.com/news/2007/08/29/e_test
- [9] M. Jovanov and M. Gusev, "Performance evaluation of a new approach for automatic question production," in *ICT Innovations 2009*, D. Davcev and J. Gomez, Eds. Springer Berlin Heidelberg, 2010, pp. 275–283.
- [10] M. Gusev, S. Ristov, G. Armenski, "E-Testing question development technologies and strategies," *Global Engineering Education Conference (EDUCON), 2015 IEEE*, March 2015, pp. 552–560. <http://dx.doi.org/10.1109/educon.2015.7096024>
- [11] K. Scalise and B. Gifford, "Computer-based assessment in e-learning: A framework for constructing "intermediate constraint" questions and tasks for technology platforms," *The Journal of Technology, Learning and Assessment*, vol. 4, no. 6, 2006.
- [12] S. Ristov, M. Gusev, G. Armenski, and G. Velkoski, "Scalable and elastic e-assessment cloud solution," in *Global Engineering Education Conference (EDUCON)*, Turkey, 2014, pp. 762–769. <http://dx.doi.org/10.1109/educon.2014.6826180>
- [13] M. Gusev, S. Ristov, G. Armenski, P. Gushev, and G. Velkoski, "E-assessment with interactive images," in *Global Engineering Education Conference (EDUCON)*, 2014, pp. 484–491.
- [14] W. G. Lehnert and W. Lehnert, *The process of question answering: A computer simulation of cognition*. L. Erlbaum Associates, 1978.
- [15] S. A. Stahl, "Three principles of effective vocabulary instruction," *Journal of Reading*, pp. 662–668, 1986.
- [16] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic question generation for vocabulary assessment," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 819–826. <http://dx.doi.org/10.3115/1220575.1220678>
- [17] S. Kalady, A. Elikkottil, and R. Das, "Natural language question generation using syntax and keywords," in *Proceedings of QG2010: The Third Workshop on Question Generation*. questiongeneration.org, 2010, pp. 1–10.
- [18] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan, Overview of the first question generation shared task evaluation challenge," in *Proceedings of the Third Workshop on Question Generation*, 2010, pp. 45–57.
- [19] Y. Liu, S. Li, Y. Cao, C.-Y. Lin, D. Han, and Y. Yu, "Understanding and summarizing answers in community-based question answering services," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 497–504. <http://dx.doi.org/10.3115/1599081.1599144>
- [20] X. Yao, G. Bouma, and Y. Zhang, "Semantics-based question generation and implementation," *Dialogue & Discourse*, vol. 3, no. 2, pp. 11–42, 2012. <http://dx.doi.org/10.5087/dad.2012.202>
- [21] M. Gusev and G. Armenski, "e-Testing usage in e-Education," in *e-Learning Paradigms and Applications, Agent-based Approach*, M. Ivanovic and L. Jain, Eds. Springer Verlag, 2014, vol. Studies in Computational Intelligence, vol. 528.

AUTHORS

Sasko Ristov is with "Ss. Cyril and Methodius" University – Skopje, Macedonia (e-mail: sashko.ristov@finki.ukim.mk).

Marjan Gusev is with "Ss. Cyril and Methodius" University – Skopje, Macedonia (e-mail: marjan.gushev@finki.ukim.mk).

Goce Armenski is with "Ss. Cyril and Methodius" University – Skopje, Macedonia (e-mail: goce.armenski@finki.ukim.mk).

This work was partially financed by the Faculty of Computer Science and Engineering at the "Ss. Cyril and Methodius University", Skopje, Macedonia.

This article is an extended version of a paper presented at the IEEE EDUCON2015 Conference, held from March 18th-20th, 2015, in Tallinn, Estonia. Manuscript received 02 May 2015. Published as resubmitted by the authors 20 August 2015.