Diagnostic Testing in Mathematics

Ng Tze Beng
Department of Mathematics
National University of Singapore

Introduction

Instructors or lecturers in higher educational establishments or universities oftentimes echo the following sentiments:

> " Students don't seem to have the basics"
> "Their standards seem to be falling from year to year "
> "School education does not seem to produce students who are mathematically ready for university education"

The problems perceived by those in higher education, see for instance [8, Gatsby Report], are :

- A serious lack of essential technical facility - the ability to undertake numerical and algebraic calculation with fluency and accuracy.
- A marked decline in analytical powers when faced with simple problems requiring more than one step.
- A changed perception of what mathematics is - in particular of the essential place within it of precision and proof.

There are many reasons and contributing factors to this state of affairs. One of them is that schools are concerned with producing good examination results rather than preparing their students for higher education. There are also inherent difficulties in learning mathematics. The process by which learning takes place and the growth of mathematical knowledge within an individual is not very well understood. There is now more emphasis on embodied concepts and processes and, in more recent times, on computer technology in much the same way the calculator has become an instrument of embodied arithmetic. A quick scan of the syllabi will review learning outcomes phrased in terms of ability to carry out certain processes, which may in turn be interpreted as the goal of achievement, for example, the ability to differentiate be translated to be the ability to remember the few rules for differentiation of polynomials, trigonometric, exponential, logarithmic and hyperbolic functions. Rote learning and other devices each individual may invent to cope with the ever increasing sophisticated knowledge, in unfamiliar and new ground, will present many aspects of contradictions, conflicts and confusions and may be an impediment to learning. Unless one is able to adjust to conflicts, resolve contradictions that new knowledge presents, reassemble or reconstruct or reprocess these pieces of knowledge and their linkages, even at times abandoning conflicting pieces of knowledge, it is difficult to move on to new knowledge. It is hoped that a diagnostic test may reveal some of these difficulties, namely that of linkages between embodied concepts and procepts, and procepts and formal concepts [3, David Tall, 2005]. It is envisaged that diagnostic tests will help to provide a good foundation for the higher goals of a mathematical course. With the valuable information that diagnostic test provided, we can then plan curriculum revision and devise teaching methods to address some of the short comings. There are a number of learning theories, theories on mathematical growth, on mathematical thinking and on various levels of conceptual learning that one can become attuned to, so as to design our curriculum and learning and teaching activities . More recently, through the works of Gray, Skemp, Tall and many others, there begins to emerge a theory of mathematical growth [3, David Tall, 2005] and new insights into cognitive development encompassing the meaning of mathematical concepts, procepts, cognitive units, connections and cognitive compression and how linkages play in the development of cognitive structures. I have used terms like *embodied concept*, *procept* and *formal concept*. These and the terms, such as *cognitive unit*, *embodiment* and *compression* are explained in [4, Gray & Tall, 1994], [1, Barnard & Tall, 2001] and [2, David Tall, 2004]. Briefly, a procept refers to the manner in which we cope with symbols representing both mathematical process and mathematical concepts. Examples include, $2 + 7$, $ax^2 + bx + c$, $\frac{d}{dx}(e^x \sin(x))$, $\int e^x \cos(x)dx$. Embodiment, in the sense of Tall, refers to conceptual embodiment in which we reflect on our sensory perceptions and imagine relationships through thought experiments. Examples include velocity (as embodiment of derivative) or the infinite line (as embodiment of the real number system). A formal concept is built up from formal definitions, axioms and logical deduction.

The broad aim of diagnostic testing or more specifically computer diagnostic testing is to discover student proficiency in "basic mathematical skills". Its other modest aim is to provide whenever possible, explicit identification of errors or misconception or other possible confusion and wherever possible, corrective actions for or source for further investigation of errors or tutorial assistance. This is where it is most valuable.

In what follows, I shall give an anatomy of a question type and elaborate on possible technology that can support its delivery over the world wide web. These question types cannot replace the traditional pencil and paper test, where summative information is sought and where the question spans a range of skills and levels of difficulty. For example traditional questions may ask for proofs of results which may require the ability to process the information given (using basic skills) and form linkages with pieces of knowledge that are relevant and form logical deduction to reach a conclusion. At each stage of the process or thought, there has to be some skills and some procept linkages to fall into a coherent whole. An objective test could not reveal some or all of these but rather aims at the skills level and the possible linkages to the procept or procedural-concept level. These question types target basic skills, specific tasks and perhaps certain cognitive units. They can still reveal valuable information for learners as well as instructors. Learners can assess how much of the knowledge at the procept level is assimilated and understood and perhaps how far towards the symbolic-formal conceptual level he or she has reached. For the instructors, the information will help to assess how much support and assistance needs to be given to enable the linkages between embodied concept and procept and between procept and formal concept to form and strengthen and to design or formulate a baseline for curriculum development.

We shall next concentrate on aspects of the preparation of question style and how technology can be harnessed. There are a number of platforms commercially available for construction of question type but they are mainly for static question with little or no avenue for tailored or customized feedback.

Anatomy of a Question.

Objective test uses various question types. Question type may be categorized into the following groupings, largely based on visual configuration or presentation, input methods, answering methods or supported technology.

1.Numerical input
2.Multiple choice
3.Multi-respone
4.2 part sequence
5.3 part sequence
6.Data table
7.Hot spot
8.Hot line
9.SVG
10. Symbolic input
11. String Input
12. TFU (True, False or Undecidable)

We shall give a brief description of the types of question listed above. Numerical input is, of course, one where the participant needs to submit a number, either integer or decimal or numerical strings. For the case of decimal input, the number of significant figure or decimal places may be specified, ostensibly to deter guessing. For multiple choice question type, a number of choices may be presented with one or none correct answer. A multiple response question type is similar to multiple choice except it has more than one correct answers or more than one answer is sought from the presented list. Sequential question type, as its name suggests, requires input of more than one answer, usually numerical, where the second or third answer depends on the previous answers. Data table question type is driven by a data and question is posed based on the data and it may be numerical input, sequential, multiple choice or multi-response question possibly requiring logical input. Hot spot question type consists of a typical scenario rendered in graphics and the correct answer is encoded in pixel on the graphic, where the participant has to place the marker or pointer for submission. Hot line question type is based on a sequence of steps of mathematical processing, where one or more steps contain an error. The participant has to determine where the incorrect steps are. The question is usually presented in lines, where each line is numbered and the participant has to input the line number or numbers as appropriate for submission. At the present SVG question type is strictly speaking, not a special question type, it consists of a question with graphics rendered in SVG and algebraically coded and the question may be anyone of numerical input, multiple choice, multi-response or sequential and is not limited to these types. It is listed as a question type because of its potentially demanding technical nature. (It is possible but not demonstrated yet using SVG to render question where the participant is required to move objects rendered in SVG graphics, create geometric objects or drag nodes to a point in response to the question.) Symbolic input question type requires the user to input mathematical expression using an input control applet. It may be sequential and may involve SVG component. There are other types of question such as matching question and Flash question where Macromedia FLASH coding and plug-in is used. Of course there are also the usual string input question where participant is required to input a string and the ubiquitous TFU question where the participant has to make a decision as to whether the statement is true, false or undecidable.

Whatever the question type, each question has different components for different purposes, different pedagogic intent or different technical presentation. We call this assembly of different components a question style, where the question is realized only at run time and are not the same each time. Thus there will be random parameters that construct the configuration at run time for presentation. Obviously the question style will have a pedagogic structure and algebraic or symbolic structure. Thus a setter will need to do a pedagogic study at the outset, for the intended question, the designer will need to decide which skill to be tested, which skill is to be assumed and at what difficulty level to target. Once the decision is made and the pedagogic structure is in place, then it has to be translated into algebraic code. Whatever the question style, it will have correct answer or answers and anticipated incorrect answers based on the pedagogic structure. For instance, if we are testing the Chain Rule for differentiation, we may at the outset decide what kind of functions are to be involved, the rule for differentiation of the constituent function may be assumed, the mal-rules or the possible incorrect answers may be based on the incorrect order of the derivative of the function, incorrect application of the rule for a derivative, or one may throw in a completely unrelated answer, for instance, integrate one of the functions just to trap wild guesses. The correct answer together with the incorrect answers based on mal-rules (a term used by Martin Greenhow of Brunel University) [5, Martin Greenhow, Dan Nichols & Mundeep Gill, 2003] will have to be algebraically coded so that at run time the question style may be realised as one of the question types.

The question structure will have random parameters in the encoding so that at run time, the data are generated together with possible configuration, question phrasing, such as names, places, system of units and perhaps possible geometrical configuration. The mal-rules need to be prepared carefully so that their numerical or symbolic realization should be distinct. This may at times present some mathematical problem in itself and may present itself as a mathematical resolvent avoiding common intersection of the mal-rules. We may more conveniently use programming techniques using data within a specific range to prograrm out the common intersection. Pedagogically, feedback to questions is important. There are basically two types of feedback, feedback to the setter and feedback to the participants. Different kind of feedback to students may be realized at run time and may have an immediate impact to participants. Feedback may be specifically linked to specific choices or actions or input by the participant. If the answer is an anticipated incorrect answer, the feedback may present some general theory and background information and a specific solution to the question. We hope the participants will study the feedback and having answered the same question style many times and view the feedback many times over, will learn the algebraic structure of the correct answer. We hope the participant will then realize that the algebraic structure is the same and form a linkage to the cognitive unit presented in that particular question style and thus be able to do similar kinds of problems. The next kind of feedback relates to the kind of error or misjudgment the setter has anticipated. This information is presented, highlighted together with the specific place where the mistake has occurred and how it could be avoided. Finally we may wish to present appropriate resources for references or for remedial action. This is usually based on the tagging of the question, either according to difficulty level, or by the more familiar educational level, ‘O’, ‘AO’, ‘A’, ‘AS’ , etc. One may also tag the question according to the response or by a modified BLOOM taxonomy.

All of the above desired components for the question style can be coded with the various modes of feedback with considerable effort. Each question style has to be coded almost manually. One consolation factor is that we may actually realize with one question style, thousands or millions of questions at run time. It is a wishful thinking that we can automate the whole process of question style construction. It may be possible in the future when higher level of AI is built into standard programming engine. We can, to some limited extent, automate the coding of the question, once the mal-rules are algebraically coded and mathematically resolved and we can do this together with the feedback. Once the algebraic structure is resolved, we can even automate the coding of the visualization and we can definitely automate the marking. The almost unlimited configuration of question styles make it a daunting task to devise an interface, a platform where the pedagogic structure can be designed according to certain mal-rules, where mal-rules can be automatically resolved and the feedback can be generated or retrieved as and when necessary. The following is a list that a computer assessment system should ideally be able to perform.

1. Give question analysis to code algebraically the anticipated incorrect answers based on mal-rules.
2. Automate the coding of the different question types.
3. Automate linkages to a data bank of information for feedback
4. Automate the graphical input for the question as well as the graphical output for the feedback
5. Automate linkages to appropriate learning system for individual for follow up and exploration
6. Automate marking, result analysis, reporting and or class profiling

It may not be possible to devise a system that can perform all of the above tasks. However with some careful consideration of various types of cognitive units within different topics in algebra, calculus or statistics, it is possible to confine ourselves to a subsystem in a smaller scale to do most if not all of the above tasks. For instance, a system catering to the learning of fractions along the lines of the paper “The development of a fraction testing and tutoring system” by Qinping He and Paul Brna [6, Qingping He & Paul Brna, 2005].

Technical and Pedagogical Issues

A great deal of the pedagogic consideration of question style construction is about how to engage the participant in learning. Responsive numerical input, MCQ and multi-response style questions reward instantly by providing tailored feedback. It is hoped that the participants will read the feedback and wish to follow this avenue for learning that may not be possible with conventional class room activities, where the tutor is too busy to deal with this type of cognitive units or questions where the skills or knowledge are assumed. MCQ or multi-response questions should give information from the answers. Wrong answers do tell a great deal, when taken by a class, about the kind of mistakes or misconceptions that have taken place. It can help the teachers with curriculum design, remedial or corrective lessons or just about where to start teaching by designing a reasonable baseline. Non MCQ or multi-response questions may be processed in the same way with anticipated incorrect answers forming an invisible component of the question style. MCQ or multi response question should have meaningful distracters constructed from mal-rules. Knowing which incorrect answer the student chooses or input is vital for diagnosis.

Mal-rule, a term used by Martin Greenhow [5, Martin Greenhow, Dan Nichols & Mundeep Gill, 2003], roughly speaking is incorrect reasoning or incorrect derivation arising from the following (not an exhaustive list):

1. Unfamiliarity with the topic
2. Confusion of symbolic representation
3. Incorrect application of algebraic rules
4. Conceptual mistake
5. Conceptual misrepresentation
6. Pro-cept conflicts
7. Instructional biased reasoning
8. "Cultural" slant.

This list is of course not exhaustive. Too many mal-rules presented in parallel will make the question clumsy, too time consuming and even too difficult, A rational way is to choose only the most commonly occurring types and limit ourselves to not more than four. Therefore, the question style designer would have to base his or her choice on analysis of past performance or answer files or examination statistics or feedback.

It is not easy to codify mal-rules. Using mal-rules to construct distracters presents some problem in "automatic" or "random parameters" question style construction. The question is constructed at run time; the question style simply specifies how the individual components are to be configured at run time. The questions will have a very small probability of being the same and so the students will not be able to pass the question information around. Mal-rule, if appropriately algebraically coded may give rise to identical numerical values or geometrical configuration or be algebraically equivalent (to be avoided altogether). Therefore, the mal-rules need to be algebraically resolved so that the realization at run time are distinct, whether it is numerical or non-numerical. This can be done by avoiding common intersection of numerical realization either by (1) suitable constraint of the values for the random parameters (2) mathematically worked out the intersection data within a certain range of the values for the parameters or (3) use a programming technique to resolve the realization. There is also the presentation problem for mathematical expression with random parameters realized at run time. For instance, we do not write $1x$, or $x^2 + -x$ or x^1. We may have to put our data through an engine to weed these presentations out.

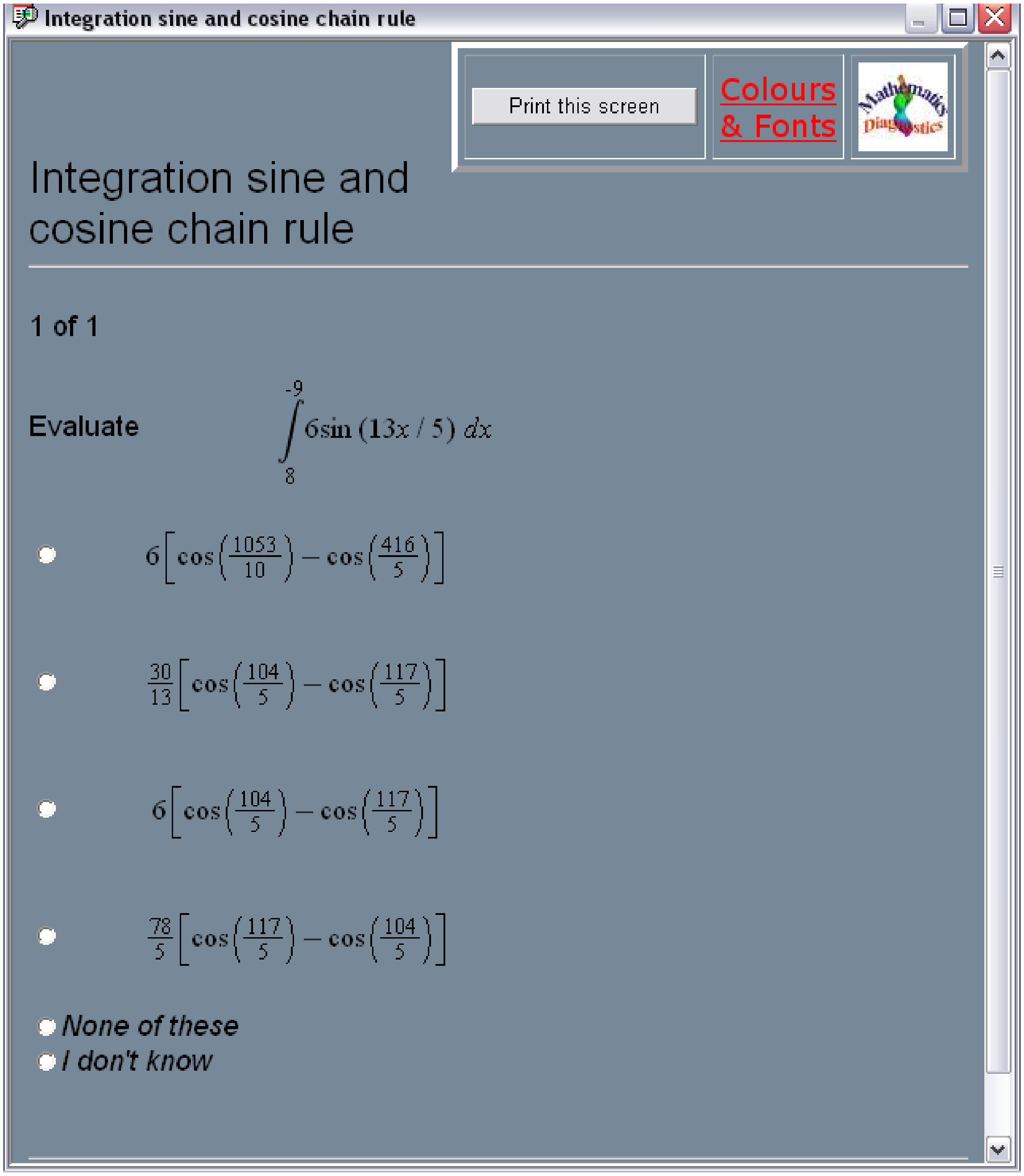
Presentation and Technology issues

Firstly we should be able to detect students answering question backwards or incomplete answers (quick guessing). For instance, a question about anti-derivative can be done backwards by differentiation. To deter this mode of answering we may want to target at the specific method of finding anti-derivative, for instance integration by parts, change of variable or substitution, trigonometric substitution or partial fractions and target at some determined intermediate steps and the specific skills or rules involved. We may just simply consider converting into a definite integral, where of course the knowledge of antiderivative will be required. Identical numerical realization should be avoided as discussed before. For instance, $2x$ and x^2 give rise to identical numerical value when $x = 2$. Mathematical display problem can be avoided by the use of MathML and java applet that allows for string concatenation such as Webeq control from Design Science. Display problem can arise from randomized parameters taking values of 1, −1, or 0. For instance, we normally do not wish to display $0x$, $1x$, $+-x$ or x^1 . We can put our algebraic code strings through a presentation resolver to avoid presenting the above display. There is also the problem with graphic display. At run time realization, the relevant figure may be too small or part of it may be outside the view window. The algebraic coding would need to take this into consideration, with an additional constraint presented by graphical visualization. Certainly a tiny triangle is still a triangle but it is not as forceful as a triangle with all 3 sides clearly visible and the pertinent data reinforces that configuration. Scalable Vector Graphics (SVG) coding can encode the relevant information at run time with string concatenation. A free Adobe SVG viewer (a plug-in) is available for the internet explorer to display the SVG graphics. SVG can indeed encode 2D and, to some extent, 3D graphics with interactive animation.

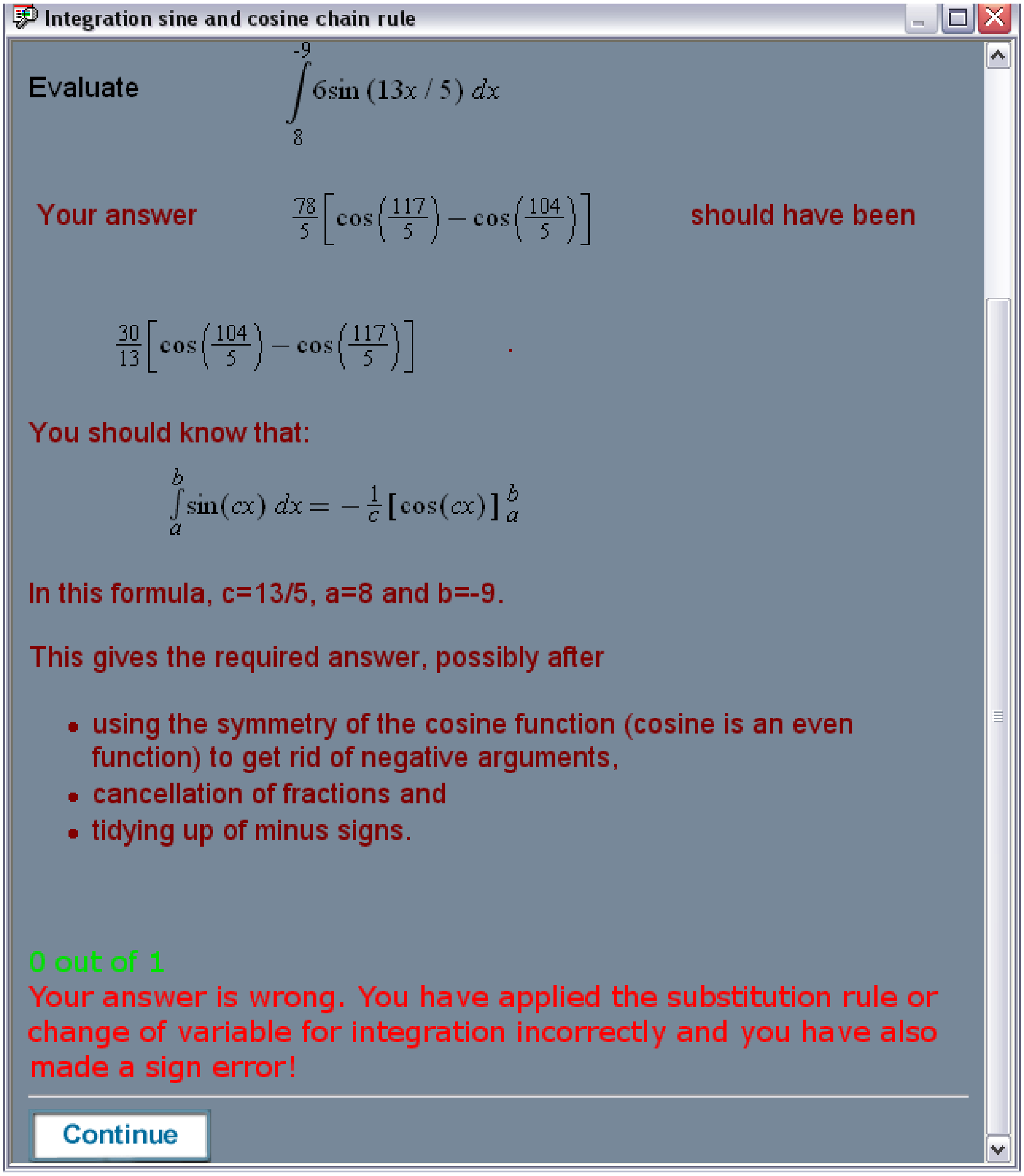
Question styles, be they MCQ, multi-response, sequential, numerical input or hot line question uses javascripts to assemble xhtml codes, MATHML and SVG codes to display text, mathematics content and SVG graphics. The javascripts are also used to resolve mathematics display problem, construct question type and mathematical functionality such as matrix multiplier to construct the MATHML code for display, polynomial multiplier, etc. Besides those that are used in general question type assembly, the javascripts mainly fall into three main types: (1) javascript that returns numerical values or strings, (2) javascript that returns MATHML strings and (3) javascript that returns SVG coding for rendering SVG graphics. These functions are components of the question and feedback constructor. It assembles XHTML code with inline MATHML and inline SVG coding at run time for a document to be displayed by an internet browser the various type of question and the feedback. A typical feedback for incorrect answer will have some or all of the following items:

1.The correct answer
2.The general theory
3.The actual run time realized question with fully worked out solution in parallel and possibly a diagram rendered in SVG
4.A setter’s interpretation of why the student went wrong
5.Possible pop-up window for lengthy general theory or details or reference for reading or other resources.

The following screen shots shows an example of a question and its feed back. Here we use an extended version of the Questionmark Perception software, where our questions are codified as discussed above and are portable, independent of the software, being standard xhtml and javascripts.



This is a simple feedback. Notice the difference in the font color. It contains actual detail feedback and comment about the answer. We may include information for resources and anything the setter wish to say. A pop up window button maybe included for lengthy detail feedback, such as a general theory. Inline SVG graphics when necessary, may be included, as well as automated resource information according to the question tag.



Finally for the purpose of assembling a testing programme, we can select our questions according to some scheme, topic by topic and according to BLOOM taxonomy or by some more familiar means of selection criterion such as by syllabus or by difficulty level. This means we need to tag the question according to our chosen classification. Thus we can tag the question either by a modified BLOOM taxonomy, the Smith taxonomy [7, Sangwin, 2002], or by the familiar, ‘O’, ‘AO’, ‘AS’ or ‘A’ or by difficulty level: easy , intermediate and hard or by attainment level: undergraduate level or advance undergraduate level. An assessment regime can at run time select questions according to the tagging or classification. We can also tag or classified the response. This is for the purpose of answer file analysis. The categorization or classification of response is the same as classifying mal-rules and depends on the type of mal-rules constructed which is based on teacher's experience, students' revelation, analysis of past examinations, CAA answer files, etc. All the above data can be collected for a profile building of a group or a class.

The advantage of using javascripts and XML compliance xhtml is that it is more economical than using a proprietary system in terms of cost and that the codes are portable and may be converted into standalone unit to be incorporated into a lesson or just for online formative learning. Schools and universities can share resources, question banks and development costs. Mathematics support centre may act as a coordinating centre for resource building, archiving and technology support. In particular, teachers' valuable experience may be codified and share, thereby fostering a greater awareness of cognitive development and pedagogic issue, which will help to promote a responsive and collaborating community of teachers.

References

[1] Barnard, A. D. & Tall, D. O. (2001) A Comparative Study of Cognitive Units in Mathematical Thinking. In Marja van den Heuvel-Panhuizen (Ed.) *Proceedings of the 25th Conference of the International Group for the Psychology of Mathematics Education* 2, 89–96. Utrecht, The Netherlands.

[2] David Tall (2004). Thinking through three worlds of mathematics, *Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education*, Bergen, Norway, 4, 281–288

[3] David Tall (2005). A Theory of Mathematical Growth through Embodiment, Symbolism and Proof. Plenary Lecture for the *International Colloquium on Mathematical Learning from Early Childhood to Adulthood*, Belgium, 5-7 July 2005.

[4] Gray,E.M. & Tall, D.O.,(1994) , Duality, ambiguity and flexibility: A proceptual view of simple arithmetic, The Journal for Research in Mathematics Education 26, 2, 115-141.

[5] Martin Greenhow, Daniel Nichols, Mundeep Gill, Pedagogic issues in setting online questions, MSOR connections 25-28 Nov 2003 Vo3 No. 4

[6] Qingping He and Paul Brna, The Development of A Fraction Testing and Tutoring System, Proceedings, 9th CAA Conference 2005.

[7] C J Sangwin, New opportunities for encouraging higher level mathematical learning by creative use of emerging computer aided assessment. May 2002, Mathstore LTSN Maths, Stats & OR Network, CAA series

[8] Measuring the Mathematics Problem (Gatsby Report), 2000, The Engineering Council.

December 2005.