A WEB-BASED DISTRIBUTED AND INTEROPERABLE TOOL FOR SHARING MATHEMATICAL ASSESSMENTS AND SUPERVISING ONLINE TESTS

A dissertation submitted
to Kent State University in partial
fulfillment of the requirements for the
degree of Doctor of Philosophy

By

Saleh Al-shomrani

December, 2008

Dissertation written by
Saleh Al-shomrani
B.Sc., King Abdulaziz University, Saudi Arabia, 1997
M.S., Ohio University, USA, 2001
Ph.D., Kent State University, USA, 2008

Approved by

_____, Chair, Doctoral Dissertation Committee
Dr. Paul Wang
_____, Members, Doctoral Dissertation Committee
Dr. Arvind Bansal
_____
Dr. Jonathan Maletic
_____
Dr. Michael Mikusa
_____
Dr. Murali Shanker
_____
Dr. Ulrike Vorhauer

Accepted by
_____, Chair, Department of Computer Science
Dr. Robert Walker
_____, Dean, College of Arts and Sciences
Dr. John Stalvey

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

DEDICATION

I would like to dedicate this dissertation to my beloved parents, for supporting me since my childhood.

To my wife, my love, for her support and patience through the entire degree process.

To my sons, Mohammed and Yazeed for their understanding that I must study, spend time away from you, and write this dissertation.

Now that the dissertation is finished, there will be more time for family!

To the remainder of my family, brothers and sisters, thank you all for your support and encouragement.

ACKNOWLEDGMENTS

I thank Allah for my educational accomplishments, and the patience to complete this degree. Next, I thank Dr. Paul Wang, my advisor, for his support and his high expectations of me. He was a very important person in the shaping of my intellectual thinking. Dr. Wang, I have learned much, in the years that I have worked with you.

I thank my exceptional committee members for their support and encouragement: Dr. Bansal, Dr. Maletic, Dr. Mikusa, Dr. Shanker and Dr. Vorhauer. Without your suggestions, this dissertation might not have been finished. The Computer Science Department and the staff have been extremely supportive of me through my studies here at Kent State University. I thank my fellow WME members, Xiao, Xun, Wei Su, and Dean, for their encouragement and support. And the many acquaintances and friends I have made during this journey.

CHAPTER 1

INTRODUCTION

1.1 The WME Project

An interdisciplinary group of researchers at the Institute for Computational Mathematics (ICM) has been working on a *Web-based Mathematics Education* (WME) system to apply Web and Internet technologies in mathematics education [22]. The WME work grew out of research on *Internet Accessible Mathematics Education* (IAMC) and collaboration between the ICM group and professor Michael Mikusa of the College of Education at Kent State University. Work on WME shifted into high gear with the Ohio Board of Regents (OBR) Research Challenge grant in 2004 when a pilot system was first put to trial at Kimpton Middle School (Stow Ohio) [21]. WME is a modern distributed system on the Web for mathematics education. It connects mathematics education content created by experts to K-12 teachers and students. WME can deliver classroom ready, dynamic, and hands-on lessons and modules as well as provide assessment services, teacher guides, education research materials to teachers. *Research on DMAS was started to address the assessment needs of WME to have an Innovative Web-based Mathematics Assessment System at the middle school level.* DMAS actually started as a component of WME system and was called (DMAD) to serve as assessment database for WME

assessment materials (Figure 1.1). Today DMAS is an independent and flexible system that can be used with WME as well as other Web-based education systems.

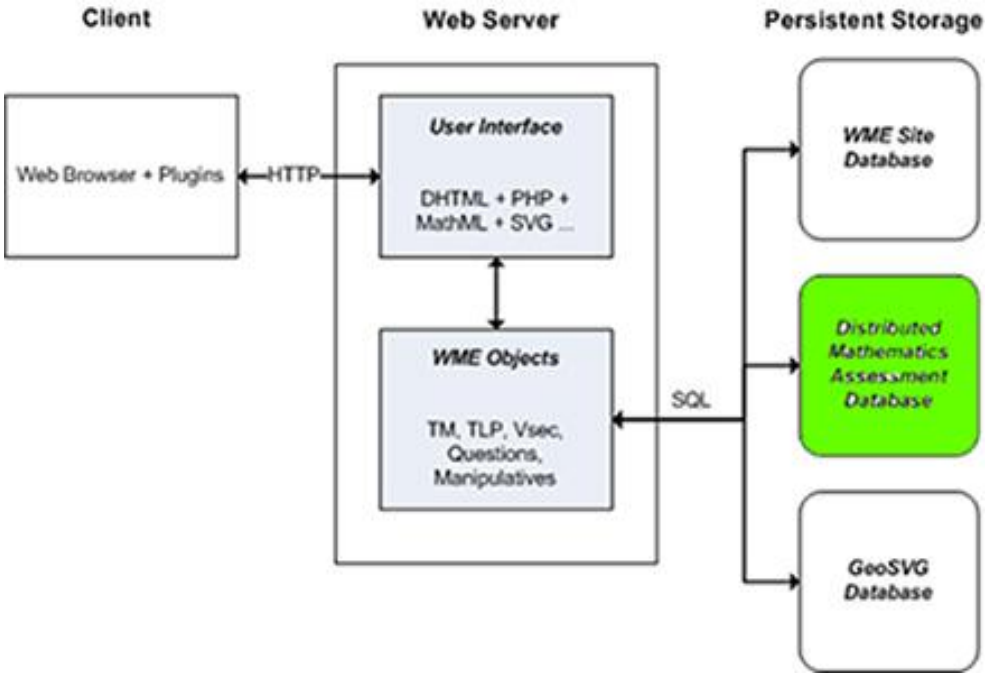


Figure 1.1. DMAS Started as a Component of WME

1.2 The Need for Assessment

Definition: In this dissertation, I use the term "*mathematical assessments*" to mean tests and test questions, for teaching and learning mathematics. One important goal of DMAS is to enable teachers and experts from many different DMAS installations to author/customize mathematical tests and test questions to form an ever-growing assessment database that is easily searchable and widely shared.

Assessment is important to measure the effects of educational concepts, student performance and comprehension. Assessment tests should also diagnose learning difficulties and determine knowledge shortfalls. The United States No Child Left Behind Act of 2001 accountability components include “performance on state designated assessments in reading and mathematics”. To support mathematical assessments, I offer a Web-based Distributed Mathematics Assessment System (DMAS) that can support the assessment needs of mathematics for teachers and students and can serve and interoperate with other online systems such as WME [25] as well as work independently. DMAS aims to become a key resource for sharing mathematical assessments.

1.3 Motivations for DMAS

Online assessment vs. paper assessment: Many research papers and studies suggest using online assessment over paper assessment and list many advantages of using internet based assessment over paper based assessment [11, 12]. Some advantages include: online assessment makes it possible for students and teachers to get quicker and more feedback than they get using paper based assessment. The online assessment has good support for teaching and learning environments. The online assessment has easier and wider accessibility through the internet. Time saving and some other advantages from teachers and students were mentioned in [12].

1.3.1 Existing Systems

Current existing research work: Research work on online assessment for teaching and learning mathematics has been going on for quite some time [1, 2, 4]. Such

systems have a lot of useful mathematical questions and resources, comply with educational standards, some are supported by governments and non-profits organizations, and some have been used in schools and educational institutions. Many other existing systems provide various assessment materials for mathematics education. Existing systems such as the Math Forum [35] has the Internet Mathematics Library which provides different types of assessment resources and ideas including articles, books, and links to other assessment websites. The PBS TeacherSource [42] has various assessment techniques, ideas, and strategies on the Web. The WebCT [47] provides a wide range of assessment information and articles. The AAC (Alberta Assessment Consortium) is a non-profit organization [49], and provides assessment rubrics, materials, and publications. Other assessment systems and resources are also available in [39, 45, 46, 48]. However, such systems are *Traditional Web Assessment Systems (TWAS)* in which they have similar system structure as a one fixed and centralized architecture (Figure 1.2).

The central system architecture has many *disadvantages*: Assessment materials and resources are *centralized* (centralization problems), this causes traffic congestion and if a central assessment site goes down then all assessment materials on this site are inaccessible. Other disadvantages include that some of these systems require users to have some previous knowledge of technology, programming, or CAS syntax to be able to use their assessment systems [1, 2]. Such TWAS systems have *static (pre-made) mathematical assessments* that limit customization, usability, and sharing of

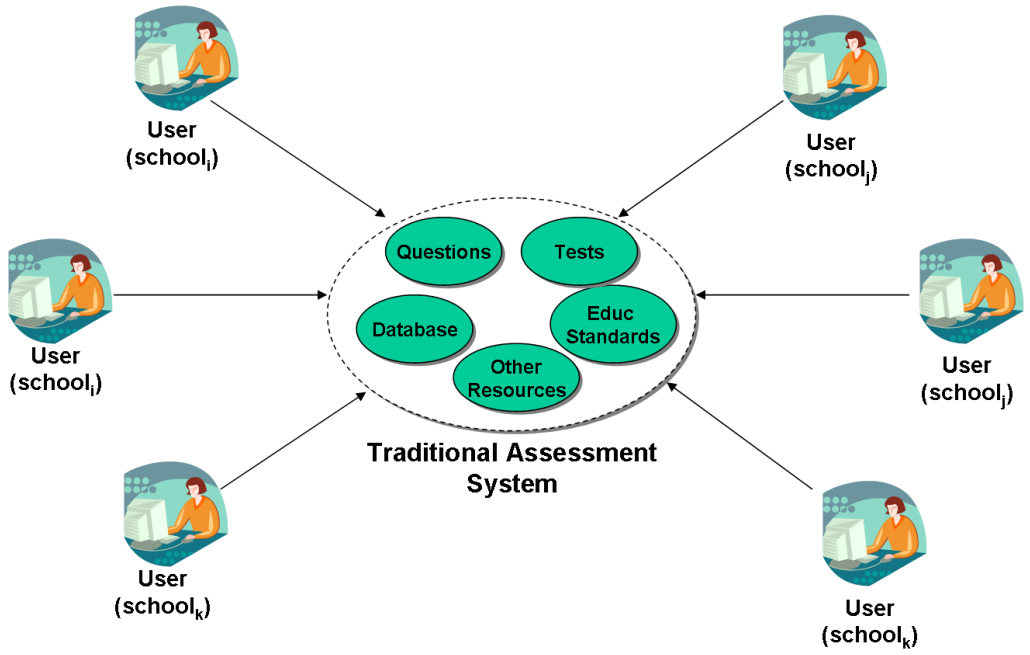


Figure 1.2. Traditional Web Assessment Systems (TWAS) Architecture

mathematical assessments on the internet. Another problem is *lack of locality*, if teachers from same school need to (or required to) do mathematical assessments locally within school that is not supported by TWAS systems. For instance, if we look back at (Figure 1.2), and if the two users from *school*$_i$ want to share their mathematical assessments locally within their school, that will not be supported by such TWAS architectures. Math formulas in some TWAS are also represented in *non-standard Web formats*: infix, image formats, or LATEX, and geometry objects are represented as images (gif, jpeg, png, etc.). Finally, they *lack interoperability* with other online systems on the internet to share their mathematical assessments.

As a result of TWAS central structure problems, it was a motive to consider an alternative direction with more integrated, flexible, and customizable framework to

support mathematical assessments in mathematics education on the Web and to be interoperable with other online systems through standard Web technologies and protocols. In this dissertation, I am offering the DMAS system as an *alternative assessment framework* to the current TWAS systems. This will cover the research issues, goals, and concepts of DMAS system, in addition to describing the design of DMAS system and many practical features including online authoring tool, test giving and its security, handling *mathematical expressions and geometry, automatic interaction and communication between teacher and students in class room (TSIM) for teaching and learning purposes*, providing *Automatic generation of Mathematics formulas/questions using Free Parameter Question (FPQ) and Mathematical conditions, and automatic Sample Question Generation Feature (SQGF),* a simple Web-Based Mathematical Answer Checking Service for DMAS (DMAS-MACS) in two ways: *Client-Side and Server-Side Synchronization Method (CS-Sync)* and *Client-Side and Server-Side Asynchronization Method (CS-Async),* and interoperability between DMAS and other online systems such as WME system.

1.3.2 DMAS Research Goals

The goals of this research are to:

(a) Find and design an *alternative assessment framework* to the existing TWAS systems to better allow sharing of mathematical assessments.

(b) Enable teachers and experts to contribute into question bank.

(c) Help teachers create and manage online mathematical tests.

(d) Allow students to take mathematical tests online.

(e) Provide real-time interactions between teacher and student(s) during a test for educational purposes.

(f) Capable of working with other Web-based mathematics education systems.

1.4 Research Hypothesis and Questions

This dissertation investigates the following hypothesis and questions: The problem is finding another reasonable assessment system structure as an alternative to TWAS. In other words, how do we design an *underlined framework that will collect* mathematical assessments from teachers and experts in different school locations, promote *sharing* of mathematical assessments, handle searching, importing and exporting questions of different types with different components, and create growing question bank? (DMAS Framework). How can an assessment system provide a *real-time learning/testing environment* through useful online *interactions and communications* between a teacher and students in a class room privately and securely? (TSIM tool). Can we dynamically create *multiple instances on-the-fly* of same mathematical problems? (FPQ and SQGF). How to make such important mathematical assessments *shared widely on the Web*? (MAML). How do we author/represent *mathematical expressions and geometry graphs* in questions on the Web? (DMAS Authoring Tool).

1.5 Research Results and Contributions

1.5.1 Main Research Contributions

This dissertation gives the following research contributions: Designing a cross-browsers structure, a *Web-based Distributed Mathematics Assessment system (DMAS)*,

that enables a *growing shared database* of contributed mathematical assessments and a *DMASEngine* to search for assessment questions to better deliver mathematical assessments to teachers in different installations. The system has customizable authoring tool for teachers that handles customizing, authoring, importing and exporting questions, and sharing of tests and questions, and allows questions to contain mathematical formulas and geometry graphs. *Real-time test supervision through TSIM (Teacher-Student Interaction Mechanism),* it is a way of secured and private interactions and communications between teacher and students in a class room to provide a teaching/learning environment. Another contribution is an *Automatic generation of Mathematics formulas/questions based on mathematical conditions*: Free Parameter Question (FPQ) and Sample Question Generation Feature (SQGF). Finally, *Web Interoperability and XML representation of tests and test questions, MAML*, to allow sharing of mathematical assessments widely on the Web.

1.5.2 Implementation (and other) Contributions

This dissertation has some other contributions: An experimental and simple Web-Based Mathematical Answer Checking Service, DMAS-MACS, (CS-Sync: Client-Side and Server-Side *Synchronization* Method and CS-Async: Client-Side and Server-Side *Asynchronization* Method), online test giving/taking, handling grading and results administration, supporting different DMAS file formats DFF (.xls, .doc, .pdf, .txt, .rtf, .ppt, .csv) for questions and grades, building of experimental system to work with WME, and Pilot use in schools to collect teacher and student feedback.

1.6 Organization of the Dissertation

This dissertation is organized in sequence chapters as follows: in chapter two, an overview of Web-based DMAS system, and viewing DMAS from different perspectives: User view, global architecture view, per-school view, and Web application view. Also in this chapter, an introduction to DMAS components: Starting with organization of the distributed assessment database, storage and representation of tests and test questions, test database search engine (DMASEngine), test authoring and sharing, online test giving/taking, grading and grades management, real-time online test supervision.

We will look into real-time supervision of online tests in more details in chapter three. This includes TSIM test authorization through TCP: a message window called *Teacher Control Panel*, TSIM real-time monitoring of student progress and performance, interactive instant text-messaging system using TSIM object, traditional online testing environment, real-time online testing environment, TSIM as an interactive teaching and learning tool. In chapter four we will see how DMAS system handles Mathematical expressions and formulas, fullfils mathematical conditions, and generates Free Parameter Question (FPQ) and automatic Sample Question Generation Feature (SQGF), and then finishes this chapter with DMAS and Geometrical Graphing.

Chapter five is about test sharing, customization, authoring, publishing, and importing and exporting questions. Chapter six is online test taking, automatic grading and grades management, importing and exporting DMAS File Formats (DFF), and DMAS trial at middle school. Chapter seven is DMAS System interoperability with other online systems. This covers DMAS interoperability with existing Web systems,

integrating DMAS with WME system in two modes: teacher mode and student mode. Building of experimental system to work with WME and then pilot use in schools to collect teacher and student feedback, and concludes this chapter with DMAS and other fields.

Chapter eight is an introduction to Mathematics Assessment Markup Language: MAML, to be used as a representation and encoding for sharing mathematical assessments. In chapter nine, I have started invistigating the answer checking challenges. This includes experimental works with DMAS answer checking service, answer checking problems and challenges, DMAS-MACS Answer Checking Service Implementation, CS-Sync: Client-Side and Server-Side Synchronization Method, CS-Async: Client-Side and Server-Side Asynchronization Method. I will finish with conclusions and future work including limitations of this work, and finally the References.

CHAPTER 2

OVERVIEW OF WEB-BASED DMAS SYSTEM

As with any design of a project, the starting point is to determine the requirements and specifications of the DMAS system. I started with identifying different parameters and views that will shape such distributed system such as: the intended users, goals, purposes and functionalities, and exactly what will be achieved by this design. Consulting with middle school mathematics teachers and mathematics education researchers and experts, I have specified the following requirements for the DMAS design: *Easy to access*, to be available on the Web through standard Web browsers (cross-browsers). It should be *easy to use*, having a simple and intuitive user interface for teachers and students. The system must be *safe and secure*, providing a secure environment in which to store and retrieve assessment information such as tests, answers, grades, and statistics. It supports *different types of questions*, supporting true-or-false, multiple-choices, short-answer, essay (extended answer) questions, two-columns matching, and fill-the-blank and being open for adding other types of questions. It should have *rich data format*, allowing question to contain text, images, mathematical formulas/expressions, and geometric graphs.

The DMAS supports a *WYSISWYG authoring tool*, providing an environment for teachers to create new assessment tests, locate relevant assessment materials, import, edit

and customize mathematical assessments. It should also *online testing* such as conducting assessment tests online, generating and displaying tests correctly with mathematical formulas/expressions, storing answers, and providing automatic grading support. DMAS has also a *statistical support*, supplying statistics of student performance, fast and immediate response, advice and hints for teachers regarding answers and mistakes. The system should maintain *privacy of authors* to make it possible to contribute assessment materials anonymously. *Interoperability with other existing systems* is important, having a well-defined API (Application Programming Interface) to interact with the other systems on the Web (WME system is an example), and to serve the Internet as an *assessment server*. Supporting *different Access modes* is required to provide different types of testing modes: teacher mode and student mode. Finally, the DMAS should support *extensibility* as being flexible and easily extensible for adding new features and functionalities in the future.

So after an extensive design investigation, I came up with a distributed system that will consider such requirements mentioned above. It is *DMAS*, a Web-based Distributed Mathematics Assessment System that provides a systematic way to support the assessment needs of mathematics education and a platform for teachers at different schools to contribute and share mathematical assessments. It uses distributed databases and Web technologies to achieve these functions. DMAS helps mathematics teachers to quickly and easily author, customize, administer and manage tests. They can also easily import/export mathematical assessments from/to question bank and share questions with other teachers from same or different school (Figure 2.1).

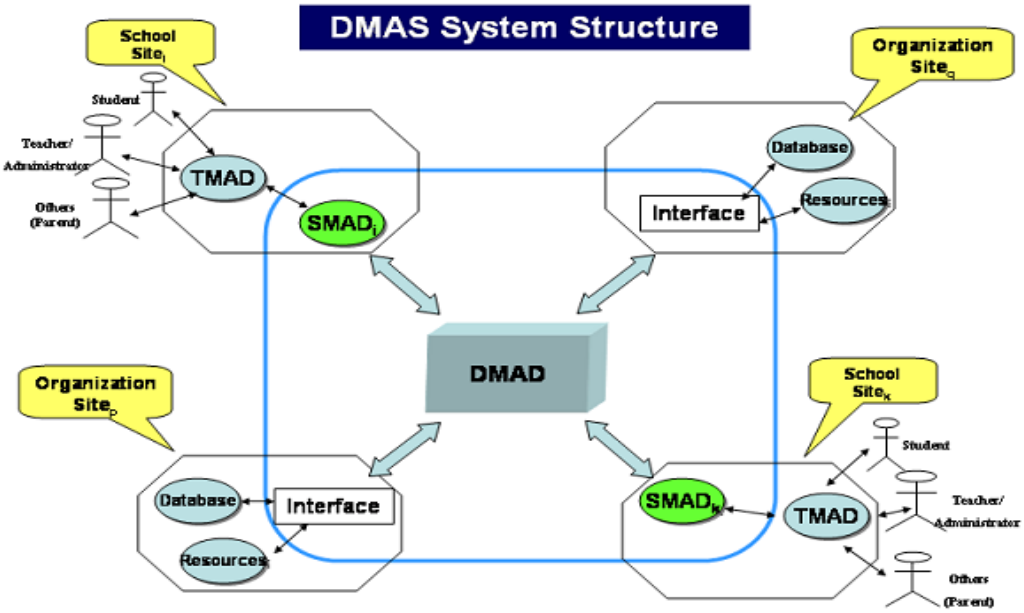


Figure 2.1. DMAS Assessment System Structure.

2.1 Viewing DMAS from Different Perspectives

DMAS system has different views and can be seen from various perspectives: From a *user view*, where the user sees the whole distributed system as one unified system much like a central system. *Global architecture view*, in which we look into the big picture of the system as a whole, whereas *per-school view* is related to doing Assessment in schools locally. Finally, the *Web application view* where through it, DMAS can interoperate with and support other Web applications (Figure 2.2). Now let's take a closer look into these views in more details.

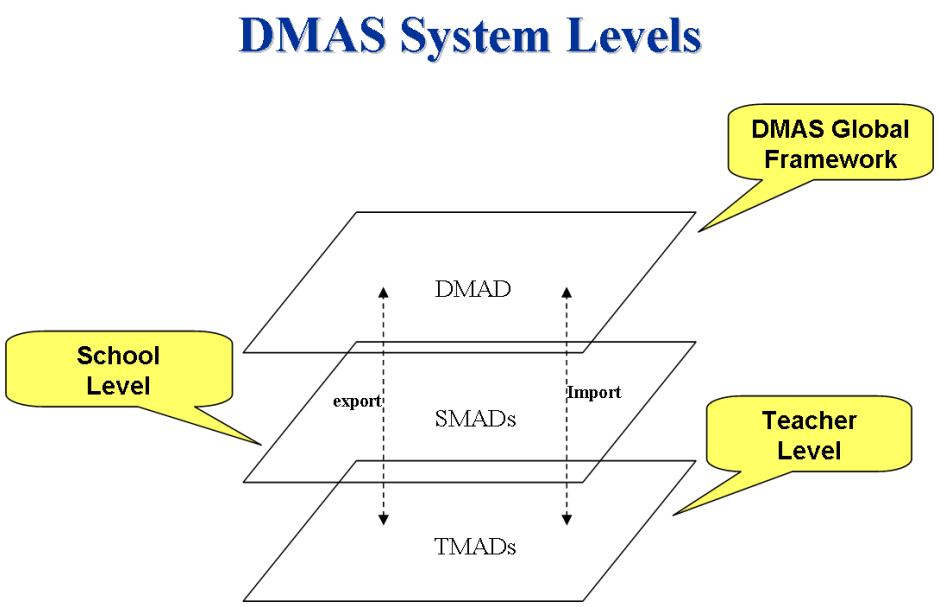


Figure 2.2. Assessment System Levels.

As we can see from the above graph, the system has three different layers/levels in each school location (site) interacting with each other through import and export operations of questions. Teachers can use the search engine (DMASEngine) to search for assessment questions to import or export through such layers. Search can be narrowed by subject, topic, grade-level, question type, keyword, and author.

2.1.1 User Views

The user has the illusion that the whole DMAS distributed system is a one and unified system much like a central system. All searching and importing and exporting processes from other locations are completely transparent from the user. Through this view, the users can do assessment tasks locally which may include:

- *Authoring Tool*:

  Teachers can create new questions/tests, view, and customize existing ones. Questions in a test may include text, geometric graphs, images, or mathematical formulas represented in MathML codes.

- *Managing Assessment Tests*:

  Teachers can prepare/generate their own assessment tests, homework assignments, or quizzes either from scratch (by authoring their own questions) or by importing questions from the DMAD bank.

- *Conducting Assessment tests*: DMAD provides teachers with two testing options: online testing and paper testing.
- *Automatic grading and results administration*.
- *Interactions and communications between teachers and students in class rooms using TSIM* for teaching and instant messaging.

2.1.2 Global Architecture View

This view represents a general picture of the whole DMAS system. More specifically, we see there is a real distributed system and mathematical assessments are scattered and distributed in different locations. DMAD, the basic component of the DMAS system, represents a huge bank/collection of mathematical assessments. This view includes:

- *Importing and exporting mathematical assessments*:

  DMAS user can import questions of interest and then all selected questions with different components (e.g. text, images, mathematical formulas, geometric graphs, etc.)

will be copied and migrated from the source site to the destination (teacher's TMAD database) transparently. Imported materials can be customized and used in assessment tests, homework assignments, and quizzes. Changes made on an imported item do not affect the original copy. On the other hand, Teachers can select questions from his/her personal TMAD and export them either to the local SMAD or to the public DMAD.

- *DMAS search*:

A teacher can easily search for assessment questions on particular subjects and at specific grade levels using the search engine (DMASEngine). The search covers the local SMAD and transparently the rest of DMAD. The search can be narrowed by subjects, topics (for example, fractions, algebra, geometry, and measurement), keywords and grade levels.

- *Adding/Deleting TMADs/SMADs from the DMAS system*:

Since DMAS system consists of different participating schools (SMADs) and Teachers Assessment Databases (TMADs) in different locations or sites, adding a new participating school site (new SMAD) or deleting existing one is an essential to the DMAS structure (Figure 2.1). Unlike central systems, the distributed nature of the DMAS system adds an extra burden on the DMAS Search engine to successfully handle issues of adding new site or removing existing school sites from the collection in such good and transparent way.

2.1.3 Per-school View

This is related to doing assessment in schools locally through this view. This means DMAS users from same school ($school_i$) can search, import or export, and share

their mathematical assessments *locally within school boundary*. This gives the teachers the freedom of searching, publishing, and sharing their mathematical assessments with other teachers from same school only without even publishing it to the public or to the global bank.

2.1.4 Web Application View (How DMAS can support Web applications)

As mentioned before, there are a lot of good resources provided by the central TWAS systems. However, they lack a way of sharing their useful materials with other systems. So, one of DMAS key features is *interoperability* with other existing Web systems. Unlike TWAS systems, DMAS aims to interact and promote sharing of mathematical assessments with other online systems through standard Web formats (XML documents), protocols, and technologies. It provides such systems and applications with required assessment data, files, and other mathematical assessments.

2.2 DMAS Components

The DMAS system structure consists of the following components: Three different types of distributed databases DMAD, SMAD, and TMAD, storage and representation of test questions, test database search engine (DMASEngine)**,** test authoring and sharing, online test giving/taking, grading and grades managements, real-time, and online test supervision.

2.2.1 Organization of the Distributed Assessment Databases

The assessment system consists of local databases at individual school websites. Each local database collects assessment questions contributed (exported) by teachers in a particular school to be used (copy imported) and perhaps shared with other teachers from same or different schools. Here are the distributed databases involved:

DMAD – The core bank and basic component, is a distributed database with *local databases* at different school sites. Within DMAD we have:

TMAD – *Teachers Mathematics Assessment Database* is a database assigned to mathematics teachers of same school. It stores and manages assessment tests, homework assignments, questions, student answers, grades, statistics and other info for each individual teacher.

SMAD – A *School Mathematics Assessment Database* is created for individual school as part of its school site. SMAD connects TMADs within the school and SMADs at different schools through the DMAD core database. The SMAD performs a critical role in enabling the sharing of assessment materials within and without a school.

DMAD is designed to work as one distributed database while providing power, unity, and convenience at each participating school. The system helps create, revise, administer, and grade exams that can contain various types of questions: MCs, true-or-false, extended (essay) questions, short answers, two-columns matching, and fill-the-blank.

2.2.2 Storage and Representation of Test Questions

Assessment tests and questions imported from question bank or authored by a teacher are stored locally in the local TMAD database so that they can be changed and manipulated by the teacher at any time. A question may be represented and stored in different formats such as text, images, MathML codes of formulas (MathML presentation code for browser display and MathML content code for computations), and SVG text code representing geometrical graph of a question. Actual images of a question are stored on the local server to save space in the database and quicker data retrieval so that the storage URI is always the same whereas the URI and other metadata are stored locally in TMAD database. On the other hand, all published (exported) question components are copied and stored globally in either SMAD or DMAD depending on the publishing type (school or public). This way of storage organization helps the database search engine to search in the right place (database) and only as needed and narrow the search area to bring only the intended data in a more efficient way.

2.2.3 Test Database Search Engine (DMASEngine)

The system allows authors not only to easily create various types of assessment questions from scratch but also to use the DMAS search engine, DMASEngine, to search for questions in their local school only or in the whole DMAD bank from all participating schools for questions with specific criteria and components and then import copies of them locally. Question can have different contents such as text, formula, graph, and other data (Figure 2.3).

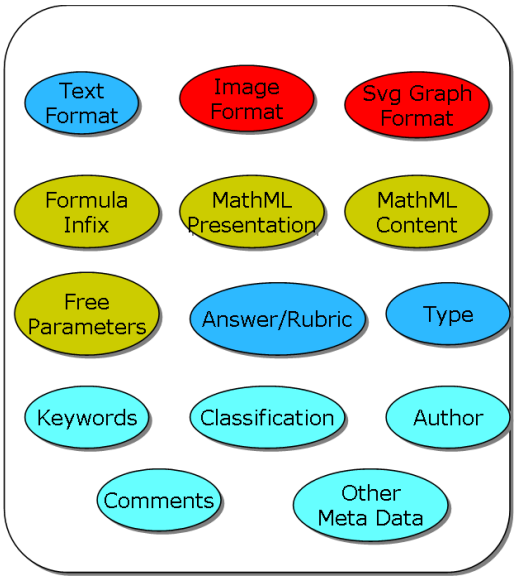


Figure 2.1. Question Contents

Such different question content formats add an extra burden on the DMASEngine to search and then on the process of importing and exporting questions from one place to another. More details on test sharing and search engine will follow in more depth in chapter five.

2.2.4 Test Authoring and Sharing

The system has a *WYSISWYG test authoring tool* to easily enable teachers to author their tests and test questions from scratch or customize the imported ones. Through this tool, a user can create a mathematical formula/expression using the infix mode or using the mathEdit editor [51, 52]. In either way, the *MathML codes* will be created automatically for the formula and presented correctly on the Web (using MathML presentation code) while typing the formula, for easy visualization. All conversions between formula's different formats are done automatically and transparently (hidden)

from the user for simplicity and convenience. A user can also upload images or draw a geometric graph using geoSVG [53] and all graphical contents will be taken care of using the DMAS authoring tool transparently from the user. Unlike TWAS systems, the user is not required to know MathML or SVG at all to use the DMAS authoring tool. Any mathematical assessments created (or imported) of any kinds (text, image, formula, or graph) can be published and shared with others. More details on test sharing, customization, authoring, and publishing will be in chapter five.

2.2.5 Online Test Giving/Taking

One component of DMAS is a capability of online test giving by teachers in a simple way and online test taking by students on the other side. The system supports two types of testing: *online testing* and *paper testing* and two test views: *all-questions-per-page* and *one-question-per-page*. The system supports an accumulative submission of test questions so that in case of loss of power to a laptop, closing browser window by mistake, computer station hanging up, or any other incident, a student can retake the test after authorization from the teacher and from the point where he stopped before the problem occurred. The teacher has the ability to allow/disallow taking of online test at any point of time. It also provides teachers with *useful, real-time, and dynamic statistics of students' performance*. More details on test giving and taking come in depth in chapter six.

2.2.6 Grading and Grades Managements

DMAS system supports an automatic and *instant* online grading of an assessment test. The system automatically grades the multiple choices and true-or-false questions and gives the option to grade text answers either extended (essay) questions or short answers. The system will also grade questions with mathematical formulas or expression and will allow students to check automatically against the correctness of their answers. It provides teachers with real-time grading and results administration. Chapter six has more details about instant and automatic grading and grades managements.

2.2.7 Real-time Online Test Supervision

One main component of DMAS is providing *real-time online test supervision*. TSIM, a *light-weight* test supervising tool, is to allow interactions and communications between a teacher and students in a *class room* privately and securely. The goal here is to promote teaching and learning environment during testing time. TSIM provides such online communications as *instant text-messaging* provided by DMAS system. Through TSIM, the teacher can monitor progress and performance of students from his TCP in *real-time* and *interact privately* with a group of students fro help and educational purposes. The next chapter, chapter three, talks in details about TSIM as a *light-weight* online test supervising tool.

I will conclude this chapter by mentioning the *advantages* of the distributed structure of DMAS system over the TWAS systems. DMAS system represents a *new direction and underlined framework for delivering and sharing mathematical assessment materials* on the Web and importing/exporting questions between teachers from

same/different locations. It is designed as a *distributed system,* so no centralization problems, less traffic congestion, and if one assessment site goes down, other sites work. Unlike TWAS systems, DMAS supports *locality* of sharing mathematical assessments at individual school sites. One of the main goals is supporting *sharing of mathematical assessments*. It supports WYSIWYG authoring tool for teachers and questions are contributed by teachers *dynamically* and not necessarily pre-made. Another advantage is supporting customization including mathematical formulas and geometrical graphs. Also mathematical formulas are represented/computed in infix and Web standard MathML code (presentation and content), and unlike other TWAS systems, DMAS does not require users to know MathML at all. Geometry objects are editable and represented in *text* formats using a standard SVG format. Another key feature of DMAS is providing *real-time online test supervision* through TSIM tool. DMAS is completely Web-based and complied with Web standards. Finally, DMAS supports *interoperability* with other existing online systems. However, DMAS does not have many questions yet but the key contributors are teachers and educational experts themselves and also DMAS needs to have more real trials in schools in the future to determine its effectiveness and have more improvements.

CHAPTER 3

REAL-TIME SUPERVISION OF ONLINE TESTS

In this chapter, I introduce another contribution of this dissertation**,** *TSIM (Teacher-Student Interaction Mechanism).* It is a way of interactions and communications between a teacher and students in a *class room* privately and securely to promote teaching and learning during the test. TSIM provides such online communications as instant text-messaging as a part of DMAS system. Through TSIM, a teacher controls student login system *dynamically and electronically* as an alternative way to the standard login system (Figure 3.1).

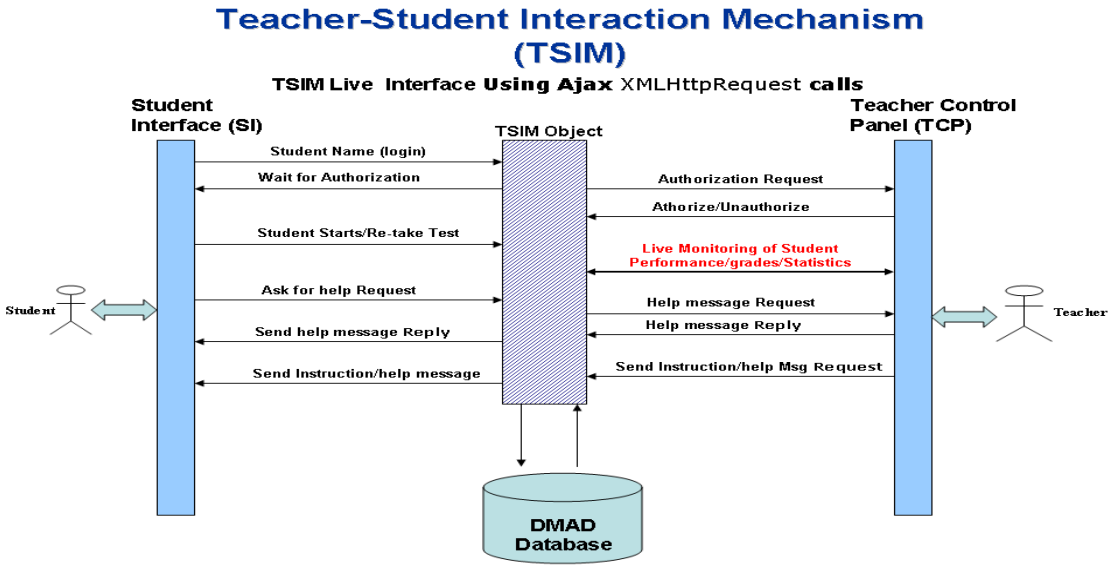


Figure 3.1. TSIM Live Interface

The teacher can also monitor all students and their performance from the teacher TCP in *real-time* and either verbally guide the whole class on certain points or *interact privately* with one or more students via TSIM. In other words, TSIM tool aims to handle online interactions and communications between a teacher and students in a *class room* for educational purposes.

3.1 TSIM Test Authorization Through TCP

In TWAS systems, students login to the online test using standard login system (username/password). However, after a real trial of DMAS system in seventh grade in Kimpton Middle using the standard login system and DMAS login approach of randomly generated and manually distributed test codes (chapter 6), and after consulting teachers and educational experts on this matter, I found out that standard login system using username/password or manual code distribution may not be a good way to use specially for early school grade levels, due to time wasting and difficulty in distributing such codes to students. So I came up with two new approaches which are in simpler forms: First one is that a teacher can *dynamically and electronically distribute codes* to students using *TSIM mechanism*. Second approach is logging in using *TSIM Test Authorization Through TCP*.

1. Login using *Randomly Generated Test Codes and Electronically distributed*: Here are the steps:
    - Students go to the test login page directed by the teacher and waiting for test codes to be distributed.

- A teacher dynamically, through *TSIM teacher-interface*, distributes exam codes to students to take a test electronically (Figure 3.2).
- Then each student can see his assigned *unique code* appears automatically inside a box of his login page to enter to start the exam.
- The system makes code *verification and matching* (Figure 3.3) to make sure this is a *valid test code* and a student is *authorized* to take a test.
- The teacher always has the control over *publishing/hiding* test codes at any time and also has a full control of *enabling/disabling* student(s) from taking test at any moment.

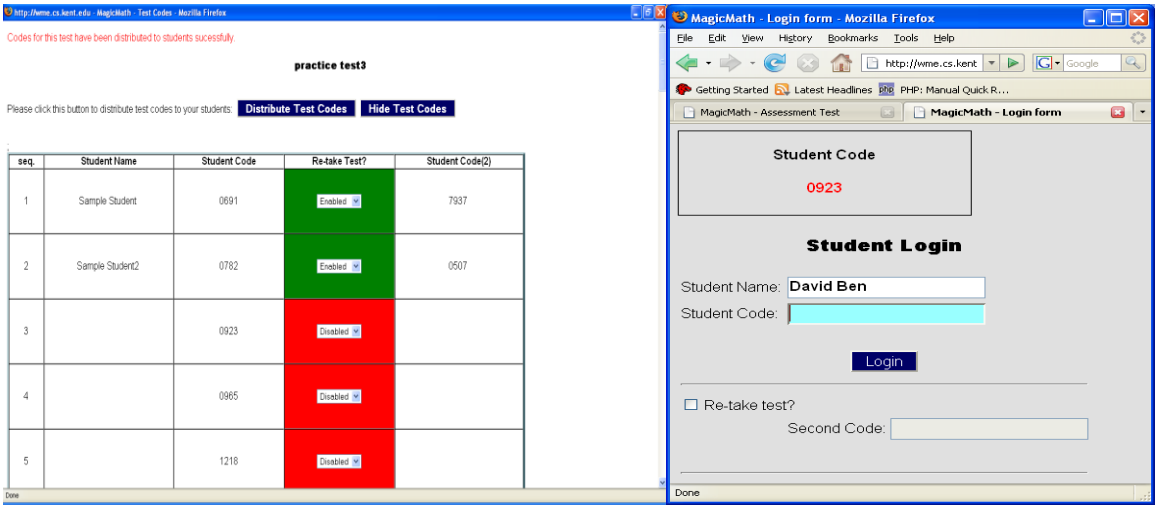


Figure 3.2. Electronic Test Code Distribution Through TSIM

2. Login using *TSIM Test Authorization Through TCP*.

After trying the previous approaches (using standard login and test codes) and after consulting mathematics education experts, I have come to a conclusion that we may not need to use test codes for student login system at all. So I started searching for a

better, simpler, yet more convenient way. I came up with a new approach called *TSIM Test Authorization Through Teacher Control Panel (TCP)* (Figure 6.3).

In this approach, I got rid of using usernames/passwords and test codes completely and came up with another yet simpler approach which saves class time, teacher's effort, and student time too. In such approach *TSIM object gives full control and authorization to a teacher using a simple and secure interface, TCP*. On the other hand, students need only to enter their names to take the test, wait for authorization from the teacher, and nothing else. Now here are the details:

- Students go to the test login page directed by the teacher, enter their names and wait for authorization.
- A message "Please wait for authorization..." will appear for students.
- A teacher dynamically, through *Teacher Control Panel (TCP)*, can see all student names that have been entered and waiting for authorization (Figure 3.3).
- The teacher can choose all (or a group of) students to authorize (or retake) or to ban students from taking the test. A teacher can prevent and delete unwanted/unauthorized names from the system completely.
- Then if authorized, a student will see a message "Now please click the button to go to test" and an enabled button "Start Test" will appear dynamically for the student to proceed to test.

- The teacher always and at any moment of time has a full control over Authorizing, banning, or removing student(s) through the *Teacher Control Panel (TCP)*.

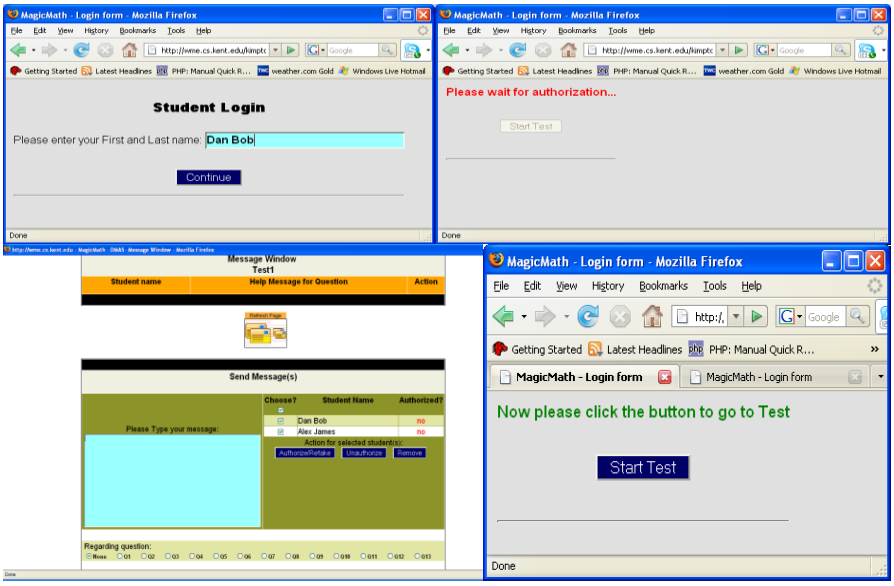


Figure 3.3. Login Using TSIM Object and Teacher Control Panel (TCP)

3.2 TSIM Real-time Monitoring of Student Progress and Performance

Unlike TWAS systems, DMAS system provides teachers with a TSIM tool as *real-time online test supervision*, to allow a teacher to have a live-monitoring of performance of students and the progress they are making. So TSIM tool was created to do three main jobs: First, to be used as a simple login system for students and as an alternative way to the standard login system. Second is to provide live-monitoring of students performance so the teacher knows exactly how students answer questions and the progress that's being made at any moment of time. So the teacher can have better idea of the test status and may act accordingly. The third function of TSIM is to provide new

and secure online communications/interactions as *instant text-messaging* between the teacher (TCP) and students (SI) for help and instruction purposes.

3.3 Interactive Instant Text-Messaging System Using TSIM Object

3.3.1 Traditional Online Testing Environment

The meaning of *traditional testing environment* here is that students login using standard login system (username/password), proceed to take the online test, and at the end grading results come after the test (Figure 3.4). In my previous work [25] in a *traditional testing environment* where students login using username/password and results come after the test and after real trials and consulting mathematics teachers and experts, I found:

1. Some students tend to forget their usernames, passwords, or both especially at the lower grade levels.
2. Vulnerability of using already known usernames and passwords that may not be secured enough.
3. Grading Results and statistics come after the test (at the end of testing environment).

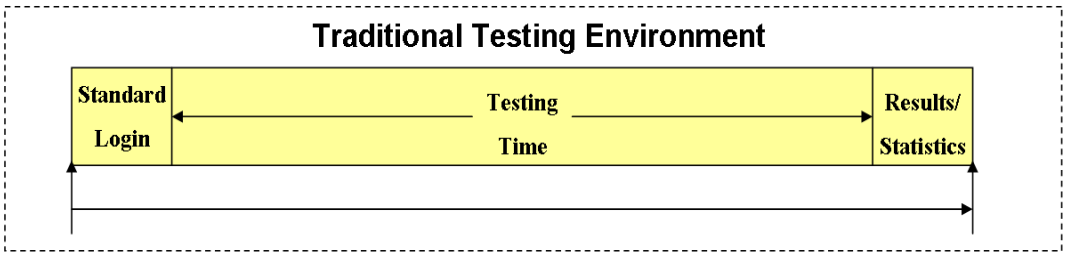


Figure 3.4. Traditional Testing Environment

To avoid such problems in traditional testing environment, it was necessary to find an alternative way to such standard login system, *TSIM Test Authorization Through Teacher Control Panel (TCP),* which was mentioned in (section 3.1). The idea of using TSIM tool here is to minimize student login time, ease the login process, and save more time for testing.

3.3.2 Real-time Online Testing Environment

Another problem with traditional testing environment is that producing grades results and other useful statistics *after* the test. Sometimes it is too late for a teacher after the test to know a problem that students may have during the test. So TSIM tool lets teachers know their student's progress and provide *real-time online test supervision* so he can act or help based on that (section 3.2). In other words, TSIM aims to convert traditional testing environment into real-time testing environment so TSIM can be used as a teaching and learning tool (Figure 3.5). So by using TSIM we have:

`Traditional testing environment → TSIM Tool → Real-time testing environment.`

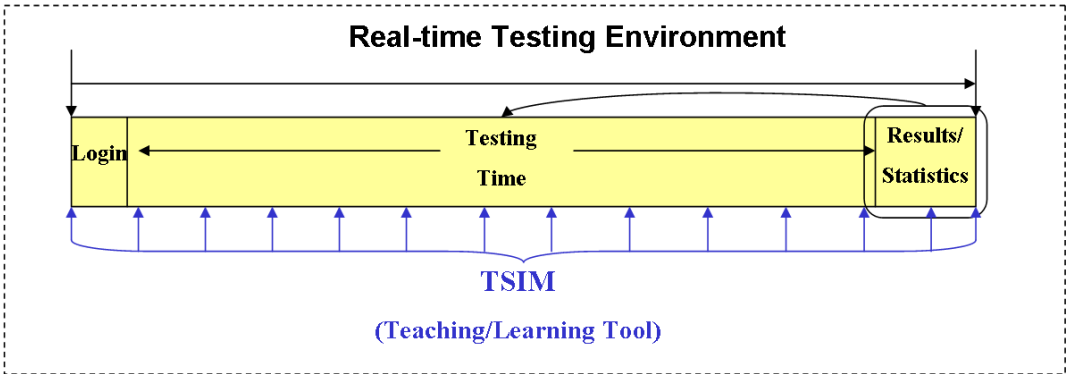


Figure 3.5. Real-time Testing Environment

3.3.3 TSIM As an Interactive Teaching and Learning Tool

TSIM is a built-in and light-weight tool provided by DMAS system so it does not affect the testing process time. The goal of TSIM is to be a useful tool for teaching and learning during testing. TSIM converts a *traditional testing environment* into a *real-time testing environment*, in which a teacher monitors student performance and progress in a test in a real time. Now TSIM goes a step further beyond having a *real-time testing environment*, TSIM can make the testing environment more *interactive* between a teacher and students through *instant-messaging* as away of teaching and learning (Figure 3.6). So here we have the following transformation:

`Real-time testing environment → TSIM Tool → Interactive testing environment.`

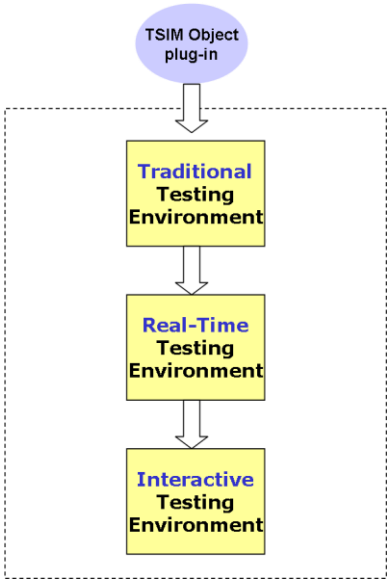


Figure 3.6. TSIM Object As a Teaching Tool

TSIM is a new way of interactions and communications between a teacher and students in a *class room* privately and securely to provide a *real-time and interactive testing environment*. So far through TSIM:

1. A teacher *controls student login system* dynamically and electronically.
2. During testing, TSIM provides the teacher with *Live monitoring of all students and their performance* from the TCP in *real-time*.
3. TSIM provides new and secure online communications/interactions as *instant text-messaging* between the teacher (TCP) and students (SI).

Thus far, we have the following transformation through TSIM tool:

`Traditional testing environment → TSIM Tool → Real-time testing environment.`

And then we have:

`Real-time testing environment → TSIM Tool → Interactive testing environment.`

Messages sent by a teacher to students through TCP are tailored into a *message queue* in a *first come first serve* basis or *first input first output (FIFO)*. Then the *message dispatcher* distributes messages to intended students (Si) and the right locations based on some parameters (messageID, studentID, TestID, qID, locationID, and some other parameters), as we see below (Figure 3.7).

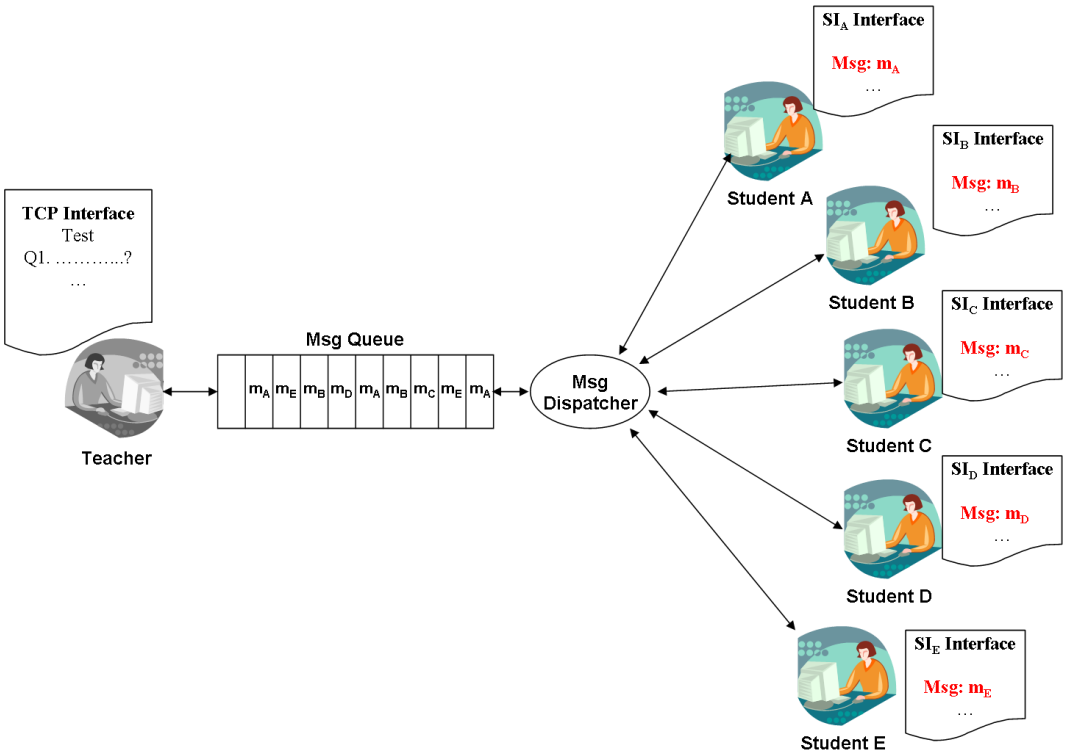


Figure 3.7. TSIM Message Queue and Dispatcher

In an interactive testing environment, TSIM supports different types of help and instructional messages that can be used for teaching and learning in a class room:

1. *General-purpose* message: A message sent by a teacher to all students for general instructions (Figure 3.8 and 3.9). Here is a teacher sends a general message to all students through TSIM object (Figure 3.8).

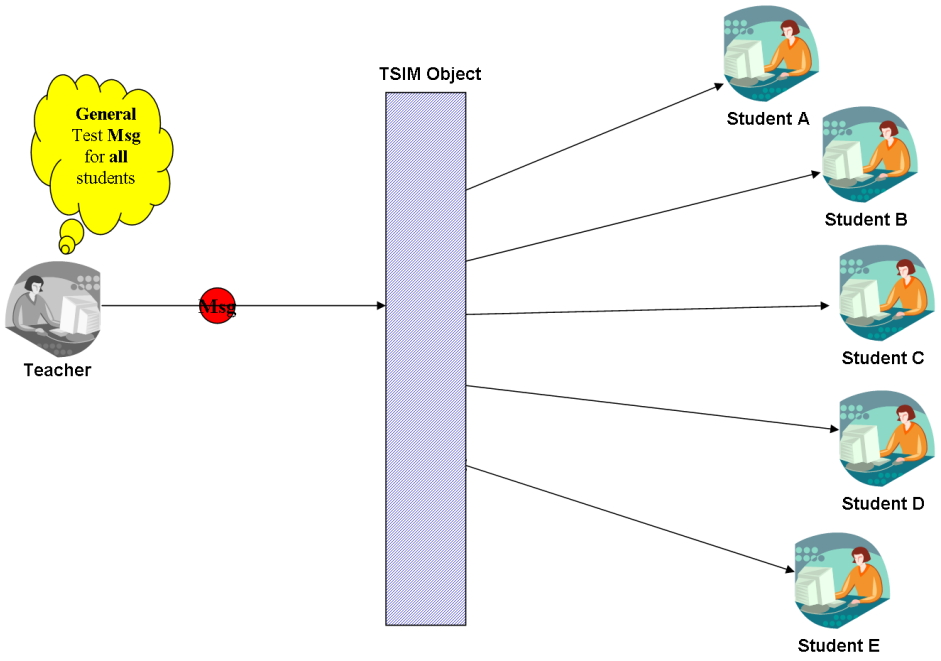


Figure 3.8. TSIM Object General Message (Part 1)

Now, all students receive the message distributed by the *message dispatcher* (Figure 3.9).

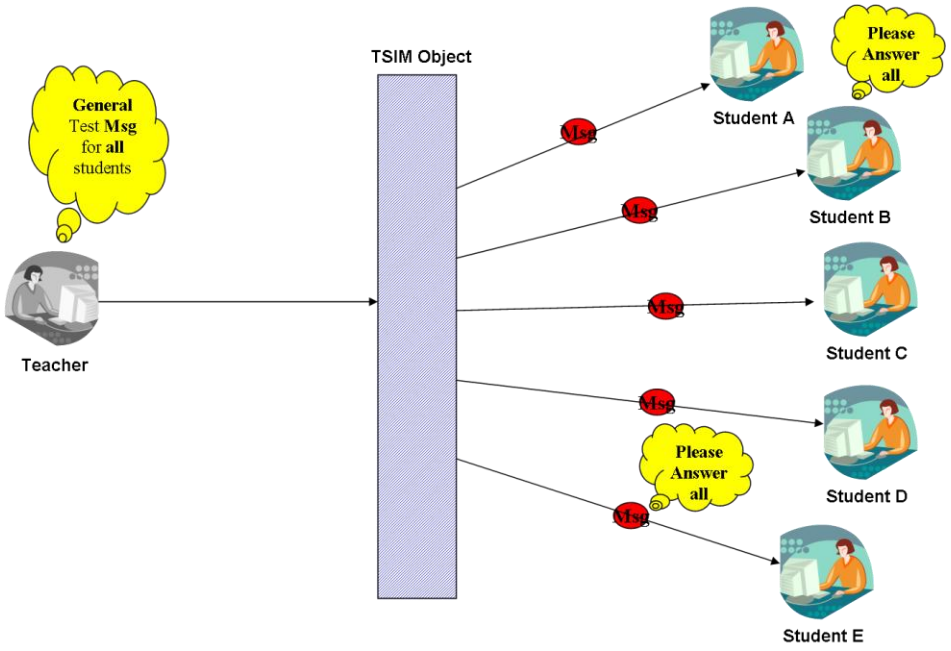


Figure 3.9. TSIM Object General Message (Part 2)

2. *Special-purpose* message: To a group of students only (Figure 3.10 and 3.11).

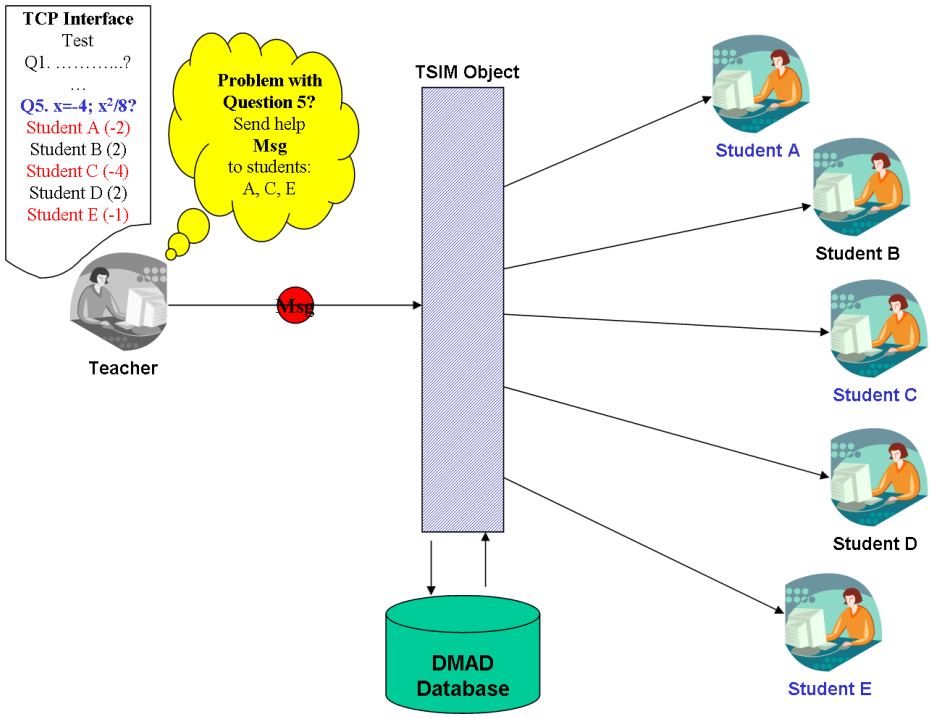


Figure 3.10. TSIM Special Instruction Msg (Part 1)

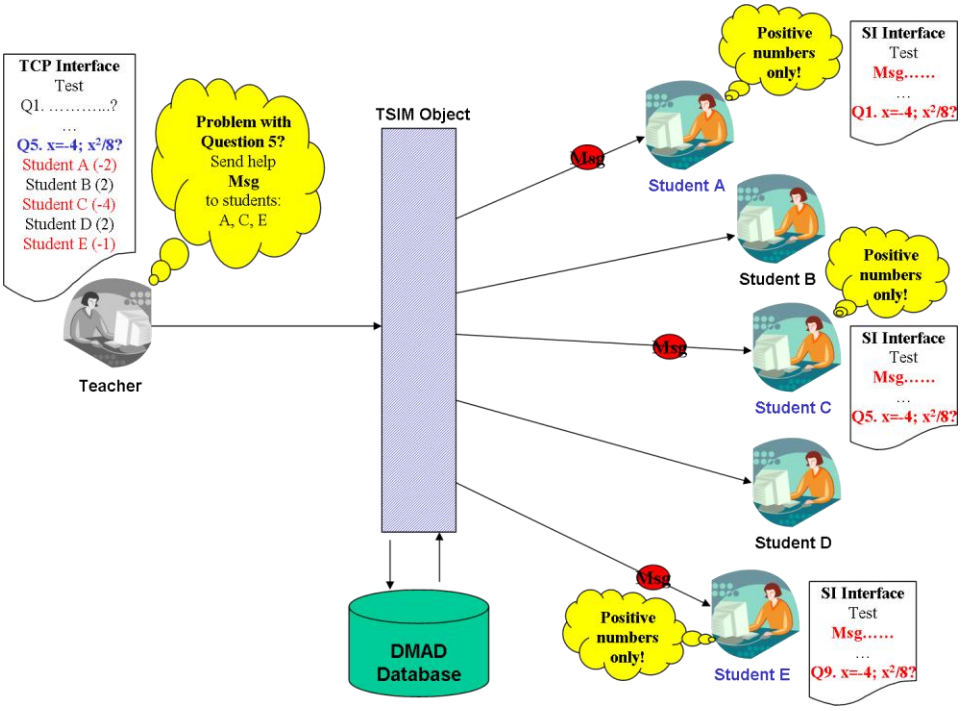


Figure 3.11. TSIM Special Instruction Msg (Part 2)

Here only students (A, C, E) who have problems with the mathematical question (`Q5. x=-4; x2/8?`) receive the message sent by the teacher and distributed through the *message dispatcher* (Figure 3.11).

3. *Ask-for-help* message: If students ask for help from a teacher on a question (Figure 3.12 and 3.13).

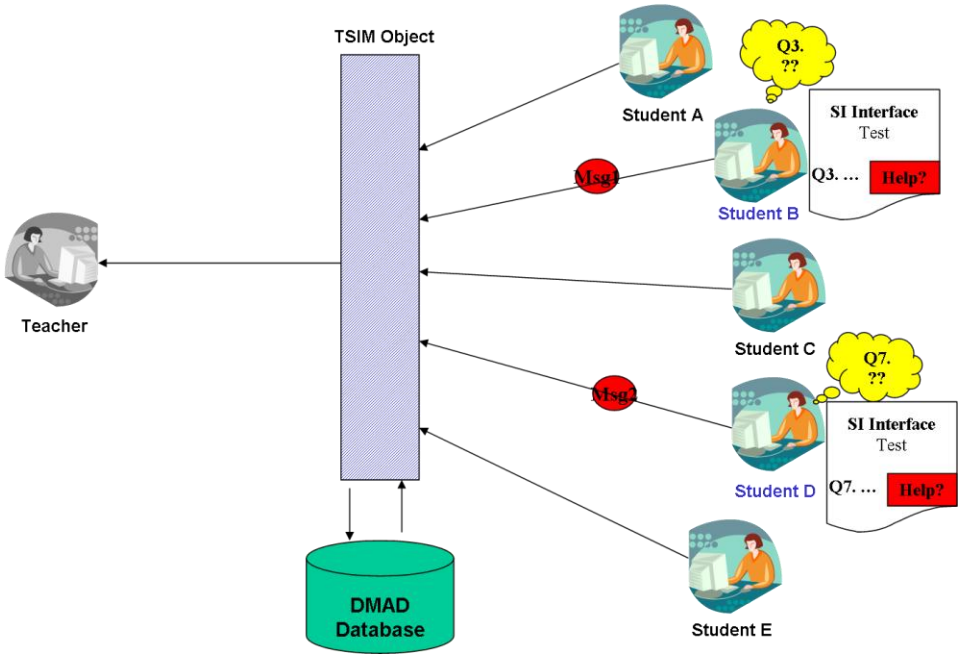


Figure 3.12. TSIM Object Student Ask-For-Help Msg (part 1)

In the above figure (Figure 3.12), two messages sent by students B and D privately asking for help on questions Q3 and Q7 respectively. Then the teacher automatically will know through his TCP that students B and D need help on those questions, so he can send same or different help or other instructional messages back to them (Figure 3.13).

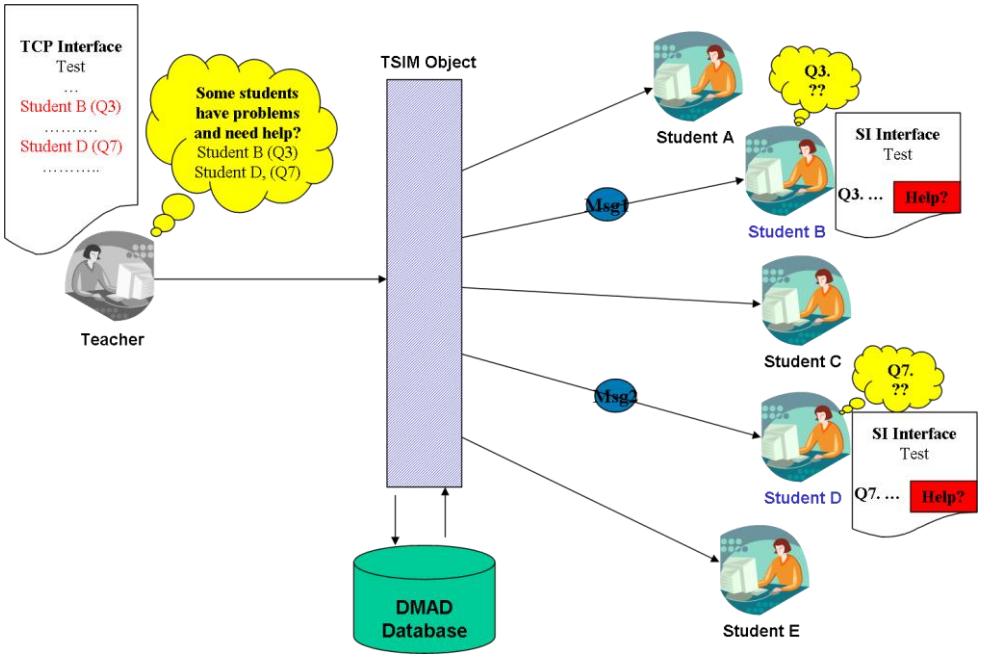


Figure 3.13. TSIM Object Student Ask-For-Help Msg (part 2)

TSIM object design and implementation involve using *AJAX, JavaScript, HTTP requests, databases, and client and server* supports. It is actually an *intermediate layer* of online interactions and communications between a teacher and students in a class room. TSIM uses both AJAX and JavaScript for communications using the JavaScript `XMLHttpRequest` object. Using JavaScript and `XMLHttpRequest` object TSIM can transfer data and information from a teacher to students and vise versa without *refreshing/reloading* the page.

The TSIM uses `ResponseXML` property that returns an XML document object or HTML data, which can be examined and parsed using W3C DOM node tree methods and properties. The TSIM dynamically fetches information from DMAS databases using AJAX technique. The returned data from the server (databases) will be *dynamically*

created and converted into an XML document format. Then TSIM uses the DOM to parse the returned XML document, to send/get the messages and information, and to convert them into well-defined formats to be rendered and displayed properly on the Web interfaces (TCP and IS). Some of the basic reasons for creating such interactive technique, TSIM, are that:

- First, through such real-time technique, a teacher can see *student progress and observe any problems or difficulties that a student might have* regarding a test or a question and then can act accordingly.
- Second, there are students who are quite and shy by nature and do not raise their hands or voices to ask for help or clarification if a problem exists. So through such live and private way of communications, a teacher can recognize such problem and help those students.
- Last, through TSIM, a teacher *does not need to disturb the whole class and stop all students at a moment of time and talk about a problem* that might concern only a very few students while distracting other student's attention in a testing environment where in fact they need to be stayed quite and focused.

Now let's first take a look at the basic requirements of TSIM object and features that TSIM supports, they are:

- ✓ To create a *live and real-time* communication technique in a testing environment.
- ✓ With *secure and private environment* of interactions and communications between a teacher and his students.

- ✓ With the *physical presence of a teacher*.
- ✓ All teacher and students are in a same location (a *class room*).
- ✓ *Saves time and effort* and not wasting class time.
- ✓ *Simple and Easy* to use and not complex.
- ✓ *Built-in and Light-weight tool (doe not need to be downloaded).*
- ✓ *Cross-browsers tool* with standard Web technologies.
- ✓ *Nothing special to download* or worry about to make TSIM work.
- ✓ Supports *different types of messages* such as help and instruction messages.
- ✓ Supports *general and specific messages to send to single, group, or all students*.
- ✓ Supports sending messages regarding *certain question or all test questions*.
- ✓ *Students need not to waste time by writing text messages* but need to stay focused and not distracted.
- ✓ Supports *pop-up message alert at student test page without refreshing the page to get his attention* or read teacher's comments/instructions.
- ✓ Provides a *red box at the top of a student test page containing teacher's comments/instructions*.
- ✓ It also *highlights a specific question background and shows teacher's comments/hints of that question* at exactly same question location for more clarification.

- Through TSIM Object and teacher-interface (TCP): a teacher can use the message window (*Teacher Control Panel, TCP)* to:
    - Send instruction/help messages to his/her students.
    - Watch/Monitor incoming messages from students asking for help.
    - Send general messages to all students or private to one or a small group of students.
    - Support general messages such as instructions for the test or more specific regarding a specific question.
- The teacher has two ways of sending instruction/help messages to students:
    - Through message window: *Teacher Control Panel (TCP)* to send general or private messages.
    - *Live grading administration interface* to send an immediate message to student for hint or clarification.
- Through TSIM Student-Interface (SI), a student can:
    - Ask for help from the teacher privately without raising hand or voice.
    - Just clicking a "help button" beside each question, and her message will be sent to the teacher.
    - No text-messages to write for students to avoid wasting exam time.
    - Just one simple ask-for action and student will be helped.

Now let's take some concrete examples of how such *TSIM* text-messaging system works between a teacher through *TSIM Object and teacher-interface (TCP)* and students through *TSIM Student-Interface (SI)*:

- <u>Example 1:</u> Suppose during a test a teacher wants to send to all students the following message "Please take your time and read questions carefully!" So from the TCP the teacher types in such message, chooses to send to all students, and makes it general message, not regarding a specific problem, but rather some test instruction (Figure 3.14).

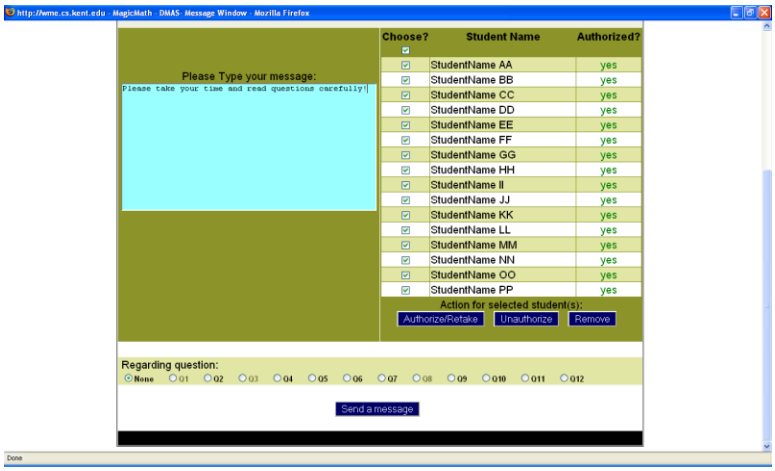


Figure 3.14. A Teacher Sending General Message Through TSIM and TCP

Now after a very short period of time (a few seconds), and from TSIM *Student-Interface (SI)*, every student taking the test will get a copy of that message. As you can see both students (StudentName JJ and StudentName PP) got same message and a small pop-up window with such message appears to get student's attention *without refreshing the page*. The system then highlights the message in a red box at the top of student test

page containing teacher's message and a student can read and delete such messages at any time (Figure 3.15).

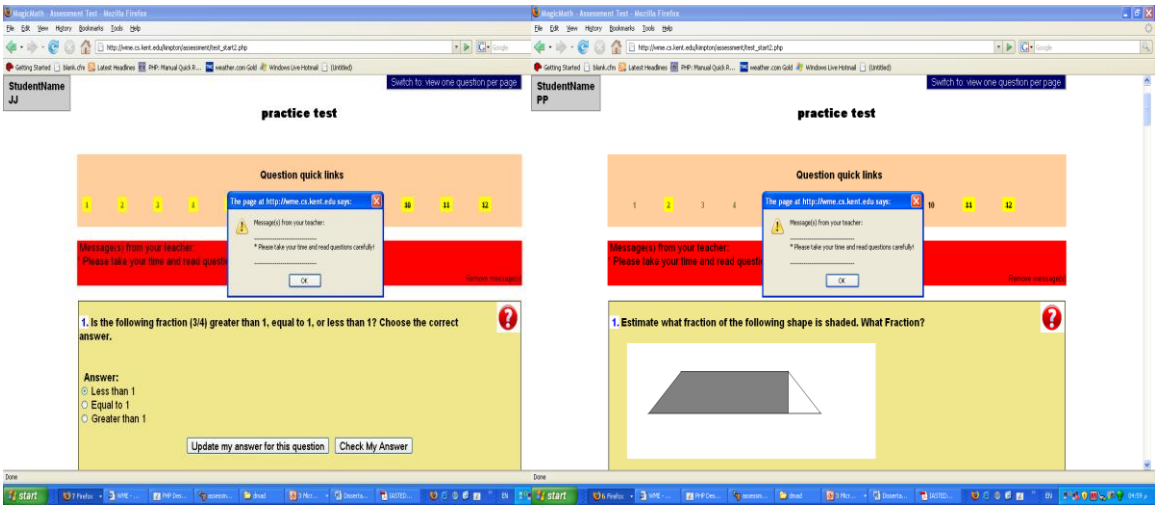


Figure 3.15. A Student Receiving General Message Through TSIM and IS

- <u>Example 2:</u> Suppose during a test a student wants to send a message to the teacher asking for help on a *specific problem or question*. The system provides a "help Button" a long with each test question at SI for a student to just click such button and *help request* message will be sent through TSIM to the teacher and student will be helped! No text-messages to write for students to avoid wasting time of the test (Figure 3.16).

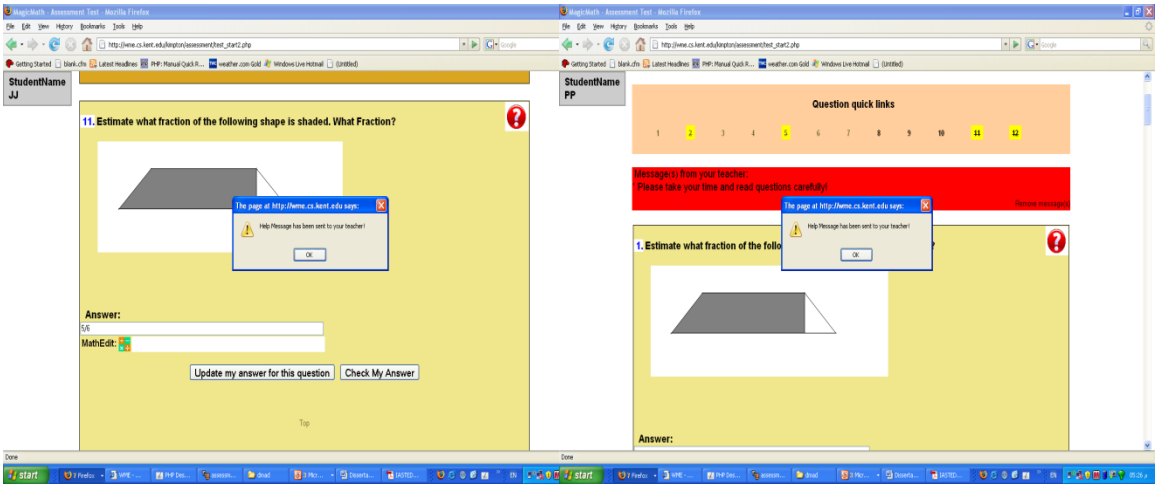


Figure 3.16. A Student Asking for help Through TSIM and IS

- <u>Example 3:</u> From the teacher's interface TCP, the teacher dynamically can see *student names asking for help and specific questions* they need to be helped on. Now assume the teacher wants to send a help message to those students asking for help on *specific problem or question* (StudentName JJ and StudentName PP in this case) (Figure 3.17 – left window). From the TCP the teacher types in a hint/instruction message "Please look at the shaded area first and then compare!", chooses those students (StudentName JJ and StudentName PP in this example), chooses a question/problem and then sends the message (Figure 3.17 – right window).

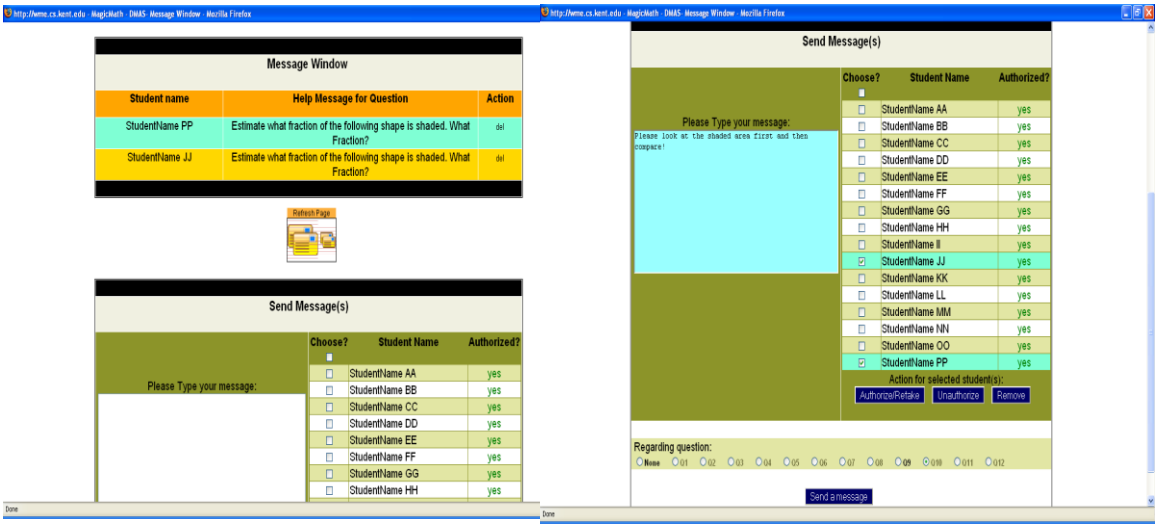


Figure 3.17. A Teacher Sending a Specific Message to a Group of Students

On the other hand, from TSIM Student-Interface (SI), *only* the intended two recipients get the message since it was sent only to them. As we can see only students (StudentName JJ and StudentName PP) got same message and small pop-up windows with the message appear to get student attention *without refreshing the test page*. The system highlights the message in a red box at the top of student test page containing teacher's message, highlights the intended question/problem background, and in the meantime it *shows the message in the exact question location in red, even if students have completely different display order of test questions, to better direct the student and get his attention* (Figure 3.18).

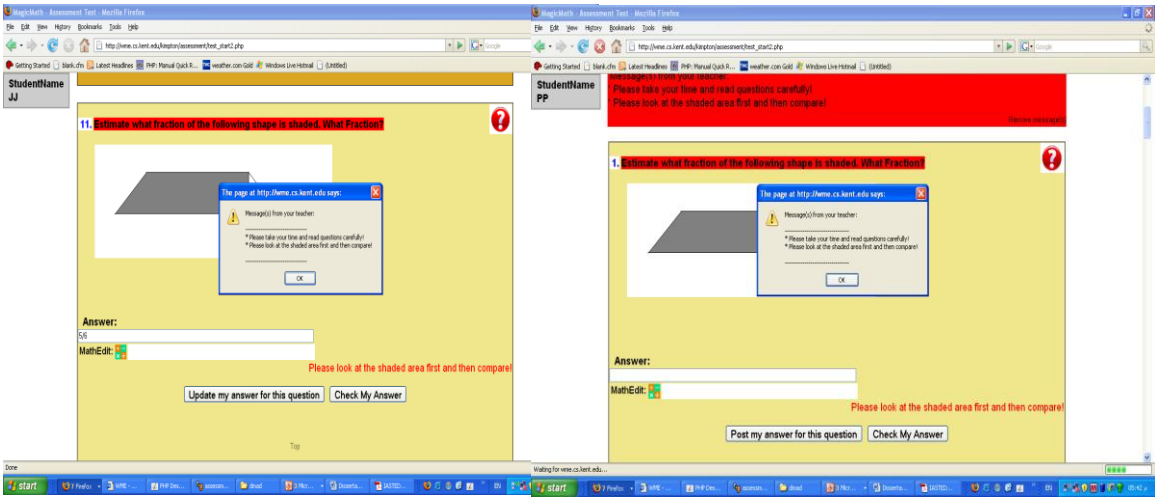


Figure 3.18. Two Students Receiving a Specific Hint for a Question

CHAPTER 4

MATHEMATICAL FORMULAS, FREE PARAMETER QUESTION (FPQ), AUTOMATIC SAMPLE QUESTION GENERATION FEATURE (SQGF), AND GEOMETRY FIGURES

4.1 DMAS and Mathematical Formula Handling

DMAS system supports mathematical expressions/formulas and they can be authored and included as a part of an assessment question. Such formulas/expressions are easily editable and customizable. The system displays mathematical formulas on the Web correctly using *MathML presentation code* and allows mathematical formulas to be entered in an infix mode. In fact, DMAS system allows entering mathematical expressions and formulas in two ways: using infix notations directly, or using WME-Math editor (MathEdit [51, 52]) to enter and edit formulas/equations (Figure 4.1).

- DMAS system uses MathEdit in two modes:
  - ✓ *Teacher Mode*: From the teacher authoring environment, a teacher invokes the the mathematical editor, MathEdit, to enter mathematical expressions and formulas and then includes them the in a question.
  - ✓ *Student Mode*: From the student testing environment, a student can enter their answer for the mathematical expressions in two ways, using MathEdit, or using the infix notation.

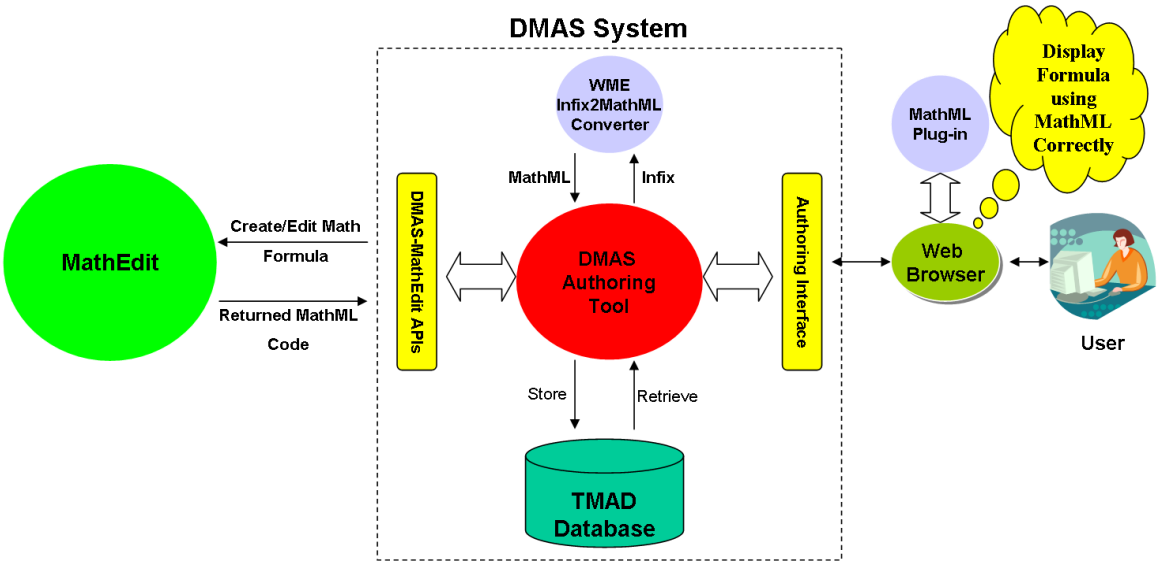


Figure 4.1. DMAS and Math Expressions

- The DMAS system interface invokes the Math editor using MathEdit APIs to author, edit, or get back mathematical expression codes.
- DMAS system interface gets different MathML strings returned by MathEdit:
  - ✓ Infix code.
  - ✓ Presentation Mode.
  - ✓ Content mode.
- DMAS stores MathML strings returned by MathEdit in the DMAS databases for future display, editing, or computation (e.g. answer-checking service) (Figure 4.2).

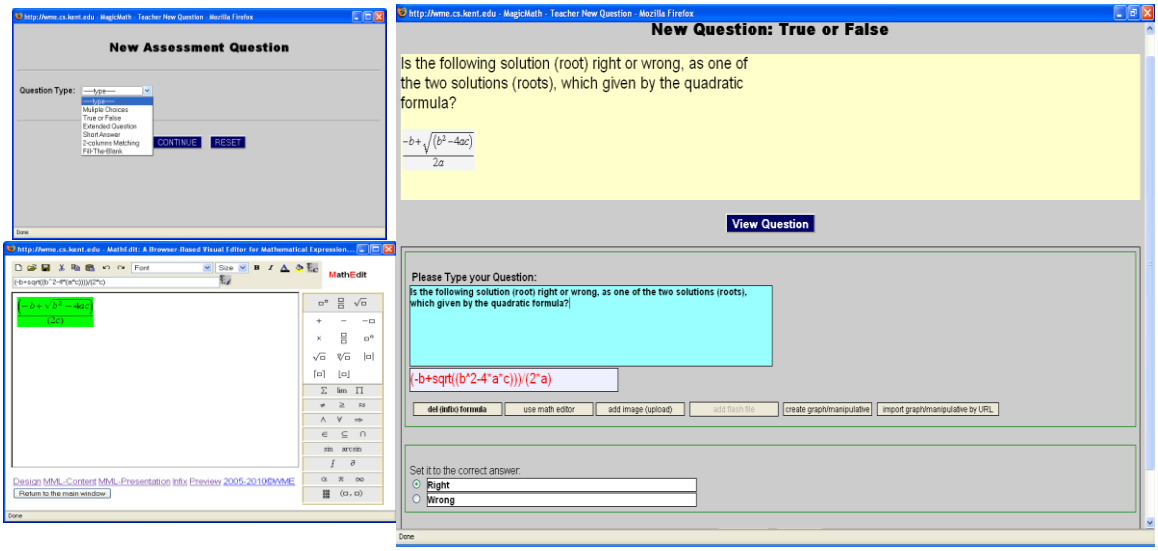


Figure 4.2. Question with Mathematical Formula

4.2 Mathematical Formula, Conditions, and Free Parameter Question (FPQ)

DMAS system supports *dynamic and automatic generation of mathematics formulas/questions using free parameters and based on some mathematical conditions*. This is useful if we need to make multiple instances of same mathematical problems *dynamically* based on some free parameters and given mathematical conditions or criteria. Such feature is a way of having different instances of same question generated dynamically for assessment tests or different students with no extra work needed by a teacher The problem now will be how to fulfill such mathematical conditions (if they can be fulfilled) and then produce the problem? How to store such problems, along with their assigned on-the-fly parameters, with each test or student? More importantly, how do we grade such mathematical problems separately and accordingly (i.e. per student)?

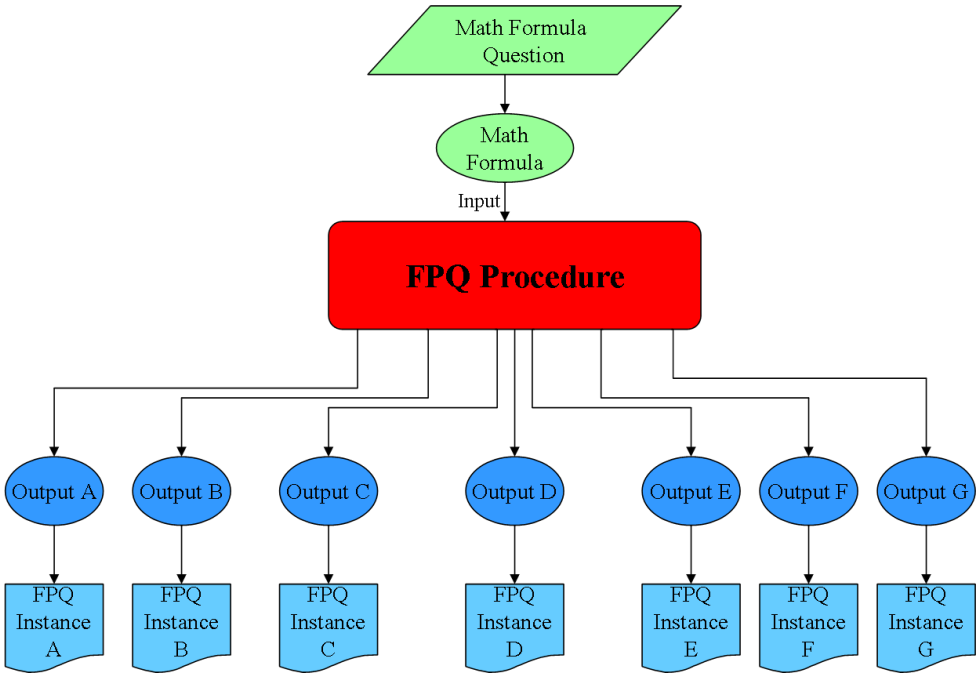


Figure 4.3. Free Parameter Question (FPQ) Concept

The basic concept of *FPQ procedure* is that it takes a mathematical formula as an *input* (extracted from a mathematical formula question), makes some mathematical computations, and then dynamically produces different on-the-fly FPQ instances as *outputs* (Figure 4.3). The usefulness of FPQ question is that from same mathematical question, each student can have a different question with different values which may reduce cheating or plagiarism between students, specially two neighboring students, and promote thinking and working independently rather than copying. Here are some FPQ features:

- A teacher can create one form of mathematical formula/expression question in a test for all students.

- The teacher can use *Advance free parameters* to give "some or all" variables specific values, range of values, or mathematical conditions.
- The system allows mathematical conditions for individual variable to be fulfilled and/or conditions between a group of variables to be satisfied.
- DMAS system will generate *different instances on-the-fly* of same question for each test or student using *random number generator module* that dynamically satisfies such mathematical conditions.
- It supports basic operations of mathematical conditions such as: Greatest Common Factor/Divisor (GCD), Least Common Multiplier (LCM), A<B, C != 0, etc (Figure 4.4).

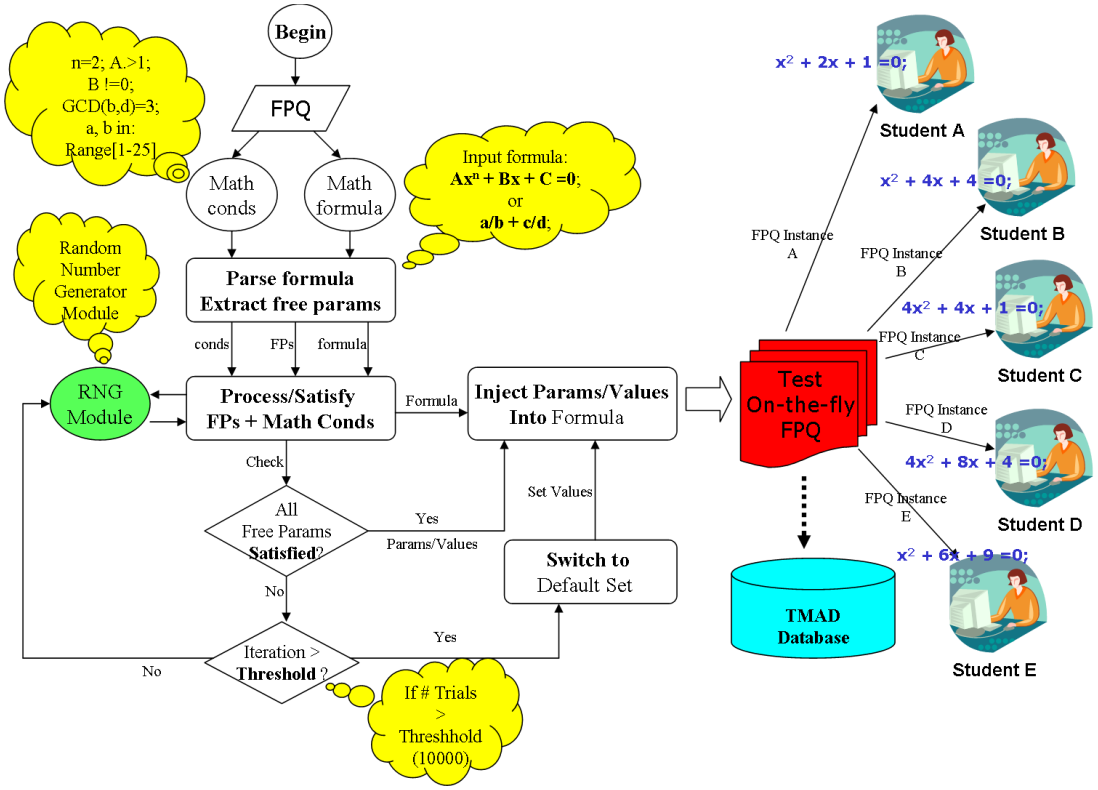


Figure 4.4. FPQ Procedure

Now here are a few examples:

- <u>Example 1:</u> Suppose we want to create a new and simple question that will add two fractions as follows:

  `A/B + C/D ?`

  So a teacher will create such question using DMAS authoring tool (Figure 4.5).

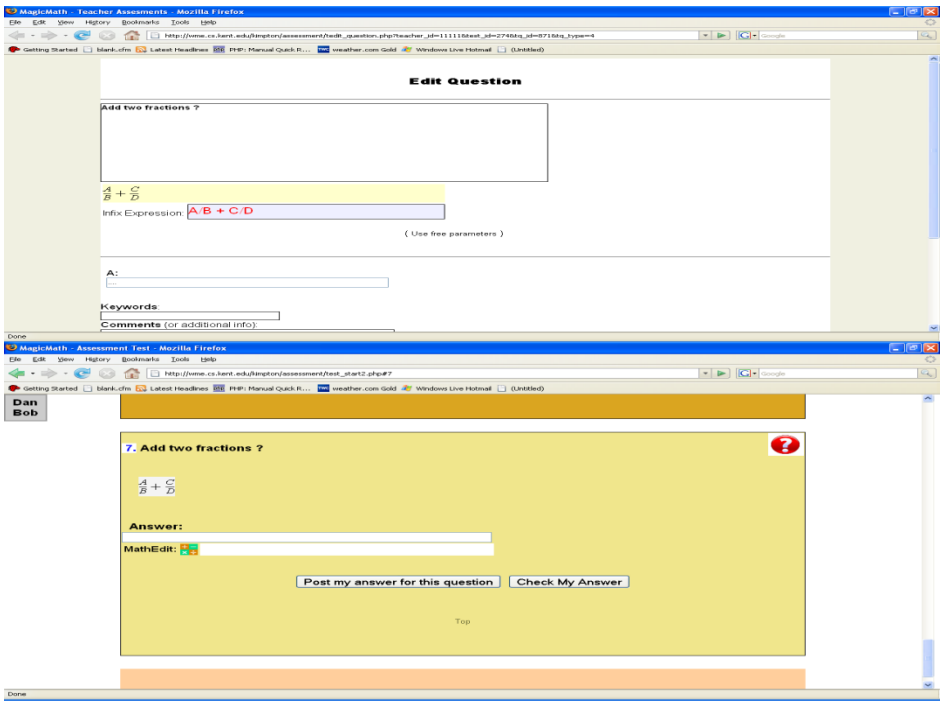


Figure 4.5. Question Views in Authoring and Testing Environments

The teacher has created one form of mathematical formula/expression question in a test for all students. Now and from the *Free Parameter Question (FPQ)* interface (Figure 4.6), the teacher can use "free parameters" option to set parameter names, range of values, and individual mathematical conditions.

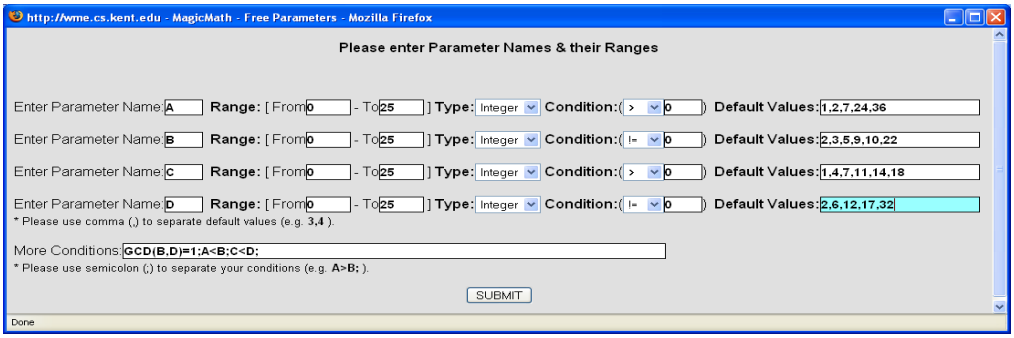


Figure 4.6. Free Parameter Question (FPQ) Interface

Also and as we see from the above window, we can set some other mathematical conditions (*bi-conditions*) between different variables. In our example, we set: `GCD(A,B)=1; GCD(C,D)=1; GCD(B,D)=1; A<B; C<D; A>0; C>0; B!=0; D!=0;`? The `GCD(B,D)=1;` condition is to have two different denominators so their greatest common divisor equals to 1. Other conditions `GCD(A,B)=1; GCD(C,D)=1; A<B; C<D;` are to make sure we have proper fractions and the greatest common divisor conditions are to have simplified fractions. Now after fulfilling the mathematical conditions and creating some random numbers, the DMAS authoring tool *automatically generates the corresponding MathML presentation code on-the-fly for the mathematical formula* so it can be rendered and displayed correctly on the Web (Table 4.1) and (Table 4.2).

```
<math  xmlns='http://www.w3.org/1998/Math/MathML'>
<mrow>
<mfrac>
<mn>2</mn><mn>11</mn>
</mfrac>
<mo>+</mo>
<mfrac>
<mn>4</mn><mn>19</mn>
</mfrac>
</mrow>
</math>
```

Table 4.1. First Fraction's Automatically Generated MathML Presentation Code

```
<math  xmlns='http://www.w3.org/1998/Math/MathML'>
<mrow>
<mfrac>
<mn>8</mn><mn>19</mn>
</mfrac>
<mo>+</mo>
<mfrac>
<mn>5</mn><mn>12</mn>
</mfrac>
</mrow>
                                        </math>
```

Table 4.2. Second Fraction's Automatically Generated MathML Presentation Code

Now here are two different student views of this question in a test (Figure 4.7).

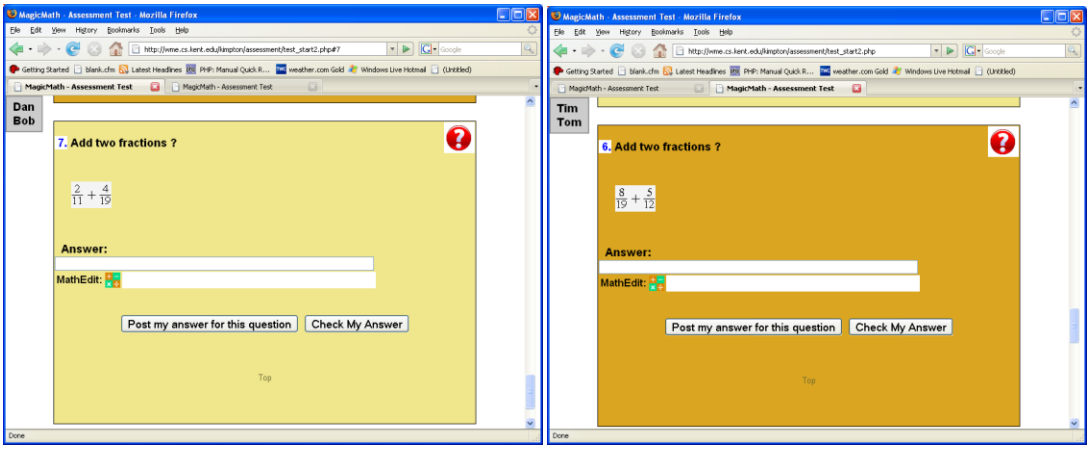


Figure 4.7. Two Different Student Views of a Test with FPQ

As we can see, each student has a different view of a question with different parameter values generated automatically on-the-fly after fulfilling the given mathematical conditions. In our example, we see how the teacher can create more than one instance of same mathematical problem with no further work. DMAS system generates *different instances on-the-fly* of same question for each test or for each student using *random number generator module* that dynamically satisfies such mathematical conditions.

- <u>Example 2:</u> If we take our example (example 1) as it is:

  `A/B + C/D?`

  But mathematical conditions this time will be slightly different:

`GCD(A,B)=1; GCD(C,D)=1; GCD(B,D)=3; A<B; C<D; A>0; C>0;`

`B!=0; D!=0;?`

Now let's take a look into two different student views of this question in a test (Figure 4.8).

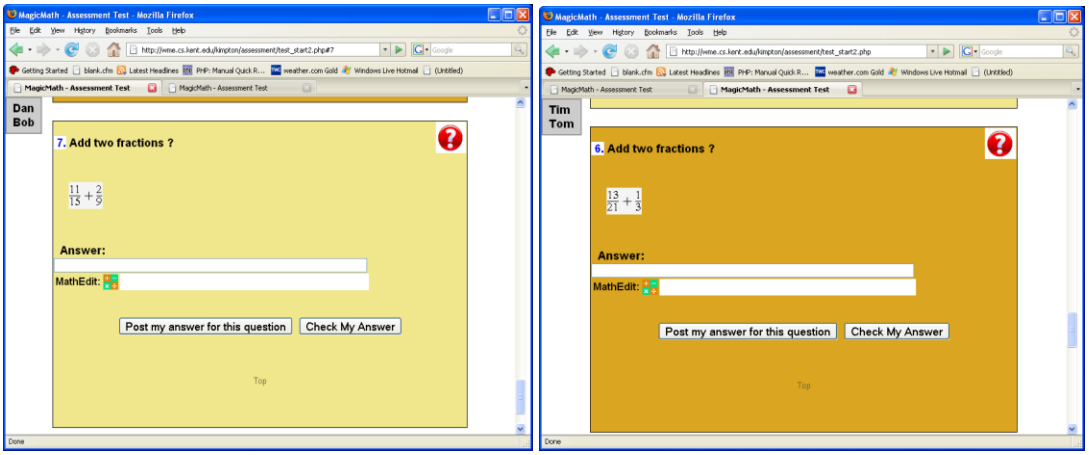


Figure 4.8. Two Different Student Views of a Question with FPQ

- <u>Example 3:</u> Suppose we want to create a question of a quadratic equation:

  $Ax^2 + Bx + C = 0$?

  Using DMAS authoring tool (Figure 4.9).

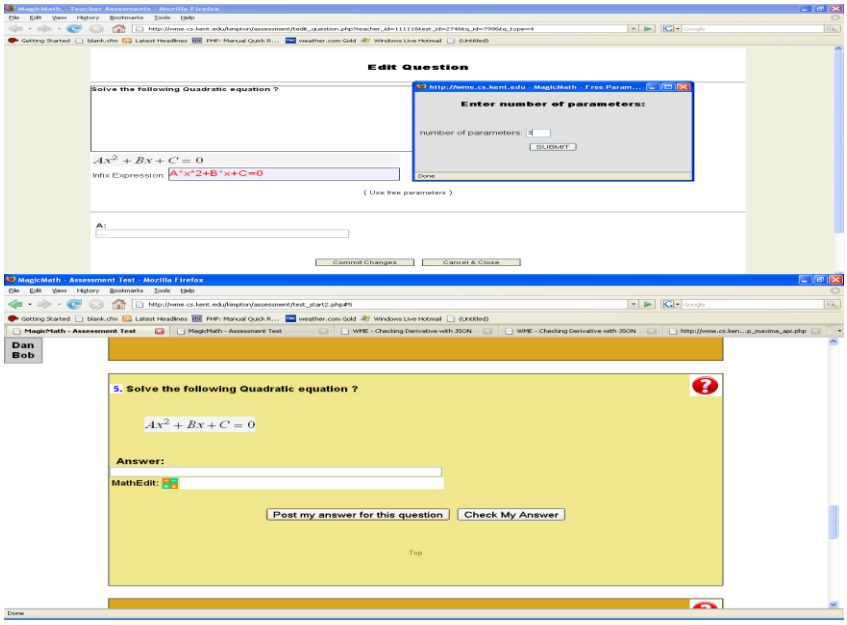


Figure 4.9. Question with Quadratic Equation

Then for the quadratic equation, we will not use coefficient value substitutions because we might end up with some problems that may not have rational root (rational numbers). Instead, we can use *multiplying polynomials* to construct a quadratic equation. So the quadratic equation will be formed from *multiplying the two factors* of the following form `(aX+b)(cX+d)`, for some numbers `a, b, c, and d`. Then the FPQ procedure will substitutes values for such numbers to form a quadratic equation. Now going back to our example, the system will generate some random values:

`a=1; c=1; b=-1; d=2;`

Now the system can generate one instance of the quadratic equation for one student:

$x^2 + x - 2 = 0$,

Also the system can generate some other random values:

```
a=1; c=1; b=-1; d=3;
```

And we will have another instance for same equation for another student:

$x^2 + 2x - 3 = 0$

Now and as we saw in the above examples, there are many problems involved in *Free Parameter Question (FPQ)* questions and tasks that the DMAS system has to perform through *FPQ procedure* such as: creating random numbers for parameter values that fulfill given mathematical conditions. What if FPQ could not satisfy the given mathematical conditions? Then what to do? (The answer to that is using *default value set*). Another task is that *FPQ procedure* parses mathematical formula and then injects free parameters correctly. Then it generates the *MathML presentation code* to display the generated/updated mathematical formula correctly on the Web. Moreover, DMAS system has to store such parameter values along with each individual student answers for automatic grading, because each student may have different form of the same mathematical formula question.

If you notice from the Free Parameter Question (FPQ) interface (Figure 4.6), the teacher has the option to give some *default values* for the free parameters to be used instead of using *random number generator module*. If being used, the DMAS system will pick some given values from the default value sets and based on that, different instances of a question will be generated for students. There are two useful ways of using *default values*: one is that by using FPQ but some students might get "harder" questions than others. So picking values provided by the teacher will make sure students get similar questions. Second, if the system using FPQ could not satisfy mathematical conditions to

generate random numbers after some threshold (say 10,000 trials or a few seconds), then the system can switch and use default values instead.

In any case, DMAS system displays such dynamically generated mathematical formula question precisely cross-browsers using a valid MathML presentation code. It also *provides the Free Parameter Question (FPQ) and default values as useful features for authors but also they can be edited or disabled at any time*. Moreover, assigned values of mathematical formula questions given by *random number generator module* or by *default value set* are stored along with student answer individually. So that the system can grade such questions based on the individual student values generated.

4.3 Automatic Sample Question Generation Feature (SQGF)

In the previous section (section 4.2) I have introduced the Free Parameter Question (FPQ) feature in details by showing problems involved, giving examples, suggesting solutions, and then implementing such solutions. In fact, DMAS system provides Free Parameter Question (FPQ) feature as an advanced feature in which the author himself (the teacher in this case) needs to supply various parameter names, ranges, mathematical conditions, etc. to generate such questions. However some authors (teachers) may prefer to have such feature but they do not have the time, ability, or interest in doing further work such as setting FPQ parameters.

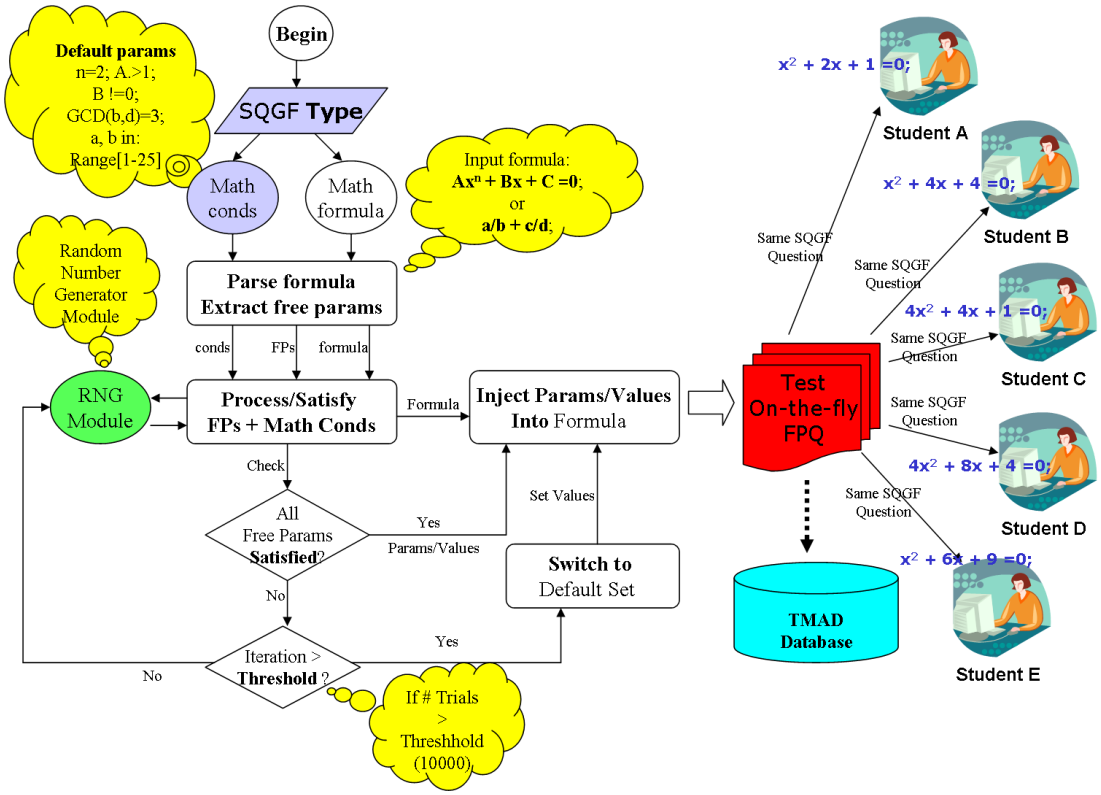


Figure 4.10. SQGF Procedure

Luckily, DMAS system supports an automatic and simpler form of Free Parameter Question (FPQ) feature which is automatic *Sample Question Generation Feature (SQGF)* for teachers to create and import questions into their tests (Figure 4.10). This means the system itself, instead of a teacher, provides a set of pre-made free parameter questions. It enables teachers to include questions generated by FPQ in their test directly. The Sample Question Generation Feature (SQGF) has *default built-in* questions with satisfied parameter names, ranges, and mathematical conditions. So that authors can choose such mathematical question with no further efforts (Figure 4.11), as simple as browse and click!

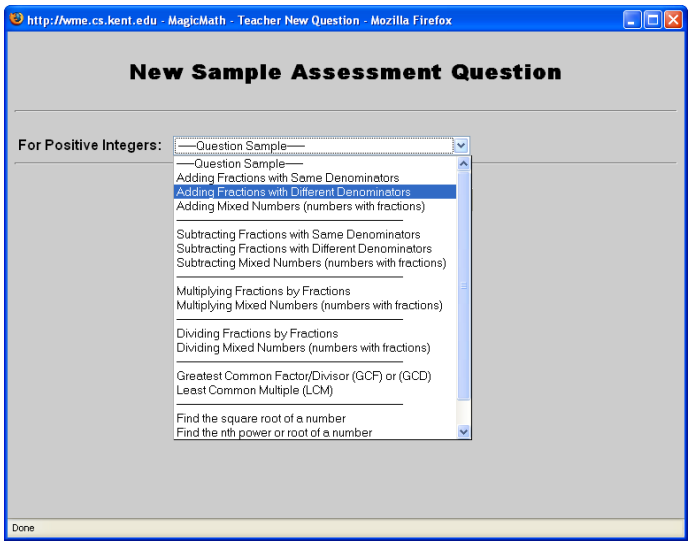


Figure 4.11. Sample Question Generation Feature (SQGF) Form

- <u>Example:</u> If we choose "Adding Fractions with Different Denominators" to create a new question that will add two fractions having Different Denominators, the system will create a new question automatically as follows (Figure 4.12):

  Q. Add the following fractions with Different Denominators?

  `1/4 + 15/22`?

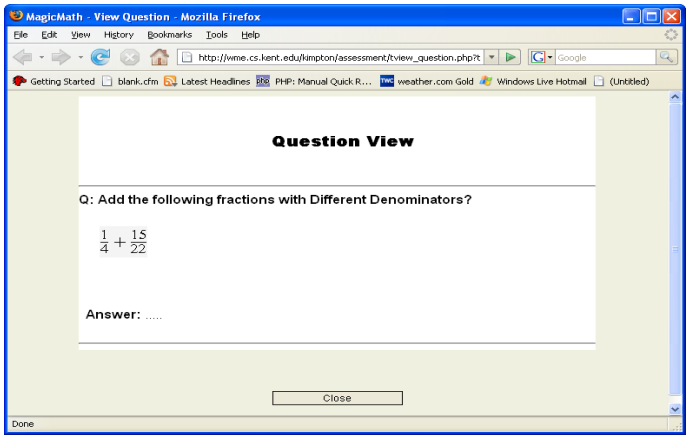


Figure 4.12. Sample Fraction Question Generated

Many question types can be generated by the *Sample Question Generation Feature (SQGF)* and can be included directly into the test being created (Figure 4.13). All parameter values generated here by the SQGF are fixed. This means *all students will see exactly same questions with same parameter values*. However, any question generated by the SQGF is also editable, can be published or exported to other teachers, and can be imported into other tests. DMAS system simply treats questions generated by SQGF as any other questions.

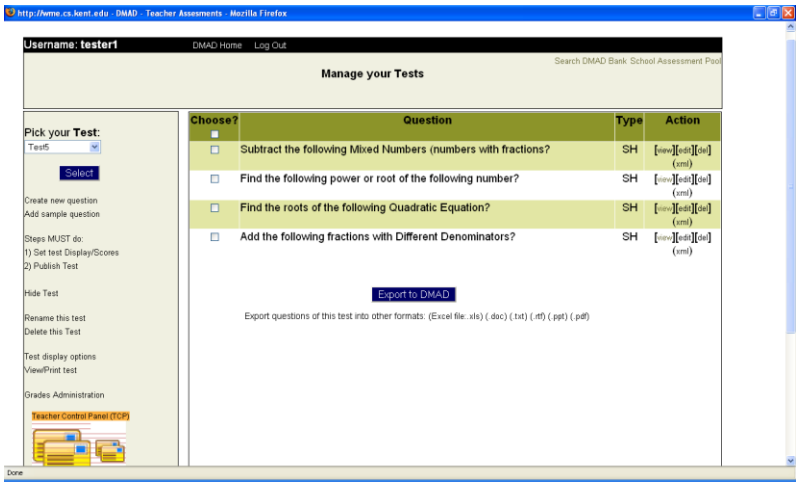


Figure 4.13. Sample Test Created by SQGF

DMAS system provides a set of pre-made free parameter questions generated by *Sample Question Generation Feature (SQGF)* with satisfied built-in parameter names, ranges, and mathematical conditions. Such questions can be included in assessment tests easily and directly. DMAS system supports the following sample questions: *Adding Fractions with Same Denominators, Adding Fractions with Different Denominators, Adding Mixed Numbers (numbers with fractions), Subtracting Fractions with Same Denominators, Subtracting Fractions with Different Denominators, Subtracting Mixed*

Numbers (numbers with fractions), Multiplying Fractions by Fractions, Multiplying Mixed Numbers (numbers with fractions), Dividing Fractions by Fractions, Dividing Mixed Numbers (numbers with fractions), Greatest Common Factor/Divisor (GCF) or (GCD), Least Common Multiple (LCM), Find the square root of a number, Find the nth power or root of a number, and Quadratic Equation.

To conclude this section, so far DMAS system has three different ways of authoring and creating new questions (Figure 4.16):

1. Through *DMAS Authoring Tool*, to create brand new questions.
2. By *searching and importing questions* from DMAD bank.
3. And by using *Free Parameter Question (FPQ) or Sample Question Generation Feature (SQGF)*.

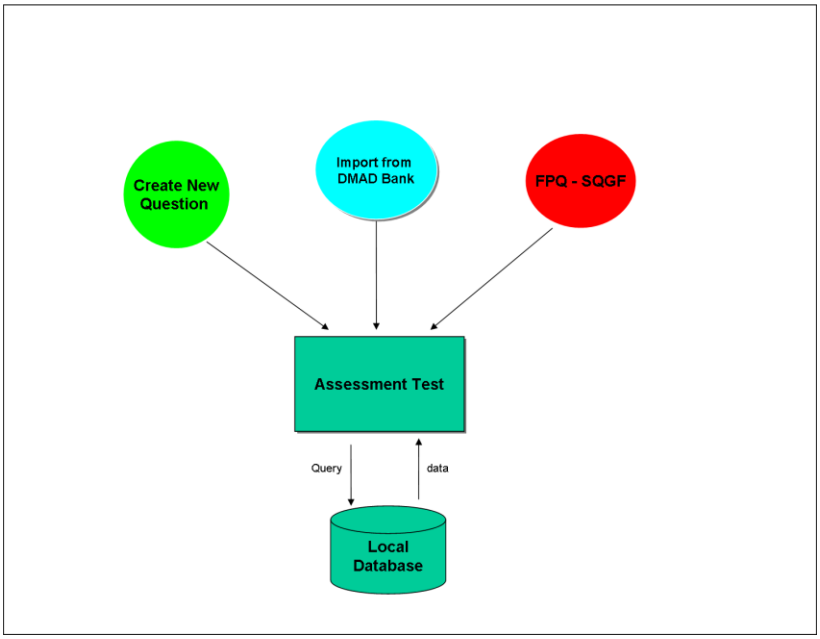


Figure 4.14. Three Different Ways of Authoring Question

4.4 DMAS and Geometrical Graphing

DMAS system supports geometrical graphs in assessment questions such as graphical images and geometrical SVG formats (Scalable Vector Graph). DMAS has an *interface that can inter-communicate with other external geometric editors* such as WME-geoSVG (SVG Web-based authoring tool for geometry) [53]. Through this interface, students and teachers can trigger the geoSVG editor to create and draw geometrical graphs and then the interface can take care of including such graphs in specified questions. All interactions and communications between DMAS system and geoSVG are transparent and hidden from the users (Figure 4.15).

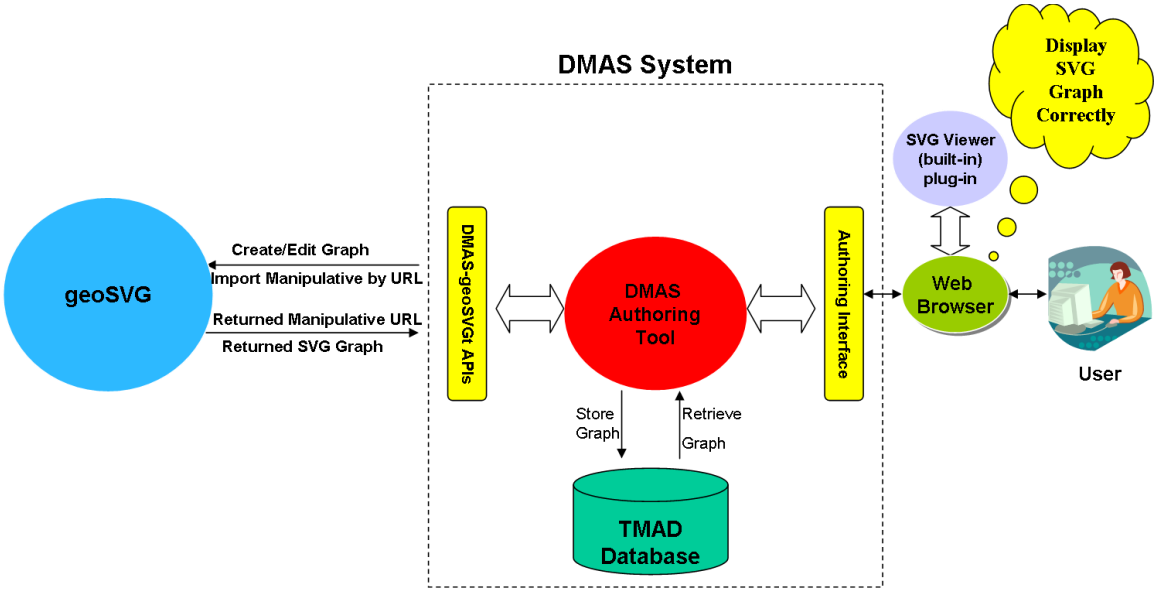


Figure 4.15. DMAS and Geometrical Graphing

DMAS authoring tool enables teachers to draw their own geometrical objects through DMAS interface and geoSVG API interface. It also allows teachers to obtain and

import SVG geometrical objects and manipulative created by other people by their URL address through DMAS interface and the given geoSVG APIs (Figure 4.16). DMAS system stores SVG graph as a text which is very small in size comparing to other image formats such as GIF or JPEG. DMAS uses the SVG formats because *geometrical graph created is just text and small in size*. SVG is also natively supported by some browsers such as Mozilla Firefox with built-in SVG viewer and SVG viewer plug-in is also easily available for free to download for other browsers such as Internet explorer (ie).

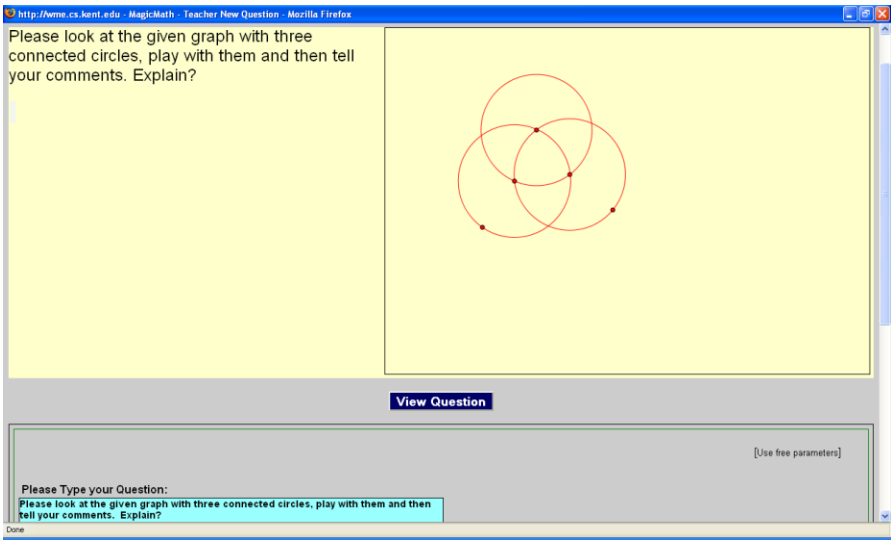


Figure 4.16. Question with Geometric Graph

CHAPTER 5

TEST SHARING, CUSTOMIZATION, AUTHORING, AND PUBLISHING

5.1 Searching for Mathematical Assessments and Test Sharing

The DMAS system allows authors not only to easily create various types of assessment questions from scratch but also use the DMAS search engine, DMASEngine, to search for mathematical assessments locally in school or globally in DMAD bank and other school Web sites from all participating schools. Users then can import copies of the mathematical assessments locally and make customizations. Mathematical assessments can have different contents such as text, mathematical formula, MathML codes, geometrical graph, images, and other data (Figure 2.3), which add an extra burden on the DMASEngine and complicate the process of importing and exporting the mathematical assessments. That's because in case of test sharing (importing and exporting the mathematical assessments), *all question contents and components need to be fetched from the source, copied, migrated from one place to another, and then stored in the destination location* in such distributed system, DMAS.

All searching in other locations, importing, and exporting processes are completely transparent to users, so searching locally or globally looks the same from user's perspective. Authors can import such mathematical assessments from DMAD bank

into their assessment tests. If imported, all such useful materials can be edited and customized easily by the teacher so they can be included in assessment tests. Any created or imported mathematical assessments can be exported to the DMAD bank so they can be *shared* with other teachers from same or different school. *Advanced Search* Also can be used to narrow the search by subject/ category, topic, grade-level, question type, keyword, and name of the author. Searching for multiple options from same question category is allowed (Figure 5.1).

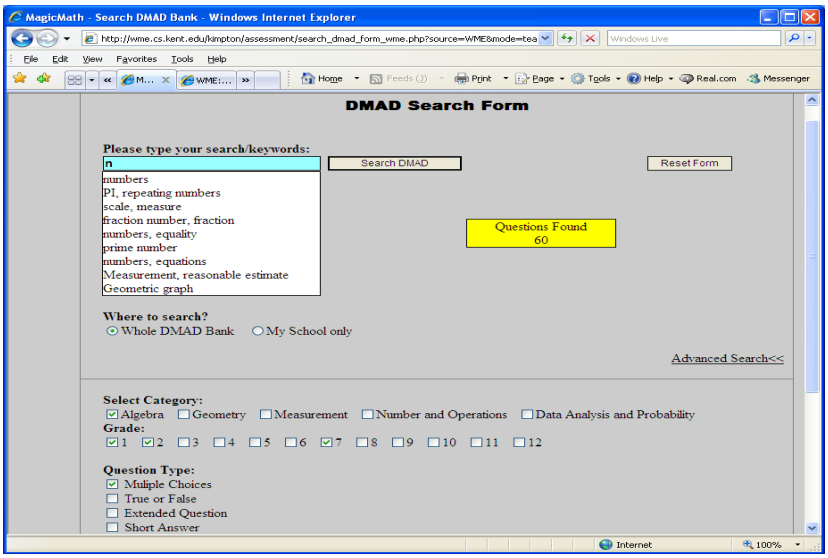


Figure 5.1. DMASEngine Page

DMASEngine uses *AJAX technique and programming* to get better, quicker, and more efficient search results and more interactive web applications. AJAX stands for ***A****synchronous* ***J****avaScript* ***A****nd* ***X****ML*. DMASEngine uses asynchronous data transfer (HTTP requests) between the browser (client) and the web server, allowing web pages to request small part of information from the server instead of reloading or refreshing the whole page.

AJAX is a new way of programming and was published in 2005 by Google. It is not actually a new programming language, but rather it is a way of using existing standards and an architecture for creating better, faster, and more user-friendly web applications. AJAX is based on JavaScript and HTTP requests and considered to be as a *middleware or an intermediate layer between client and server*. DMASEngine uses both AJAX and JavaScript to communicate directly with the server, using the JavaScript `XMLHttpRequest` object. Using this object and JavaScript, DMASEngine can transfer data from/to a web server without even reloading the page.

The DMASEngine uses `ResponseXML` property that returns an XML document object or HTML data, which can be examined and parsed using W3C DOM node tree methods and properties. The *DMASEngine dynamically fetches information from DMAS bank databases using AJAX technique*. The returned data from the server (databases) will be *dynamically created and converted into an XML document format*. Then, DMASEngine uses DOM to parse the returned XML document. It extracts mathematical assessments, values, and other information so they can be rendered and displayed properly on the Web (Figure 5.2).

5.2 Test Customization

After searching DMAS and importing mathematical assessments, all the imported mathematical assessments are fully customizable. All different contents such as text, mathematical formulas, images, and geometrical graphs can be edited, changed, and included in the assessment tests directly after customization. More specifically, if a mathematical formula changes, the DMAS authoring tool will generate the corresponding

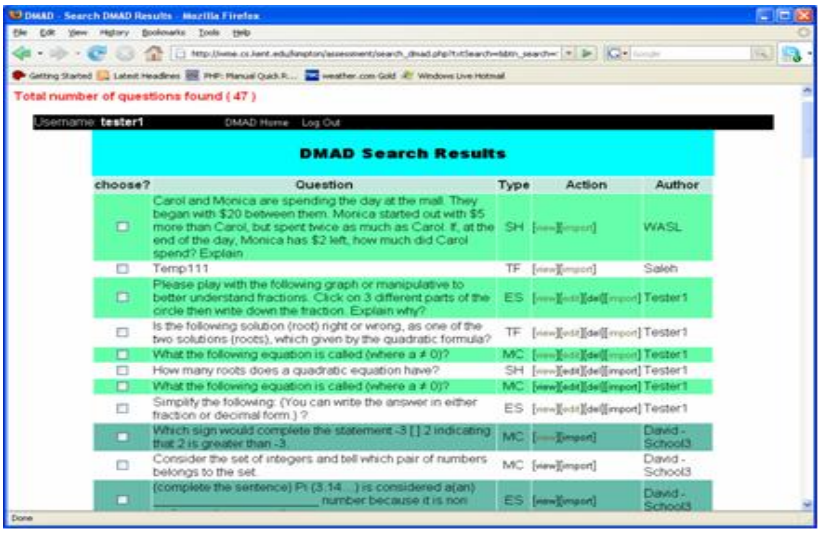


Figure 5.2. DMASEngine Search Results

MathML code dynamically for the new formula. Likewise, if the SVG geometric graph is customized and changed, the corresponding SVG text code will be changed automatically. At any time, authors may customize or delete their questions in the local bank, in addition to other options available for users such as viewing and importing (Figure 5.3).

5.3 Test Authoring and Problem Types

One of the features and basic components of the DMAS system is offering an *authoring tool* for teachers to construct assessment tests and questions. Questions and tests can include text, geometric graphs, images, or mathematical formulas. Teachers can create new questions, view, customized, and delete existing ones. There are different types of questions that DMAS system can support: true-false, multiple-choices, short-answer, essay (extended answer), two-columns matching, and fill-the-blank (Figure 5.4).

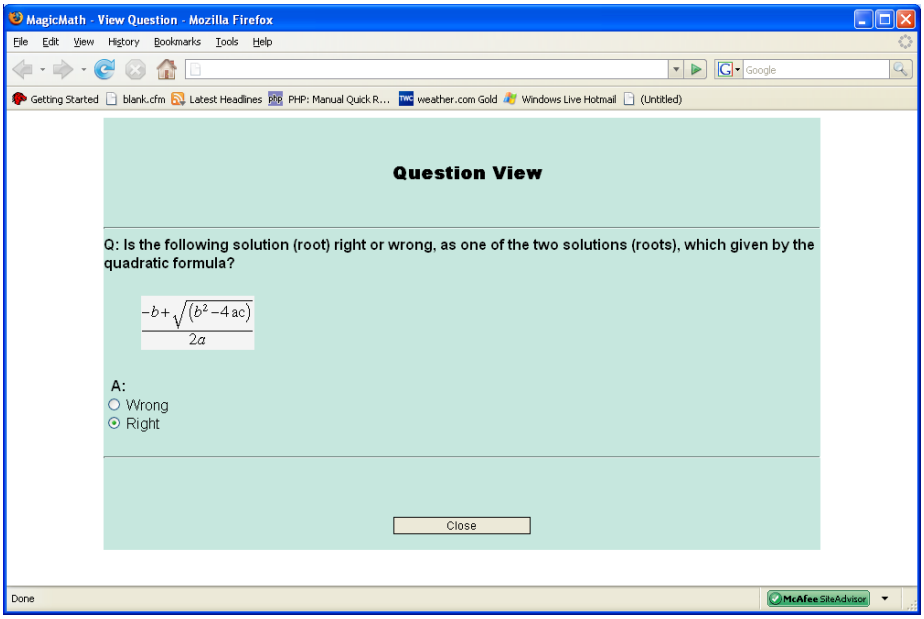


Figure 5.3. Question with Customizable Formula

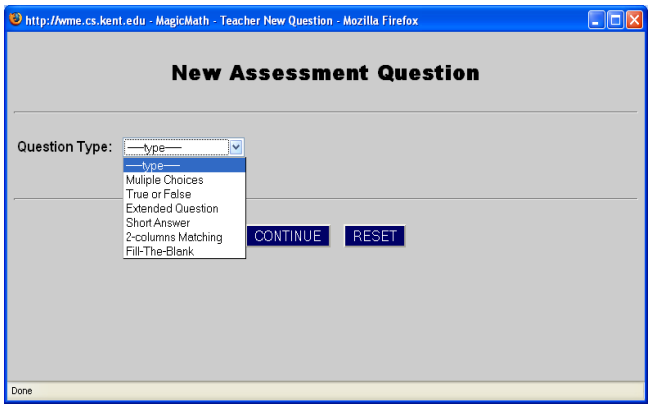


Figure 5.4. Question Types

Authors can import pre-made questions from DMAD into their tests. Once that happens, imported mathematical assessments can be customized, modified if needed, and included in assessment tests. Any question created or imported can be exported to the general DMAD database to be shared with other people. In test management environment, teachers can manage their assessment tests and questions. They can

customize, edit, delete a test, add, view, edit, or deleting questions, set test scores, publish and show a test to students, and handle grades administration (Figure 5.5).

Figure 5.5. Manage Test Window

The authoring tool is a *WYSISWYG test authoring tool* to easily enable teachers to create new test questions from scratch or customize the imported ones. Through this tool, a user can create a mathematical formula/expression and the *corresponding MathML code* will be created automatically by the system, while typing the formula for easy visualization and the formula will be displayed *instantly* in the authoring environment. All conversions between formula's different formats are done automatically and transparently (hidden) from the user for simplicity. A user can also upload images or draw a geometric graph using geoSVG [53] and customize such graphs in the authoring environment. All graphical contents will be handled by the DMAS authoring tool transparently from the user (Figure 5.6). Unlike TWAS systems, the user of DMAS is not required at all to know how to write MathML codes nor how to program in SVG. Any

mathematical assessments created (or imported) of any kinds (text, image, formula, or graph) can be published and shared with others.

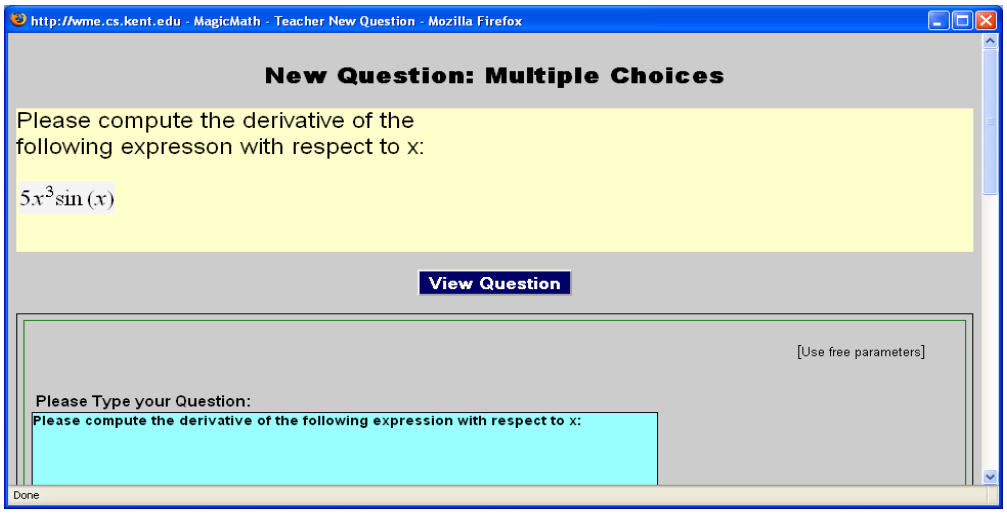


Figure 5.6. New Formula Question

The Ohio Resource Center (ORC) [43] has a plan to use DMAS system and add their assessment test questions to the DMAD bank. This is really a goal of DMAS to promote sharing and enable educational experts from different locations, like people in ORC and other educational institutions, to use the system and enrich it with more mathematical assessments and other useful resources. This will promote sharing of *standard mathematical assessments* and make them available publicly so they can be imported by teachers in Ohio, for example and beyond. Unlike TWAS systems, this hopefully will make a potential for DMAS and support its idea of *sharing mathematical assessments* between experts and educators in different locations.

5.4 Test Publishing

After test sharing, customization, and authoring, comes test publishing. After teachers generate their own assessment tests from scratch (by authoring their own questions) or by importing (and may be customizing) mathematical assessments from the DMAD bank, then they can store their assessment tests for future use. The default status of such assessment tests (questions) is hidden from students for security purposes, which allows the teacher to prepare her test long before class time. So the teacher has to clearly make his assessment test public to be shown to students. A teacher has the authority to show and hide or enable and disable an online test at any time of testing.

DMAS system provides teachers with two testing options: *online testing and paper testing*. Automatic grading and other useful statistics are only supported for the online testing. However, teachers can have hard-copies of their assessment tests for in-class testing at any time. Teachers also have control over students who can/can't take assessment tests and who is eligible for test retake. DMAS provides users with various test question display types: *by given order only, all questions and options are in random order, or random questions non-random options*, and they can set different scores for each question.

The system provides teachers with different ways or *permutations* of test display and can have multiple forms of same exams. Questions in a test can be displayed in various permutations automatically to *enable test taking by students at close proximity*, for instance. For example, the teacher has the option to display same set of questions of a test *randomly* so that each student can see different order of same question set when he

opens the browser. Moreover, options of true-false or multiple-choice questions can be themselves in random order. Teachers also can specify certain/given order for questions to be displayed (Figure 5.7).

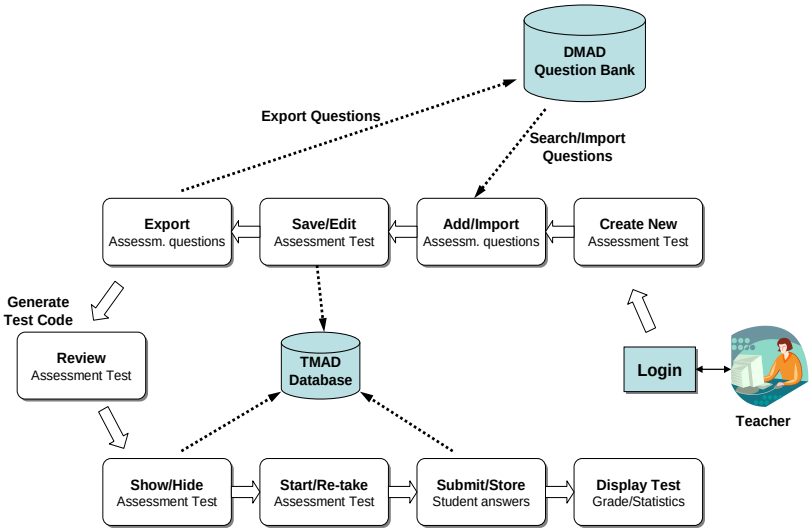


Figure 5.7. Assessment Test Tasks

After preparing a test, all assessment tests are hidden by default for security. When ready, the teacher has to publish the assessment test online and make it public for students to take the test. Every teacher has his own test page with a *unique URL address assigned by the system* and test will be shown or hidden dynamically, and fully controlled by the teacher. The teacher directs students to his page to find the test link to take the test. At the most, one test at a time can be shown on the teacher's test page to students taking the test for clarity and simplicity. A teacher can hide and publish the test at any moment of time.

CHAPTER 6

ONLINE TEST TAKING, GRADING, AND GRADES MANAGEMENT

6.1 Conducting Online Test

The system supports two types of testing: *online testing* and *paper testing*. It supports two test views: *all-per-page*, in which a student views all test questions in one page, and *one-per-page*, where only a single question is shown per page. A student has both views available and can switch between them at any time. The system also supports an *accumulative submission and storage* of student answers on question-by-question basis. In addition, it allows some specific students (controlled only by their teachers) to *Retake* assessment tests if needed. This is useful in case of unpredictable computer errors or student mistakes such as closing test browser window or a computer station hanging up. In such cases, already submitted/stored answers of an authorized student will show up and the student can complete the test from the point he stopped. Through DMAS system, a teacher has options to conduct her assessment test online or in-class (as hard copies) or can be both. During an online test, a student name *always appears* in a small box in top-left corner of a student test page. It is sticky and always on so if a student scrolls test page up or down the name does not scroll. This enables the teacher to verify students just by looking at their computer screens for authenticity purposes.

As mentioned earlier, the DMAS system provides teachers with different ways or *permutations* of test display to students. For example, if a test display is set to *random questions non-random options* by the teacher, same set of test questions will be displayed randomly so that two neighboring students will see different order of same question set when they open their browsers. In addition, if the teacher set test display to *all questions and options are in random order*, then options of true-false or multiple-choice questions can be themselves in random order. This means in a same test question, option A in test1 (for student 1) can be option B, C, or D in test2 (for student 2) which may help teachers decrease cheating of students at close proximity.

Security is a basic requirement of assessment testing in general and in an online testing more specifically. In previous work [25], I have used student login system as a standard and secured login system using student username and password. But interestingly and after consulting mathematics teachers and experts and after real trial in schools, I found the following problems:

1. Students usually forget their usernames, passwords, or both especially at the lower grade levels which can be considered time consuming and wasting class time by helping students to login.
2. Using already known usernames and passwords may not be secured enough and vulnerable because students may give their login personal information to friends or even to classmates to help them in heir tests.

For these reasons and after consulting some mathematics teachers and experts, I came up with different ways of logging into the system as an alternative to standard login system.

After investigating the problem, I came up with three new approaches of login system. The first two ways use *randomly generated test codes* for students to login, first one uses *manual* distribution of test codes while second one uses *electronic* distribution of test codes to students using TSIM tool. The third one is different using *TSIM: Teacher-Student Interaction Mechanism* (Chapter 3). The problem of the first two approaches was that how precisely do we distribute such *randomly generated test codes* to students taking the test setting on different computer machines? Here are the new approaches:

Login using Randomly Generated Test Codes and *manually* distributed: This first way allows a teacher to create test codes through teacher interface. A teacher through test codes interface can determine code type (Numeric, Alphabetic, Alphanumeric, etc), code length (number of digits), and number of students. Then the system randomly generates new test codes based on such settings. A teacher now prints out the generated test codes and *distributes them manually to students in class, one test code per student*. Such per-test codes are generated by teachers through the teacher API (Application Programming Interface (Figure 6.1).

A teacher has a *full control* over students who are allowed to retake the test and he can enable or disable students at any time of testing. Now students can enter their names and assigned test codes to proceed to the test page if successfully logged in. If a student is allowed to retake a test, the system will create a second code for him to be entered to retake the test. (Figure 6.2) shows *re-take test procedure* that verifies the student-code matching. For a student to start or retake an assessment test, he has to have at least one code to do so. The DMAS system will do the following checking steps:

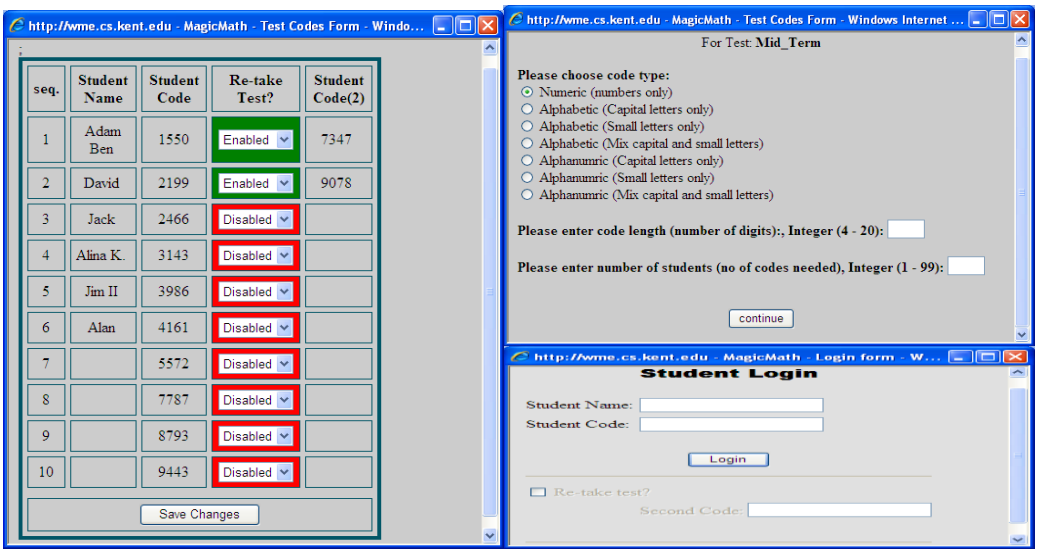


Figure 6.1. Generated Test Codes

- ✓ Check if the student code1 is *valid* (exists and matches the one in the TMAD database), if not then exit.
- ✓ Check if it was *already used*, if not then it is a new student taking the test. Otherwise, go to next step.
- ✓ Check if this student is *allowed* by his teacher to retake the test, if not, this means code is already taken by a classmate and exit. If so, test if second code (code2) matches the TMAD correspondent code given by the teacher, if not exit, or successfully login otherwise.

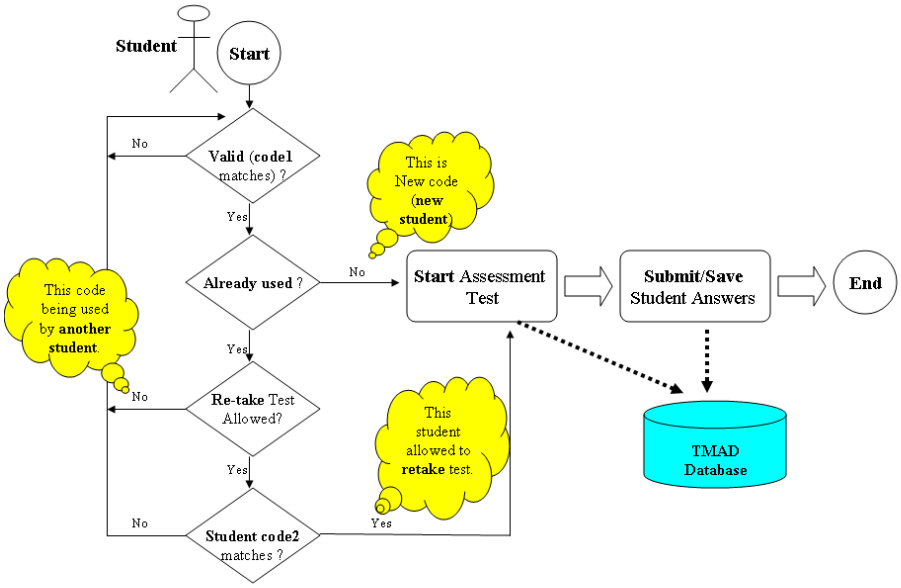


Figure 6.2. Re-take Test Procedure

1. Login using *Randomly Generated Test Codes and Manually distributed*.
2. Login using *Randomly Generated Test Codes and Electronically distributed*.

The second and third (the one currently considered by DMAS) approaches are using *TSIM: Teacher-Student Interaction Mechanism* as a login system. These two approaches were introduced and explained in details in chapter three.

6.2 Automatic Grading and Results Administration

DMAS system has gone under real trials at Kimpton middle school, here in Stow Ohio, and the feedback was encouraging (more details will come in the next chapter). So in this section, we'll find some of the actual test statistics and data but presented and described anonymously. Such actual trials contribute greatly to the continuous

improvement and development of the DMAS system in general and to investigate current ideas, tools, and techniques and come up with new other tools such as TSIM, as a real-time online test supervision tool.

One of DMAS features is supporting an online automatic grading of an assessment test *instantly*. It provides two types of display: *per-student view and per-question view*. The system automatically grades the multiple choices and true-or-false questions and gives the option to grade extended (essay) questions and short answers. The system will also grade questions with mathematical formulas or expressions and will allow students to check their answers if correct, very close, or incorrect.

During a test and if authorized by the teacher, a student always has the ability to update his test answers for mathematical problems as many times as needed and overwrite old ones. The system provides an automatic and dynamic grading of student answers based on last version of stored student answers. It also provides teachers with *useful, real-time, and dynamic statistics of students' performance*. Through Teacher Grading Administration window in DMAS system, a teacher can monitor individual student performance and see how well he does in a particular question or in the whole test, in addition to monitoring whole class performance in a test or in one question in particular (Figure 6.3).

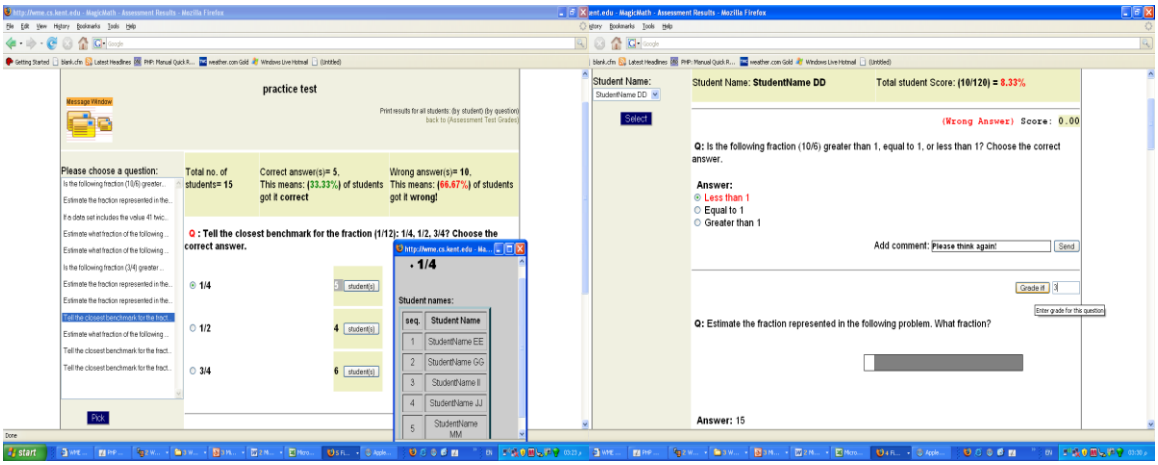


Figure 6.3. Test Grading Administration

Test Grading Administration of DMAS system provides *accumulative live statistics* of a test being taken or statistics of a previous test. In the above figure (Figure 6.3 – left window), it shows *per-question* view in which the system provides statistics based on one question for all students such as number of students taking the test, number of correct answers, number of wrong answers, and percentages. In case of multiple choices and true-or-false questions, it even shows student names that chose a single choice/option of all available choices/options. This is useful, for example in the question above, a teacher can know *instantly* who chose what and can use TSIM (chapter 3) to *send help messages, instructions, or hints to only a group of students who chose wrong options/answers to help them* without asking for help or even interrupting the whole class and disturbing other students who already got right.

DMAS system also gives the option to grade extended (essay) questions and short answers by providing a "grade it" option along with each question. Then it recalculates total grade, percentage, and all other statistics of a student answers. Also in the above

figure, (Figure 6.3 – right window), it shows *per-student* view in which the system provides statistics to a teacher based on one student for all questions. It shows student name, total score, total percentage, individual question grade, question correctness, and the chosen student choice/answer. It also provides a teacher with a *message/comment box* along with each question to send help message, instruction, or hint through TSIM tool (chapter 3) to a specific student regarding a specific question privately. DMAS system also provides print out feature of both views for a teacher of automatically generated grades. So hard copies of detailed student grades can be obtained at any time.

6.3 Question Diagnosis and Linking to TLP Materials

One feature of DMAS system is that it provides *question diagnosis and linking to TLP materials*. The system does not only provide statistics of student answers (section 6.2) but also *dynamically* provides some suggestions as a reference or support for a teacher. The system does that by *suggesting some WME lesson pages* that might help resolve such problems and difficulties exposed by the assessment system. For example, let's go back to the assessment question given in section 6.3 (Figure 6.3 – left window), the assessment question was:

Q. Tell the closest benchmark for the fraction `(1/12): 1/4, 1/2, 3/4`? Choose the correct answer.

```
Option1: 1/4
Option2: 1/2
Option3: 3/4?
```

As we can see only 5 out of 15 students got it correct (`Option1: 1/4`) and two third of students chose the wrong answers (`Option2: 1/2, Option3: 3/4`) as shown in (figure 6.4).

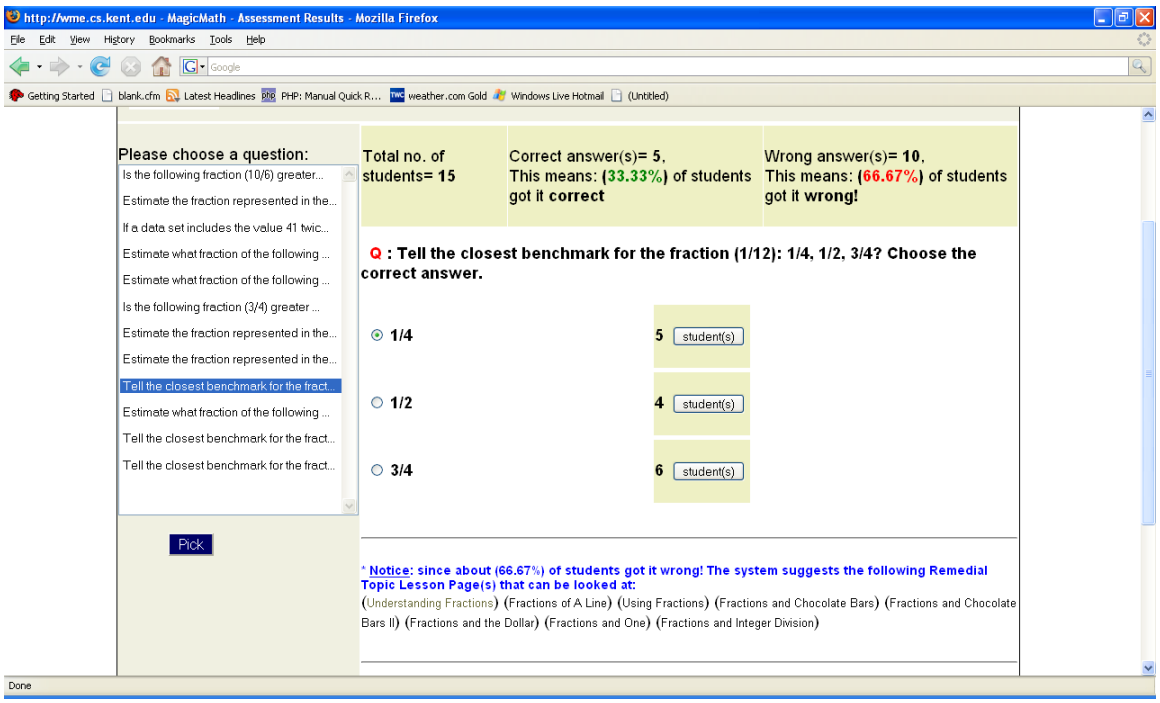


Figure 6.4. DMAS Question Diagnosis and Linking to TLP Materials

The assessment system figures out that 66.67% (about two third) of students got it wrong and so there is a problem with this question. *Thus acknowledging the big percentage (more than halve of students got it wrong), the system dynamically alerts the teacher of such problem and suggests some links to related Topic Lesson Pages (TLP)* [26] based on question itself, its type, keywords, and some metadata stored in the databases. In our example, the question is about fractions and the generated message for the teacher would be something like: "*Notice: since about (66.67%) of students got it wrong! The system suggests the following Topic Lesson Page(s) that can be looked at:*"

and then lists some links to the WME Lesson Pages so the teacher can decide to go and review such materials or use something else. So DMAS system provides such materials as *bonus features for teachers and not as a requirement or a decisive tool*. Thus, the assessment system aims to help correlate such diagnostic information with school Lesson Pages (or any other materials suggested by the teacher in the future), which might help students overcome difficulties exposed by the assessments. However, this will need more investigation from the education point of view to determine or at least narrow the actual possible causes of student's mistakes.

6.4 Importing and Exporting DMAS File Formats (DFF)

DMAS system also supports *DMAS File Formats (DFF)* of questions, test grades, and statistics. It allows *importing and exporting assessment materials such as student grades and test questions into other formats dynamically*. It supports the following DMAS file formats:

- ✓ Excel file Format (.xls)
- ✓ Microsoft word file Format (.doc)
- ✓ Text file Format (.txt)
- ✓ Rich Text Format (.rtf)
- ✓ PowerPoint file Format (.ppt)
- ✓ Portable Document Format (.pdf)
- ✓ Comma Separator Value (CSV)

Different DMAS File Formats can be easily obtained through DMAS system and by invoking the *DMAS File Format Service (DFFS)* (figure 6.5). Such formats can be

useful for teachers to have and export to local storage media for *further customization, portability, and usability* into other standard programs or applications such as Microsoft Excel, Microsoft word, Microsoft PowerPoint, and Adobe Portable Document Format (PDF).

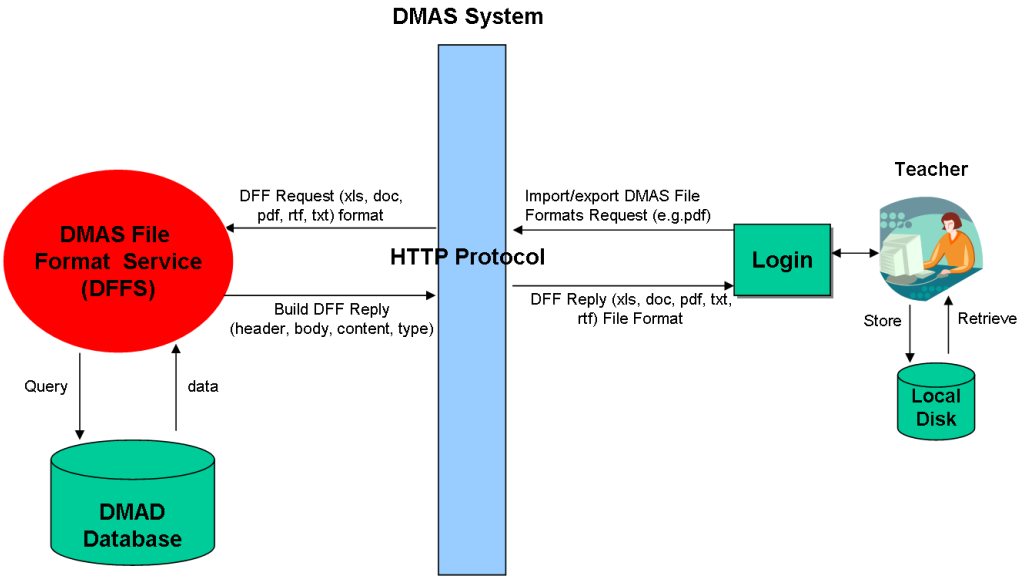
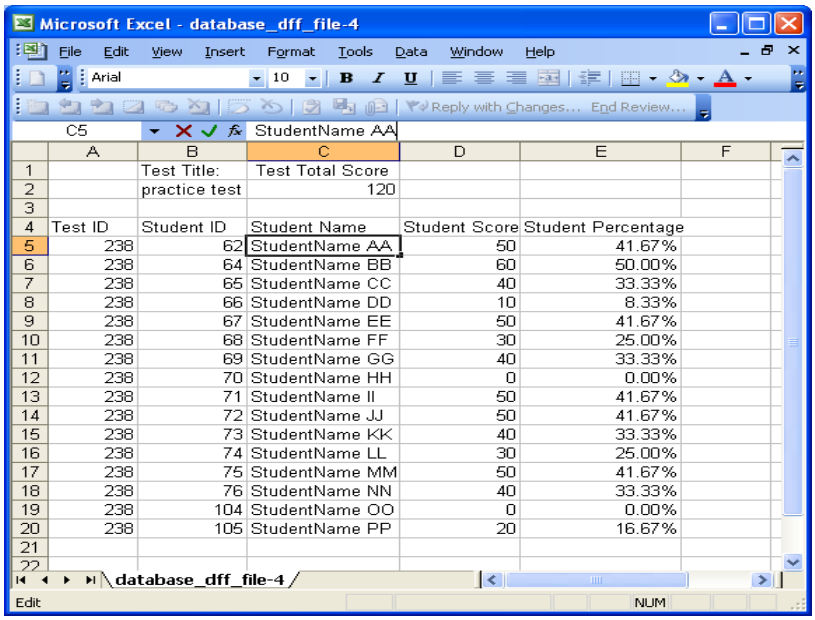


Figure 6.5. DMAS File Formats (DFF)

After he logs in to DMAS system, a teacher can choose to view and export assessment materials to different DMAS File Formats (DFF) such as PDF format. File formats will be translated internally as *DFF request*. DMAS system passes the DFF request using *HTTP Protocol* to the *DMAS File Format Service (DFFS)*. The DFFS service *queries* needed DMAD databases for required data, builds HTTP reply (header, body, content, type, etc.), and then completes the *DFFS reply* including the requested DFF formats (PDF for example). DMAS system passes the DFF contents as *HTTP reply*

to back the teacher browser and then the generated DFF file contents will be extracted correctly and can be stored on teacher's computer local disk or viewed by dynamically invoking required application/program locally. Here are a few DFF examples:

- Example 1: If a teacher wants to export student grades of an assessment test into an external excel file or PDF file and have a report with statistics, the DMAS system can do that *dynamically* using *DFFS service*. It calculates student score, student percentage, and test report, and exports all of that into an external excel file (Figure 6.6).



	A	B	C	D	E	F
1		Test Title:	Test Total Score			
2		practice test	120			
3						
4	Test ID	Student ID	Student Name	Student Score	Student Percentage	
5	238	62	StudentName AA	50	41.67%	
6	238	64	StudentName BB	60	50.00%	
7	238	65	StudentName CC	40	33.33%	
8	238	66	StudentName DD	10	8.33%	
9	238	67	StudentName EE	50	41.67%	
10	238	68	StudentName FF	30	25.00%	
11	238	69	StudentName GG	40	33.33%	
12	238	70	StudentName HH	0	0.00%	
13	238	71	StudentName II	50	41.67%	
14	238	72	StudentName JJ	50	41.67%	
15	238	73	StudentName KK	40	33.33%	
16	238	74	StudentName LL	30	25.00%	
17	238	75	StudentName MM	50	41.67%	
18	238	76	StudentName NN	40	33.33%	
19	238	104	StudentName OO	0	0.00%	
20	238	105	StudentName PP	20	16.67%	
21						
22						

Figure 6.6. Exported DFF Excel file Using DFFS

- Example 2: If a teacher wants to have his assessment test questions in other media such as Microsoft word file format or PDF file format, the DMAS system does

that *dynamically* using the *DFFS service* and exports such test assessment questions into requested formats (e.g. PDF file format) (Figure 6.7).

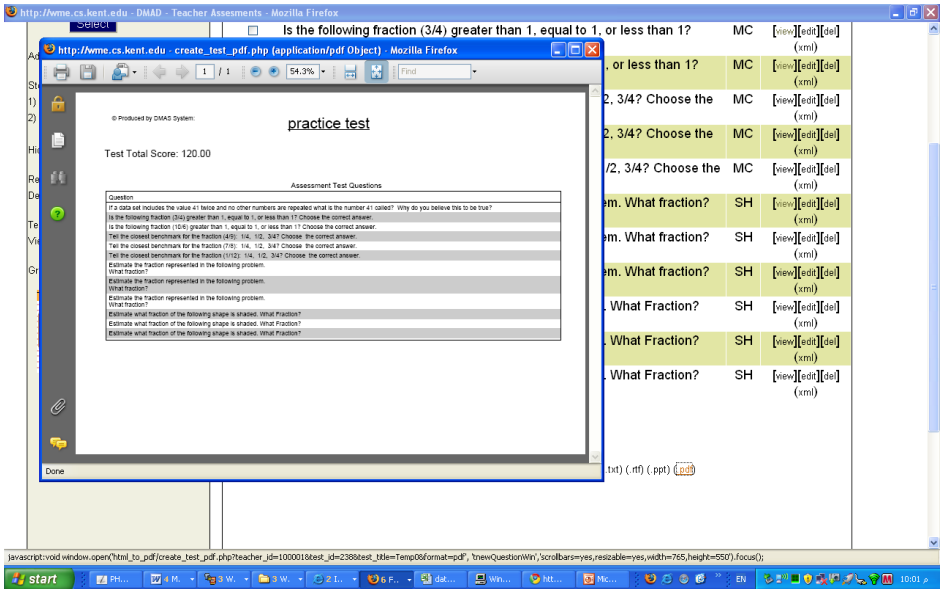


Figure 6.7. Exporting Test Questions in DFF Format (PDF file) Using DFFS

CHAPTER 7

DMAS SYSTEM INTEROPERABILITY WITH OTHER ONLINE SYSTEMS

7.1 DMAS Interoperability with Existing Web Systems

Interoperability is important for Web systems, applications, and services to consider some kinds of interactions and communications with other existing online applications for the sake of providing information or online services. Unlike TWAS and other online assessment tools, DMAS is *interoperable* with other online systems and applications such as WME system. It provides such systems and applications with required mathematical assessments, data, files, and other materials. The DMAS system has APIs interface that can serve different online requests and service calls and provides online service replies. It can be *interoperable* in two ways:

- *Server-to-Server*: This way communications are established between two servers such as WME server to DMAS server. In this case, WME server makes requests while DMAS server makes replies using *File Transfer Protocol* (*FTP*), a network protocol used to transfer data and files from one computer to another through a network such as the Internet, or through *Secure File Transfer Protocol* (*SFTP*), or *SCP (Secure Copy Program)*. Such servers (like WME server) request

mathematical assessments and DMAS server receives such requests, builds replies, and sends required data back to the servers.

- *Client-to-Server*: The communications here are from the internet browser (as a client) to DMAS server and vise versa. DMAS system uses *cross-browser* features and standard technologies to communicate and serve online requests received from different applications (Figure 7.1).

In either case, DMAS system API interface will handles all different calls and requests in proper and transparent way.

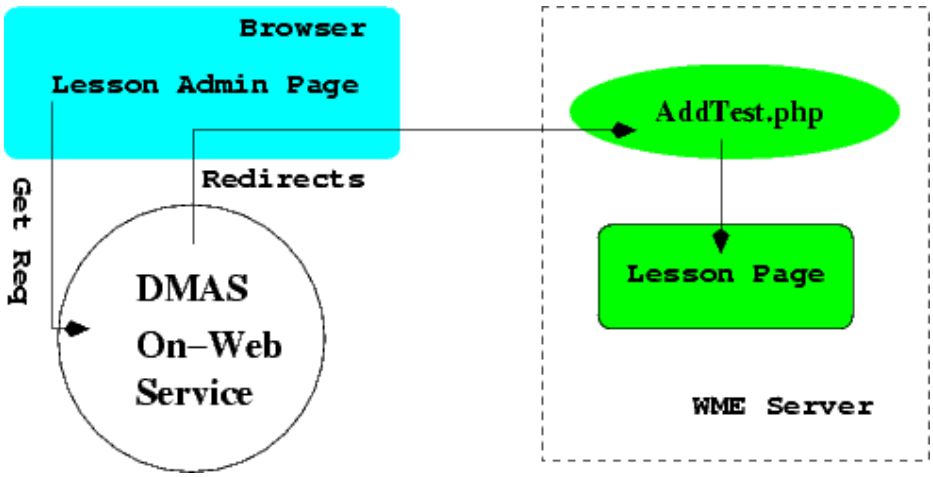


Figure 7.1. DMAS Server Provides Mathematical Assessments to WME

7.2 Integrating DMAS with WME System

DMAS system has been integrated with WME system through a well-defined APIs interface to provide assessment questions and tests to WME clients. WME users through WME client (Web browser) and DMAS API interface can search for assessment

questions in DMAS, and import/include them directly in their tests or Topic Lesson Pages (TLP) [26, 27]. DMAS *On-Web Service* through its APIs interface uses *callback* technique and function to communicate and interact with WME client. DMAS interface uses callback URL and *unique unitID* to register and know the requesting Lesson Page on WME client and then when finishes, DMAS will callback such TLP and provide it with the requested mathematical assessments (Figure 7.2).

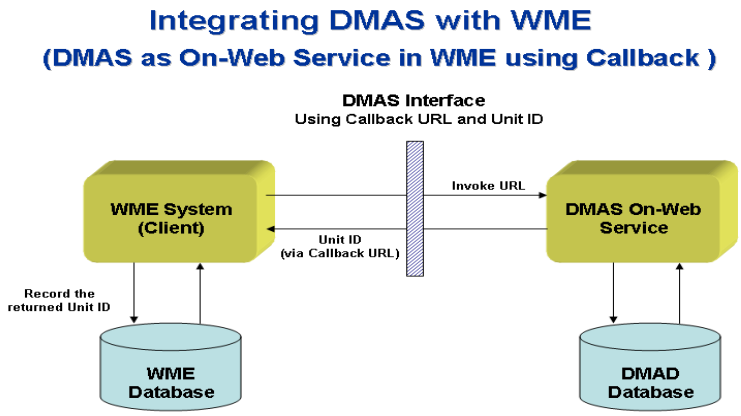


Figure 7.2. Integrating DMAS with WME Using Callback

Now let us take a look closely to the *DMAS-WME interoperability* in more details to show how WME client communicates with DMAS server (Figure 7.3).

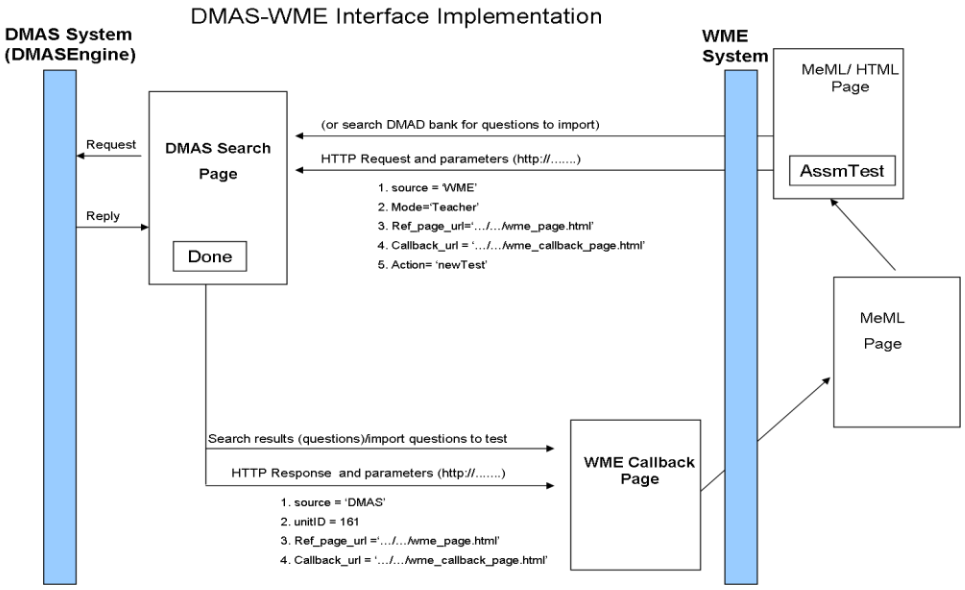


Figure 7.3. DMAS-WME Interoperability Using Callback

The online communications start from the WME client when a user (teacher) through MeML [29, 55, 59, 64] page or HTML page requests mathematical assessments such as a test or a question. The DMAS-WME callback builds *HTTP request* and then invokes DMAS search page with required HTTP request parameters through HTTP Protocol such as: *source* of client request, *mode*, WME *referring page URL*, WME *callback page URL*, and the needed action (Table 7.1).

```
1. Source = "WME"                //Source of the request: WME Client
2. Mode = "Teacher"              //Also can be: Mode="Student"
3. Ref_page_url ="http://wme.cs.kent.edu/.../wme_page.html"
                                 // URL of the referring WME page
4. Callback_url ="http://wme.cs.kent.edu/.../wme_callback_page.html"
                                 // URL of WME callback page
5. Action = "newTest"            //Action needed: can be also newQuestion, etc.
```

Table 7.1. DMAS-WME HTTP Request Parameters

The DMAS search page receives such request parameters, adds more search parameters to the request, and then passes it on to the DMAS search engine (*DMASEngine*) on DMAS server for processing. When done, DMASEngine fetches needed information from DMAD databases and sends it back to DMAS-WME Interface. The latter in turn constructs *HTTP reply* with some HTTP request parameters: *source* of server reply, *mode*, unique object identifier, *unitID*, representing an assessment test or question, WME *referring page URL*, and WME *callback page URL* (Table 7.2).

```
1. Source = "WME"                //Source of the request: WME Client
2. unitID = "..."                //It is a unique object id
3. Ref_page_url ="http://wme.cs.kent.edu/.../wme_page.page"
                                 // URL of the referring WME page
4. Callback_url ="http://wme.cs.kent.edu/.../wme_callback_page.html"
                                 // URL of WME callback page
```

Table 7.2. DMAS-WME HTTP Response Parameters

WME *callback page URL* receives the information and then WME client records the returned *unitID* that was given back by the DMAS server in its database for future calls or requests. Finally, the requesting WME client page gets the needed mathematical assessments from DMAS and can include them in a WME Topic Lesson Page (TLP).

7.2.1 Teacher Mode

As an external WME system, DMAS-WME Interface provides WME users with two modes to use DMAS system from WME: *teacher mode and student mode*. Ina *teacher mode*, a teacher can use his *page admin* in WME Topic Lesson Pages (TLP) or Topic Module (TM) page (fractions module is an example) to add a new question set to

existing WME page, to create or customize mathematical assessments, and to manage lesson page test (Figure 7.4).

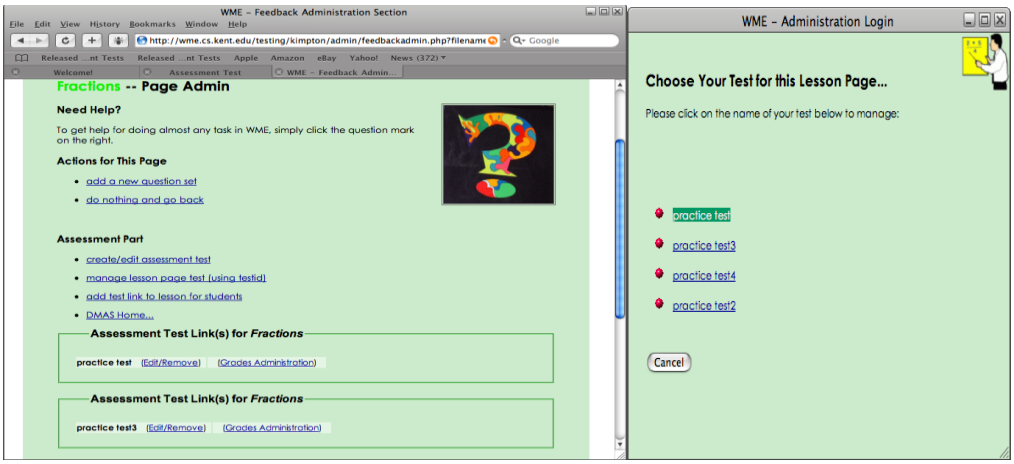


Figure 7.4. WME in Teacher Mode

Through *page admin*, a teacher can use DMAS-WME mode to communicate with DMAS system and create a new test or add assessment questions to an existing *question set* of a lesson page. DMAS-WME Interface creates links to the created test automatically and provides a teacher with a capability of *grades administration* of a test. In a feedback question test of a lesson page, a teacher can create his own questions or use DMASEngine to search for mathematical assessments in DMAS bank and then import them dynamically (Figure 7.5).

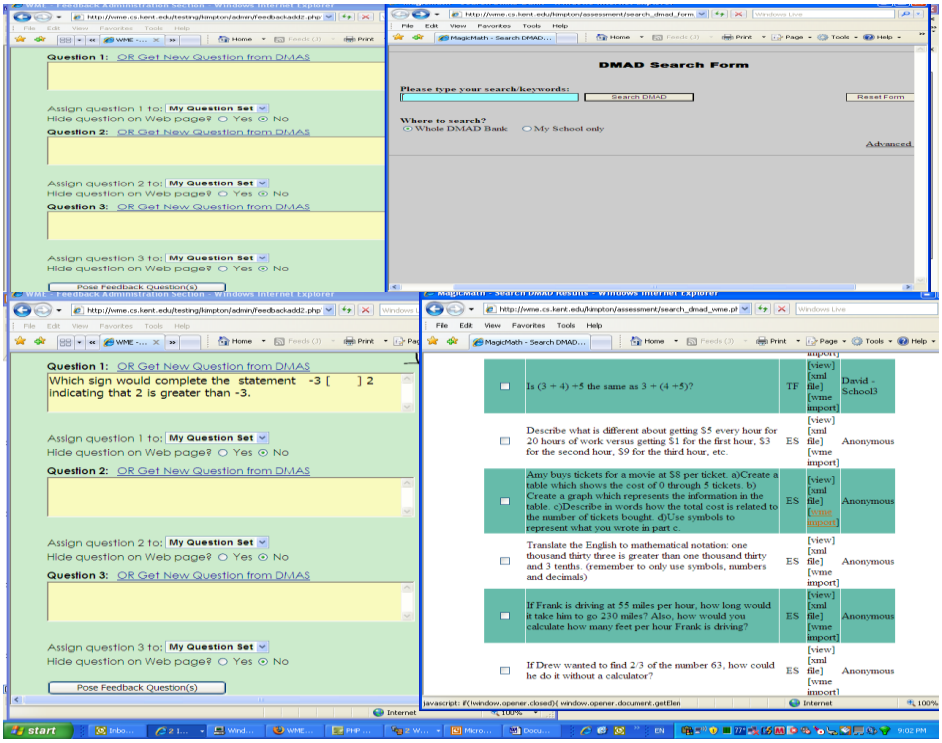


Figure 7.5. Importing Questions in TLP from DMAS

7.2.2 Student Mode

The *interoperability* between DMAS and WME also provides WME users with a *student mode* so students can use DMAS. From a WME lesson page (TLP or TM), a student can choose his teacher's name from links appear automatically in a class page without even a need for student to login to the system and feedback question sets will be displayed (Figure 7.6).

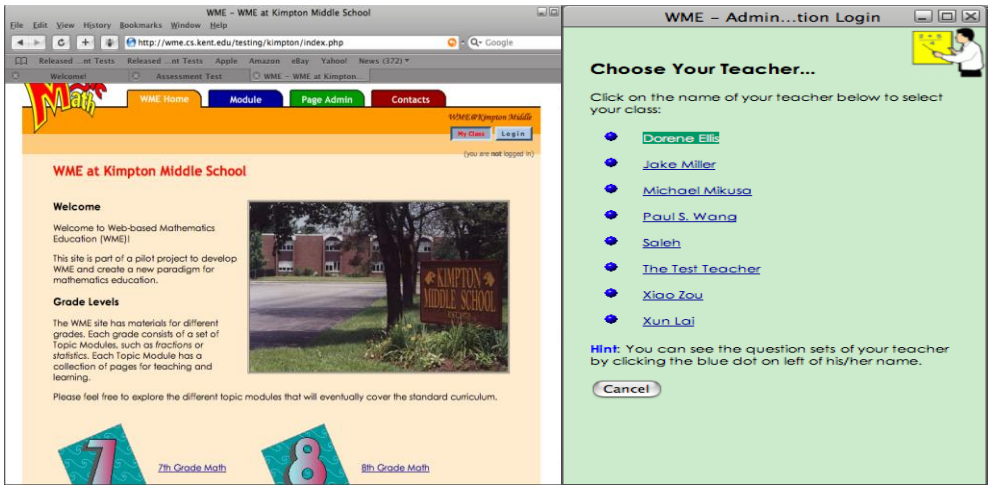


Figure 7.6. DMAS-WME in Student Mode

A teacher directs students to a certain lesson page and feedback question set or assessment test links that show up automatically so that students can answer such feedback questions or take an assessment test. Student names and answers of feedback questions will be saved along with a Topic Lesson Page (TLP) *locally* in WME system databases. On the other hand, an assessment test will be conducted in DMAS system itself through *unitID* provided by WME, and the test will be stored along with student names and answers in DMAS system databases *remotely* (Figure 7.7).

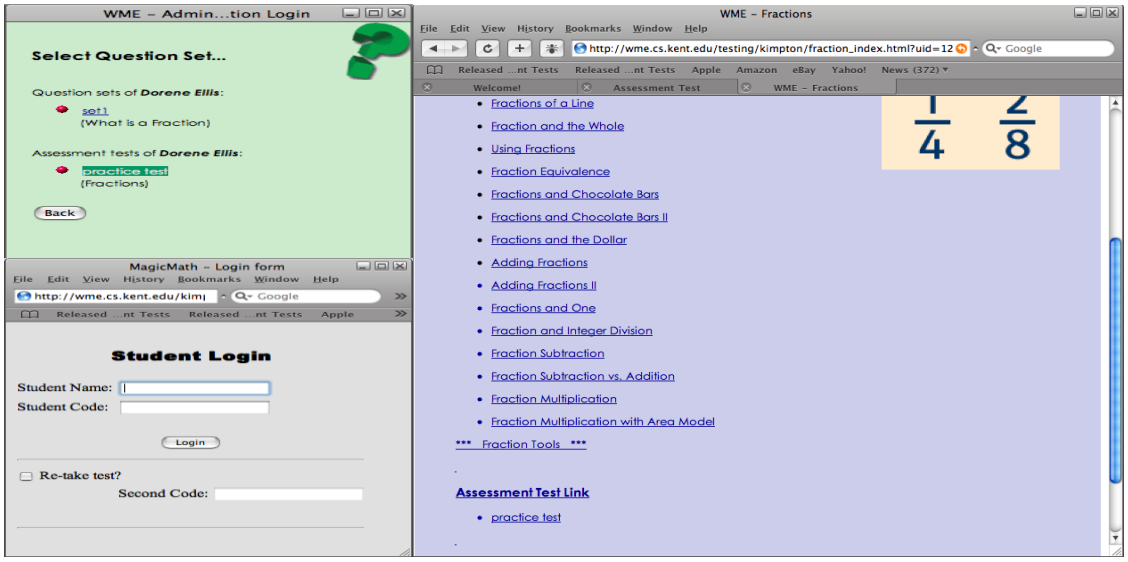


Figure 7.7. Question Set and Assessment Test Links

7.3 Building of Experimental System to Work with WME

The DMAS system has been used experimentally in WME system. It *interoperates* and interacts with other parts of the WME components such as Topic Lesson Pages (TLP) and Topic Modules (TM). It provides WME Lesson Pages with pre-assessment and post-assessment set of questions to prepare students and measure their understanding of lesson concepts. For example, a teacher in WME administration mode can use pre or post assessment options to trigger DMAS system to author or import assessment questions that can be included in a particular lesson page. Teachers using WME can also make their own assessment tests, probably independent from any specific Lesson Page such as Midterm exams, using DMAS and have them automatically stored and graded. In such cases, DMAS system can produce performance statistics and provide some real-time online test supervision.

7.4 Pilot Use in Schools to Collect Teacher and Student Feedback

DMAS system was piloted in a seventh grade class at Kimpton Middle school (Stow, United States) on 10/05/2007, and the first experience of in-class trial was an interesting one. I collected the following actual *qualitative* feedback from the teacher and students who participated in the actual trial of DMAS and after taking DMAS online assessment test in-class and. There is also a comparison to regular written and paper tests:

- Majority of students agree that the online test is more fun.
- The test went very well and smoothly with no major problems.
- The teacher and students like the random order display of questions.
- The teacher liked the automatic grading and *real-time* performance statistics of student answers.
- Most students preferred the online assessment tests over the regular paper tests.
- Ease of use, simplicity, and clarity of how to take and submit answers were mentioned.
- One student preferred the written test due to having no computer at home.
- Also easy to update answers and writing using keyboard is cleaner and easier to use than hand writing.

However, this needs more trials and educational research to measure the effectiveness of using DMAS system in WME or any other system and closely measure student understanding and performance of mathematical problems before and after using DMAS system. In the future, my goal is to put DMAS system under extensive trial in schools and collect feedback and suggestions from teachers, students, school

administrators and education experts to help me evolve DMAS. As more schools adopt DMAS, the distributed nature of DMAS will be demonstrated in realistic situations.

7.5 DMAS and Other Fields

DMAS system goal is to help teaching and learning mathematics on the Web. Mathematics problems may contain mathematical expressions, formulas, or geometric graphs which may not be trivial to represent, deliver, or render on different Web browsers. In addition to problems and issues incurred in an automatic mathematical answer checking and grading. So a basic question will be: What prevents DMAS system from becoming a useful tool for "Sharing assessments in other areas such as English or history?" This is unknown yet but if DMAS system works well in teaching, learning, and assessing mathematics on the Web, then DMAS may consider other fields such as English, history or other areas, to be a future research.

CHAPTER 8

MATHEMATICS ASSESSMENT MARKUP LANGUAGE: MAML

8.1 The Need for Sharing Assessments

Unlike TWAS systems, one of the basic goals and characteristics of DMAS system is *promoting sharing of mathematical assessments* between participants in different locations through the internet. So there was a need for DMAS system to find a standard way to share its mathematical assessments on the Web. Thus, the idea of creating *an XML markup language* for DMAS as a standard way for representation and sharing such mathematical assessments on the internet was crucial. DMAS system provides a new assessment language called *MAML (Mathematics Assessment Markup Language).*

The goal of MAML language is to encode and transmit assessment questions and tests to and from DMAS system and to interact with outside systems. MAML will be used for *representation/encoding of assessment questions and exams*. It defines markup elements and attributes such as question head, type, classification, body, rubric, and answer. Using MAML, DMAS users will be able to share assessment materials on the Web and import/export questions in *standard, well-formed, and valid XML formats*. DMAS Web service will receive and return MAML encoded data to and from other online systems and applications.

8.2 MAML Elements and Language Generator

MAML is defined by a number of XML tags and attributes. MAML elements (markup tags) and attributes are for describing different components of mathematical assessments. MAML elements can be used with combination of other XML-compatible languages such as MathML and XHTML. For example, an XML file of MAML format can include MAML tags and MathML code representing a mathematical formula together. In other words, MAML is an XML language itself that can include another XML language such as MathML or SVG. MAML language is flexible also open for further extensibility (Table 8.1).

Category	**MAML Elements**
Education	*test, question, exercise, homework, diagnosis, answer, solution*
Description	*author, type, keywords, classification, comment, rating*
Content	*dmas, q_head, q_body, q_text, q_diagram, q_image, q_choices, choice, svg, infix, math*

Table 8.1. MAML Elements

DMAS assessment question consists of two main parts: *question head* and *question body*. The *Question head* includes metadata about the question such as author's name, search keywords, organization category or classification, and question answer. This assessment question organization is similar to a Web page organization in which it has a page header that has page metadata and body with page contents (Table 8.2).

```
<q_head>
   <author>author name</author>
   <keywords>Measurement, Area, Rectangle</keywords>
   <classification>Plan Geometry</classification>
   <answer>question answer</answer>
</q_head>
```

Table 8.2. MAML Question Header

The *Question body* contains the actual contents of the question such as question text, image, SVG geometrical graph, mathematical formula in infix mode or using corresponding MathML code, choices (if any), and other contents (Table 8.3).

```
<q_body>
  <q_text>What is the Area a rectangle with height=h and base=b?
  </q_text>
  <q_diagram>
     <svg  xmlns="http://www.w3.org/2000/svg">
        <rect x="100" y="20" width="40" height="60" />
     </svg>
  </q_diagram>
  <q_choices>
     <choice id="1">
       <math xmlns='http://www.w3.org/1998/Math/MathML'>
         <mi>h</mi><mo>+</mo><mi>b</mi>
       </math>
       ..........
     </choice>
  </q_choices>
</q_body>
```

Table 8.3. MAML Question Body

DMAS provides a built-in *MAML language generator module* which retrieves mathematical assessments from databases and dynamically encodes tests and questions to create XML documents in MAML formats. Such XML documents in MAML formats containing mathematical assessments can be exported to other online systems so mathematical assessments can be extracted and shared or documents can be transformed into other XML formats (Figure 8.1).

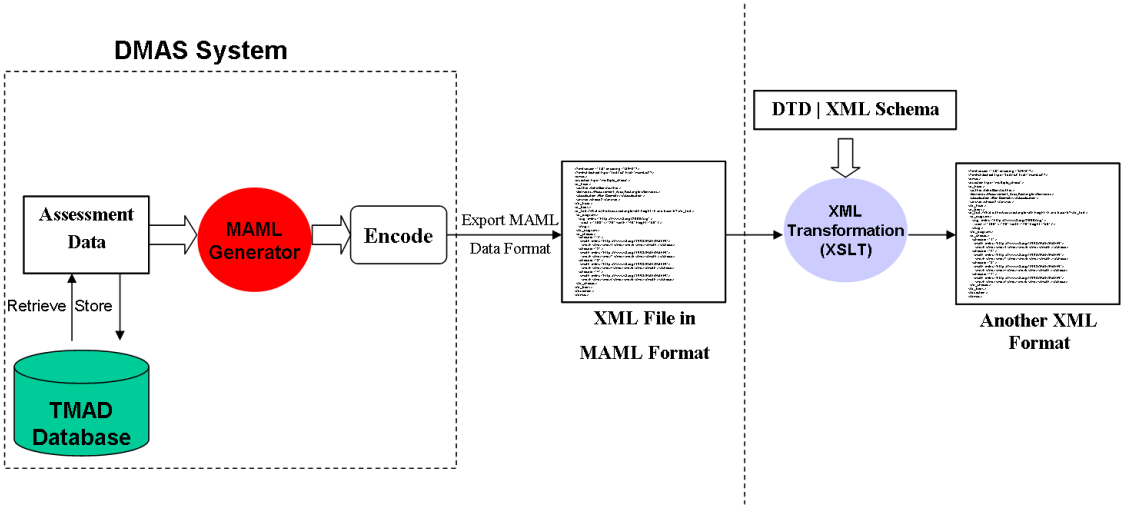


Figure 8.1. MAML Language Diagram

Here is an example of XML file of MAML markup Language (Table 8.4).

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<dmas>
   <question type="Multiple Choices">
      <q_head>
        <author>Dan Alan</author>
        <keywords>fraction number, fraction</keywords>
        <classification>Algebra</classification>
        <comments>fraction number, fraction comparison</comments>
        <answer>9/12</answer>
      </q_head>
      <q_body>
        <q_text>Which fraction below is equivalent to 3/4.</q_text>
        <q_diagram />
        <q_choices>
                <choice id="1">27/36</choice>
                <choice id="2">8/16</choice>
                <choice id="3">4/8</choice>
                <choice id="4">9/12</choice>
        </q_choices>
      </q_body>
  </question>
</dmas>
```

Table 8.4. An Example of XML File of MAML

The DMAS system uses *XML DOM* (XML Document Object Model) as a standard way for accessing and manipulating MAML and XML data and documents. The *MAML markup language creates its XML documents using PHP programming language*

and XML DOM extension (DOM only in the new version, PHP5). After retrieving data from DMAD databases, MAML uses JavaScript and DOM API (with PHP 5) to create its XML documents dynamically and to encode tests and questions (Table 8.5).

```
//----------------------- Start new XML document ------------------------
// create the new XML document
$dom = new DOMDocument('1.0', 'iso-8859-1');

// create the 'dmas' root element:
$dmad = $dom->appendChild($dom->createElement('dmas'));

// create the 'question' element:
$question = $dmad->appendChild($dom->createElement('question'));
$question->setAttribute("type", $aq_title);

// create the 'q_head' element:
$q_head = $question->appendChild($dom->createElement('q_head'));
$author = $q_head->appendChild($dom->createElement('author'));
$author->appendChild($dom->createTextNode($aq_author));
$keywords = $q_head->appendChild($dom->createElement('keywords'));
$keywords->appendChild($dom->createTextNode($q_keywords));
$classification = $q_head->appendChild($dom->createElement('classification'));
$classification->appendChild($dom->createTextNode($au_name));
$comments = $q_head->appendChild($dom->createElement('comments'));
$comments->appendChild($dom->createTextNode($q_comments));
$answer = $q_head->appendChild($dom->createElement('answer'));

// create the 'q_body' element:
$q_body = $question->appendChild($dom->createElement('q_body'));
$q_text = $q_body->appendChild($dom->createElement('q_text'));
$q_text->appendChild($dom->createTextNode($aq_question));
$q_diagram = $q_body->appendChild($dom->createElement('q_diagram'));
//-----------------if svg_data != ''---------------
if($svg_data != '')
{
        $svg = $q_diagram->appendChild($dom->createElement('svg'));
        $svg->setAttribute("xmlns", "http://www.w3.org/2000/svg");
        $svg_data = $svg->appendChild($dom->createElement('svg_data',
$svg_data));
}
//....................
//....................
//....................
$xml_result = $dom->saveXML();
echo '<br />To see the produced XML version click: (<a href="dmad.xml"
target="_blank">dmad.xml</a>) [size: ' . $dom->save("dmad.xml") . '] bytes<br
/>';
```

Table 8.5. A Partial Code of PHP DOM APIs of MAML

The following example shows the markup of a multiple-choice assessment question (Table 8.6).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="maml.xsl"?>
<dmas>
<question type="multiple_choice">
<q_head>
  <author>Johe Doe</author>
  <keywords>Measurement, Area, Rectangle</keywords>
  <classification>Plan Geometry</classification>
  <answer>choice 2</answer>
</q_head>
<q_body>
<q_text>What is the Area a rectangle with height=h and base=b?</q_text>
  <q_diagram>
     <svg  xmlns="http://www.w3.org/2000/svg">
        <rect x="100" y="20" width="40" height="60" />
     </svg>
  </q_diagram>
  <q_choices>
     <choice id="1">
       <math xmlns='http://www.w3.org/1998/Math/MathML'>
         <mi>h</mi><mo>+</mo><mi>b</mi></math></choice>
     <choice id="2">
       <math xmlns='http://www.w3.org/1998/Math/MathML'>
         <mi>h</mi><mo>*</mo><mi>b</mi></math></choice>
     <choice id="3">
       <math xmlns='http://www.w3.org/1998/Math/MathML'>
         <mi>h</mi><mo>-</mo><mi>b</mi></math></choice>
     <choice id="4">
       <math xmlns='http://www.w3.org/1998/Math/MathML'>
         <mi>h</mi><mo>/</mo><mi>b</mi></math></choice>
  </q_choices>
</q_body>
</question>
</dmas>
```

Table 8.6. The Markup of a Multiple-Choice Assessment Question

DMAS system provides DTD (Data Definition Language) for validation of XML documents generated by MAML language. The following is the *DTD* sample of the multiple-choice assessment question example (Table 8.7).

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT dmas (question)>
<!ELEMENT question (q_head, q_body)>
<!ATTLIST question
        type CDATA #FIXED "multiple_choice"
>
<!ELEMENT q_head (author, keywords, classification, answer)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT keywords (#PCDATA)>
<!ELEMENT classification (#PCDATA)>
<!ELEMENT answer (#PCDATA)>
<!ELEMENT q_body (q_text, q_diagram, q_choices)>
<!ELEMENT q_text (#PCDATA)>
<!ELEMENT q_diagram (svg)>
<!ELEMENT svg (rect)>
<!ATTLIST svg
        xmlns CDATA #FIXED "http://www.w3.org/2000/svg"
>
<!ELEMENT rect EMPTY>
<!ATTLIST rect
        x CDATA #FIXED "100"
        y CDATA #FIXED "20"
        width CDATA #FIXED "40"
        height CDATA #FIXED "60"
>
<!ELEMENT q_choices (choice+)>
<!ELEMENT choice (math)>
<!ATTLIST choice
        id (1 | 2 | 3 | 4) #REQUIRED
>
<!ELEMENT mo (#PCDATA)>
<!ELEMENT mi (#PCDATA)>
<!ELEMENT math (mi, mo, mi)>
<!ATTLIST math
        xmlns CDATA #FIXED "http://www.w3.org/1998/Math/MathML"
>
```

Table 8.7. MAML DTD of the Multiple-Choice Question

MAML also supports *XML Schema* as an XML based alternative way to DTD for XML validation. Here is the *XML Schema* sample of the multiple-choice assessment question example mentioned earlier (Table 8.8).

The JavaScript and XML DOM can be used to parse and display XML documents of MAML in HTML format. The XSLT style sheet for MAML (maml.xsl, for example) is responsible to translate MAML markup into XHTML + SVG + MathML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:ns1="http://www.w3.org/1998/Math/MathML"
xmlns:ns2="http://www.w3.org/2000/svg"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="dmas">
        <xs:complexType>
                <xs:sequence>
                        <xs:element ref="question"/>
                </xs:sequence>
        </xs:complexType>
</xs:element>
        <xs:element name="question">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="q_head"/>
                                <xs:element ref="q_body"/>
                        </xs:sequence>
                        <xs:attribute name="type" use="required">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration
value="multiple_choice"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                </xs:complexType>
        </xs:element>
        <xs:element name="q_text" type="xs:string"/>
        <xs:element name="q_head">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="author"/>
                                <xs:element ref="keywords"/>
                                <xs:element ref="classification"/>
                                <xs:element ref="answer"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="q_diagram">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="ns2:svg"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="q_choices">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="choice" maxOccurs="6"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="q_body">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="q_text"/>
                                <xs:element ref="q_diagram"/>
                                <xs:element ref="q_choices"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <xs:element name="keywords">
                <xs:simpleType>
                        <xs:restriction base="xs:string">
                                <xs:enumeration value="Measurement, Area,
Rectangle "/>
                        </xs:restriction>
```

```
                </xs:simpleType>
        </xs:element>
        <xs:element name="classification">
                <xs:simpleType>
                        <xs:restriction base="xs:string">
                                <xs:enumeration value="Plan Geometry"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:element>
        <xs:element name="choice">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="ns1:math"/>
                        </xs:sequence>
                        <xs:attribute name="id" use="required">
                                <xs:simpleType>
                                        <xs:restriction base="xs:byte">
                                                <xs:enumeration value="1"/>
                                                <xs:enumeration value="2"/>
                                                <xs:enumeration value="3"/>
                                                <xs:enumeration value="4"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                </xs:complexType>
        </xs:element>
        <xs:element name="author">
                <xs:simpleType>
                        <xs:restriction base="xs:string">
                                <xs:enumeration value="Johe Bob"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:element>
        <xs:element name="answer">
                <xs:simpleType>
                        <xs:restriction base="xs:string">
                                <xs:enumeration value="choice 2"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:element>
</xs:schema>
```

Table 8.8. MAML XML-Schema of the Multiple-Choice Question

CHAPTER 9

ANSWER CHECKING CHALLENGES

In this chapter, I introduce the challenges of computerized automatic answer checking of mathematical problems and also some simple ideas and experimental work of mathematical answer checking. The investigation of usefulness and challenges of automatic checking (grading or marking) of mathematical answers have been addressed in some other systems [1, 2, 4]. However some of these systems require users to have previous knowledge of Computer algebra Systems (CAS) and their syntax, based on non-automatically generated questions (previously known problem types), some require knowledge of using syntax of their mathematical editor to enter formulas or expressions, or require synchronization between browser (client-side) and the answer checking service (server-side) for answer and grading.

In this chapter, *I have started preliminary experimental work of answer checking* and proposed a simple *Web-Based Mathematical Answer/Expression Checking Service for DMAS (DMAS-MACS)*. This will allow users of DMAS system (students and teachers alike) to check their mathematical answers and/or expressions, to make the mathematical answer checking and grading done automatically, and make DMAS provides some feedback to users regarding their answers.

9.1 Answer Checking Problems and Challenges

I have done some research and investigation in *checking mathematical expression or formula answers, evaluating the expressions, returning some useful feedback, and grading student answers of a test.* This actually encounters some problems and difficulties such as: essay/extended and short answer questions are difficult to check or automatically grade. Also, suppose the mathematical question is about computing the derivative with respect to x of the expression:

$6x^2\cos(x)$? And if the derivative answer was:

$12x\cos(x)-6x^2\sin(x)$

Then the Answer Checking Service program in this case has to decide and return true or false otherwise.

Another problem is expanding a polynomial, there will be problems involved in particular when

x^2+2x+1

is considered correct but $(x+1)^2$ incorrect because the problem asks for expanding a polynomial. How about if the answer to the problem is not unique or has more than one form, for example: if the answer of a test question is $2x$, and a student enters $x+x$ or $3x-x$? Or a correct answer we are looking for is x^2, and a student answer was $x*x$ or x^3-x? The checking service will use CAS to decide algebraic equivalence for such problems.

If the answer is "very close" or "almost right", for example: if the correct answer of a test question is 3, and a student enters 2.98 or even 2.99? Will we consider it right

or wrong? How to decide? So here for example, I suggest so-called *tolerance-rate* or *acceptance-rate* of the answer. Such parameter will be given by the user (a teacher, for example), instead of having the system provides a fixed parameter number every time, and a question will be checked and graded accordingly.

9.2 Introduction to DMAS Answer Checking Service

It is important to use a *powerful and sophisticated computer algebra system* for mathematical answer checking that can help solve and overcome problems and difficulties of mathematical answer checking and automatic grading. We have such computer-based algebra systems (CAS) such as *Maxima*, free software that comes as built-in in UNIX systems. DMAS, as experimental work only, uses both client-side and server-side for answer checking purposes. DMAS-MACS uses Maxima for computation as a back-end (on the server-side) computer-based algebra system for the DMAS Answer Checking Service to check answers and automatically grade student answers (Figure 9.1).

9.3 DMAS-MACS Answer Checking Service Methods

DMAS-MACS service uses both client-side and server-side Answer checking to provide a response/feedback to users who try mathematical assessments. It uses *server-side for computing, solving, and grading mathematical problems and client-side for responsive answer checking*. Answer checking service will be invoked by HTTP POST/GET requests and will return true or false, showing the correctness of the input answers, and some additional results and parameters.

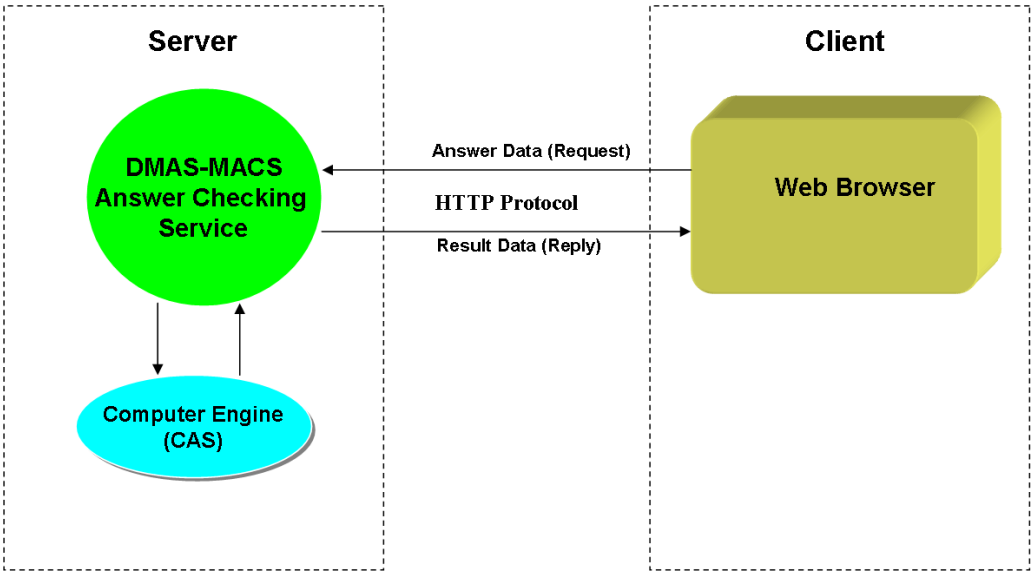


Figure 9.1. DMAS–MACS: A Web-based Mathematical Answer Checking Service

DMAS-MACS service does the following: it checks mathematical expression or formula answers, evaluates the expressions, returns true or false. The DMAS-MACS needs to use DMAS databases to retrieve assessment questions with mathematical formulas. Then it needs to *parse* mathematical formulas and determines mathematical *operation needed to be applied (compute, simplify, derive, integrate, etc.)*, and then to send to CAS (Maxima as a back-end on the Server-Side) for mathematical answer checking.

DMAS-MACS service checks the problem and sends back the answer to the client side to be evaluated and displayed to the user through a Web browser. DMAS-MACS service was *experimentally* implemented in two ways: *Client-Side and Server-Side Synchronization Method (CS-Sync)* and *Client-Side and Server-Side Asynchronization Method (CS-Async).* These two approaches use both Client-Side and Server-Side for mathematical answer checking. However, they are different in which the *CS-Sync method*

requires synchronozation between client and server while the *CS-Async method* does not require such condition.

9.3.1 CS-Sync: Client-Side and Server-Side Synchronization Method

In this approach, DMAS-MACS service displays Mathematical formulas correctly on Web pages using *MathML presentation code*. It receives user inputs as a formula using (MathEdit [51, 52]) and through HTTP protocol, in addition to an *ad-hoc HTML form* to get/post to a server-side program that checks the answer and provides a *response page*. The user input is displayed using MathML and then transmitted to the answer checker with *infix notation code*. Since DMAS questions will be *dynamically and/or automatically generated with no previous knowledge or storage of the correct answer*, the correct answer of a question will need to be computed by the checking program rather than given as part of the answer checking data posted from the client side (Figure 9.2).

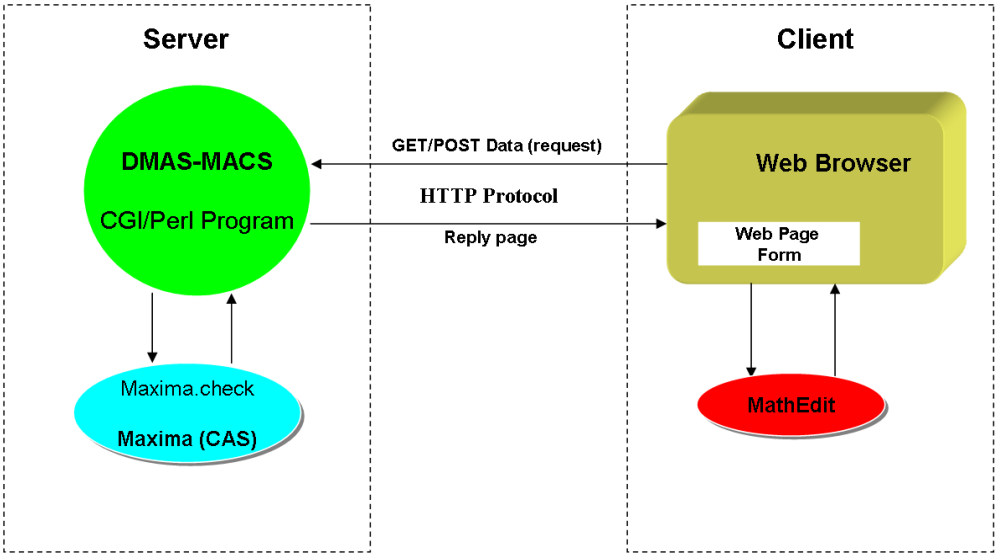


Figure 9.2. Client-Side and Server-Side Synchronization Method (CS-Sync)

If we look closely at *synchronous communications* between client and server and how DMAS-MACS service uses CAS (Maxima) for answer checking, we see a Web browser on the client side sends an *ad-hoc HTML form* with parameters as a request to a Web server on host (server-side). The Web server receives inputs and sends them to the *CGI/Perl program for processing*. The CGI program will process input request, extract parameters, and then invoke CAS (Maxima) program for computation. The CAS (Maxima) program sends back the results to the *CGI answer checker* which in turn checks the answer and provides a *response page* to the Web server which sends the response page back to the Web browser for display (Figure 9.3).

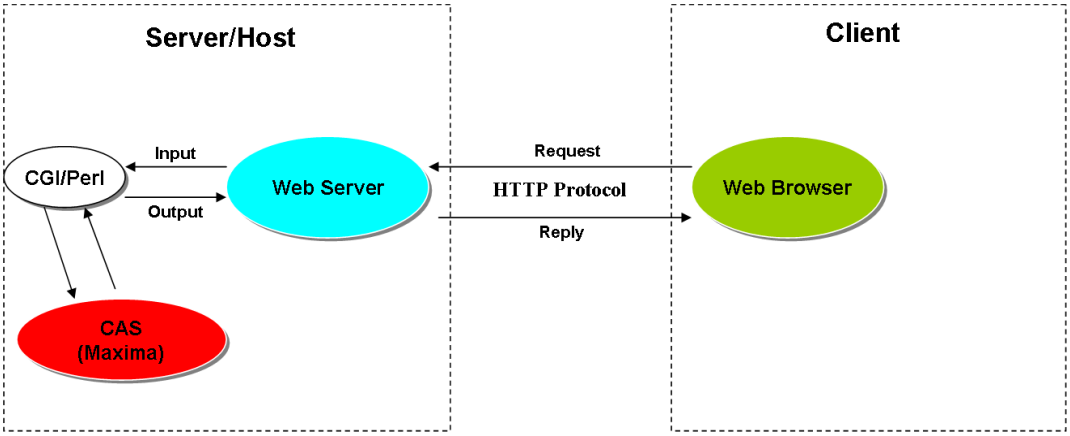


Figure 9.3. DMAS-MACS Using CAS for Answer Checking

From some experimental work I have done, I found some problems with the *synchronous communications* in this approach:

1. Client side has to *wait during all communication time* and do nothing until server side responds, which may take long time and may cause wasting time of client.

2. The result coming back from the server is a *whole new page* causes Web browser to *refresh/reload current page* which means replacing it with the response page.
3. There are too much and unnecessary data transferred between client and server.

9.3.2 CS-Async: Client-Side and Server-Side Asynchronization Method

The problem with the previous approach, *CS-Sync method*, is the *mandatory live connection between client side and server side for any particular answer checking request*. So the second approach, *CS-Async method*, overcomes such problem by using client-side AJAX (Asynchronous JavaScript And XML) and by establishing (future) *Answer Checking Protocol (ACP)* between the client and the answer checking service on the server side. The DMAS-MACS will use REST/JSON for data representation and transfer (Figure 9.4), REST is Representational State Transfer while JSON is considered as a Son of JavaScript and used as a data format.

This approach, *CS-Async method*, overcomes some of the shortcomings of the first approach, *CS-Sync method*. It *releases* both client and server from being coupled or synchronized during communication time. It constructs query string of request parameters on the client side and then uses *AJAX object* to send GET/POST request to the checking service on the server through HTTP protocol. The service processes the request and this time sends back a reply in *JSON format* (object or string), a JavaScript-defined data representation format, to the client which in turn evaluates the JSON reply format *directly using JavaScript* (Figure 9.5).

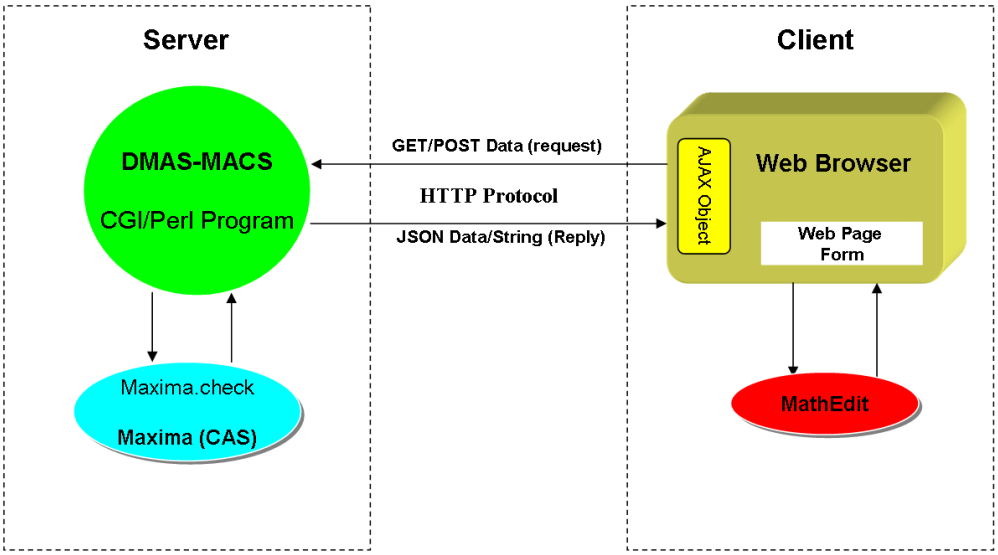


Figure 9.4. Client-Side and Server-Side Asynchronization Method (CS-Async)

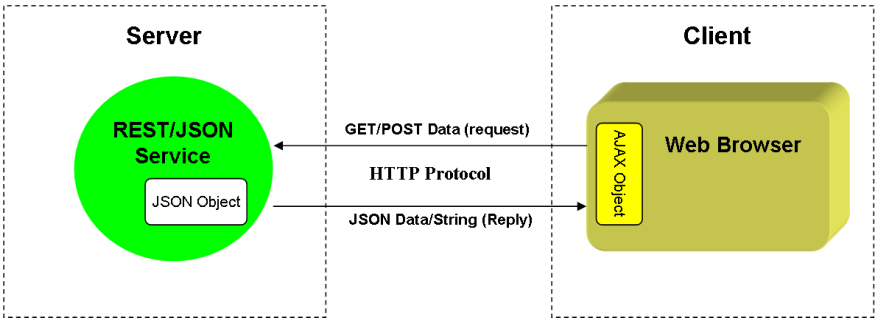


Figure 9.5. Web Browser (AJAX Obj) and JSON Service Interactions

The idea here is that because the nature of HTTP request is *asynchronous* and *stateless*, does not memorize or keep track of the state of interaction or request between client and server, the *CS-Async method* takes advantage of this. So *synchronization between client and server is not really needed* here. Also returning data from the server to

the client will be processed using a *callback function* on the client side and evaluated using JavaScript. Finally, returned and processed data will be used to update the client Web page or (*part of it*).

So from my simple experimental work, I found the *CS-Async method* that can solve some of the problems of *CS-Sync method* and can have the following advantages:

1. The *CS-Async method* is *asynchronous* so client and server do not need to *wait for each other during communication time*.
2. Query/Reply format is based on *standard JavaScript format REST/JSON* which is used to send, receive, and process requests.
3. *No need to refresh/reload the client Web page*. The result coming back from the server is smaller (not *a whole new page*) and Web browser can replace only a small part of the page.
4. Fewer data transfers between client and server.

- <u>Example:</u> Suppose we want to use DMAS-MACS answer checking service (the *CS-Async method*) to check question answer of adding two fractions:

  `1/4 + 4/9`?

  So a DMAS user can create such question using DMAS authoring (Figure 9.6).

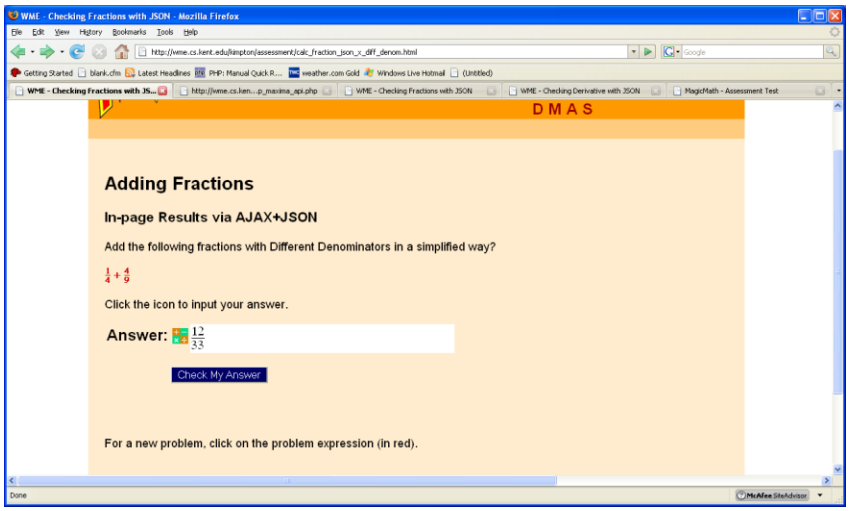


Figure 9.6. An Example of Adding Two Fractions Using CS-Async Method

The *generated MathML presentation code* of the example is shown below for browser display (Table 9.1).

```
<math  xmlns='http://www.w3.org/1998/Math/MathML'>
   <mrow>
        <mfrac>
            <mn>1</mn><mn>4</mn>
        </mfrac>
        <mo>+</mo>
        <mfrac>
            <mn>4</mn><mn>9</mn>
        </mfrac>
   </mrow>
</math>
```

Table 9.1. MathML Presentation Code

The CGI/Perl program receives and processes the request, invokes CAS for computations, makes answer checking, and builds JSON reply format to be send back to the client. Part of the CGI/Perl program with different request parameters: formula infix code, MathML content code, and the expression/formula data are shown below (Table 9.2).

```
#!/usr/bin/perl
##  Fraction Addition checking using JSON service
use CGI qw(:standard);
### obtain form input
$ans = param('answer');
$ansct = param('answerct');
$exp = param('expr');
$var = param('var');
$cmd = "ratsimp";
#### for testing
##$ans = "25/36";
##$ansct="</math><apply><plus></plus><apply><divide></divide>
##<cn>1</cn><cn>4</cn></apply>";
##<apply><divide></divide><cn>4</cn><cn>9</cn></apply></apply</math>
##$exp = "1/4+4/9";
##$var ="x";
$arg = "($exp)-($ans)";
$stdans ="$exp";
```

Table 9.2. A Partial CGI/Perl Program Code of Answer Checking

An example of *JSON data format* returned from the answer checking service (DMAS-MACS) is shown next (Table 9.3).

```
{  status: "normal",
   ans_ct: '<mathml content>',
   correctness: 'no',     ### can be 'yes' or 'close' if very close
   ans_pr: "<mathml presentation>"
}
```

Table 9.3. An Example of JSON Data Format Returned from DMAS-MACS

When a response from the answer checking service arrives, the JavaScript *callback function* on client side processes and evaluates the received JSON reply from the server, and then constructs the appropriate response message to be shown to the user (Table 9.4).

```
function processResponse(responseText)
{
   eval("answer=" + responseText + ";");
   var node=document.getElementById('checkResult');
   if ( answer.correctness=="incorrect" )
   { node.innerHTML="Your answer is incorrect."+
     " The correct answer is "+answer.standard;
   }
   else if ( answer.correctness=="close" )
   { node.innerHTML="Your answer is very close."+
     " The correct answer is "+answer.standard;
   }
   else
   { node.innerHTML = "Congratulation, " +
     "your answer is correct.";
   }
}
```

Table 9.4. A JavaScript Callback Function Code on Client-Side

Finally, the appropriate reply message showing the *correctness* of user's answer (and the correct answer if it was 'incorrect') will be displayed on the Web page for the user without refreshing or reloading the Web page (Figure 9.7).

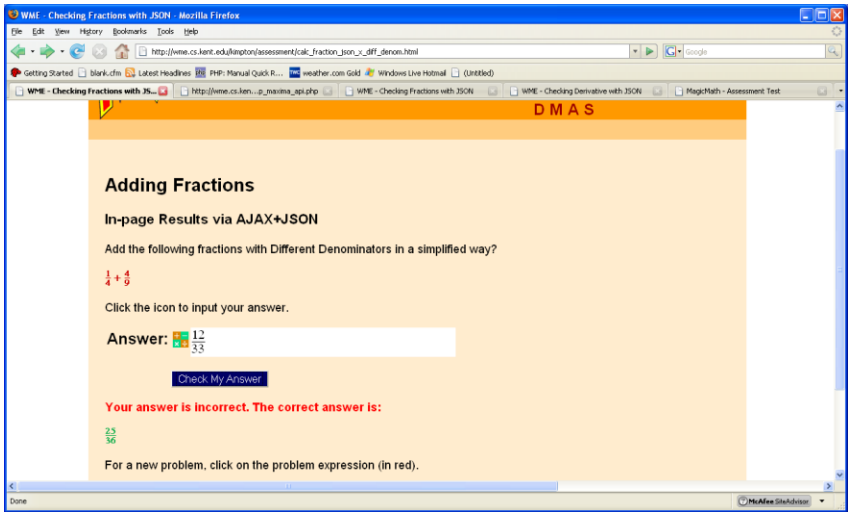


Figure 9.7. A Reply Message Showing the Correctness of User's Answer

PROGRAMMING LANGAUGES AND TECHNOLOGIES USED IN DMAS SYSTEM IMPLEMENTATION

DMAS is an open system that involves Client-Side and Serve-Side programming and implemented with standard Web/Internet/Database technologies and protocols such as:

- HTTP/1.1
- HTML/XHTML
- Apache 2.2 HTTP Web Server
- JavaScript
- DOM / XML DOM
- JSON
- AJAX
- REST
- XML
- CSS
- XSL
- XSLT
- DTD
- XML Schema

- SVG (Scalable Vector Graph)
- MathML (Presentation code and Content code)
- MySQL Databases
- Perl/CGI (Common Gateway Interface)
- PHP (Hypertext Preprocessor)
- CAS (Computer Algebra System)
- CSV (Comma Separator Value)

CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

This dissertation offers the following **contributions**: an alternative and interoperable assessment framework to the existing TWAS systems, a Web-based Distributed Mathematics Assessment System (DMAS) and DMASEngine to search and collect questions and better deliver assessment materials to teachers and students. Another contribution is TSIM technique (Teacher-Student Interaction Mechanism), a *real-time online test supervision tool* that provides instant interactions and communications between teacher and students in a class room to promote interactive testing environment and to be used as a teaching and learning tool. Also the automatic generation of mathematics formulas/questions based on mathematical conditions: Free Parameter Question (FPQ) and Sample Question Generation Feature (SQGF). Finally, DMAS promotes sharing of mathematical assessments widely on the Web through MAML, as a mathematics assessment XML markup language for DMAS.

On the other hand, this work has the following **limitations**: DMAS is still in its early stage and needs many more real trials to measure the effectiveness of this system and its components such as TSIM and DMASEngine. Also TSIM is intended to be used in a class room with the presence of a teacher to monitor student progress but not as E-Learning tool. Automatic answer checking service needs far more investigation and improvement.

Future work will include: *Putting DMAS system under extensive trial in schools and collecting feedback and suggestions* from teachers, students, school administrators and education experts to help evolve DMAS. As more schools adopt DMAS system, the distributed nature of DMAS will be demonstrated in realistic situations. Furthermore, we need to standardize the *Answer Checking Protocol (ACP)*, create a mathematical *Answer Checking Match-up Language*, and improve *Answer Checking Service (DMAS-MACS)*. Finally, *TSIM technique* and *MAML* also need additional features and more improvements.

RELATED WORKSHOPS AND POSTERS

[1] Saleh Al-shomrani participated in the WME tutorial workshop and poster session conference at ECCAD 2008, East Cost Computer Algebra Day, Shepherd University, Shepherdstown, West Virginia, Saturday, May 9-10, 2008. http://www.sigsam.org/bulletin/articles/164/eccad.pdf

[2] Saleh Al-shomrani "DMAS: A Web-based Distributed Mathematics Assessment System", 2008 Annual Open House, Awards Ceremony, and Poster Conference, Computer Science Department, Kent State University, Kent, USA, Friday 11 April 2008.

[3] Saleh Al-shomrani attended and participated in the 2007 Celebration of Scholarship (A&S Poster Session) on September 27, 2007, Kent State University.

RELATED RESEARCH AWARDS

[1] 2008 Annual Open House, Awards Ceremony, and Poster Conference, Computer Science Department, Kent State University, Kent, USA, Friday 11 April 2008. **2008 Poster Conference Winners in the Business / Industry Category, <u>Graduate level: Second Prize</u>:** "DMAS: A Web-based Distributed Mathematical Assessment System by **Saleh Al-shomrani".**
http://www.cs.kent.edu/openhouse2008/winners.html.

[2] One April 11 2005, I was selected by the Graduate Student Senate (GSS) for the **research award for 2005** from Kent State University.

REFERENCES

[1] C. J. Sangwin. *Assessing mathematics automatically using computer algebra and the internet*. Teaching Mathematics and its Applications, 23(1):1–14, 2004.

[2] Saliha Klai, Theodore Kolokolnikov, Norbert Van den Bergh. *Using Maple and the Web to administer mathematics tests*.

[3] Dirk F.M. Hermans. *Embedding Intelligent Web Based Assessment in a Mathematical Learning Environment*. Program and Online Proceedings Internet Accessible Mathematical Computation 2002 Workshop, 7 July 2002 Lille, France.

[4] Brad Lucier. *SAGE: a Homework on the Web System*. Purdue University, USA, 2005.

[5] Klai, S., Kolokolnikov, T., Van den Bergh, N., *Using Maple and the web to grade math-ematics tests*. IEEE International Workshop on Advanced Learning Technologies, 4-6 December 2000, Palmerston North, New Zealand, 2000.

[6] Giesbers, B., van Bruggen, J., Hermans, H., Joosten-ten Brinke, D., Burgers, J., Koper, R., & Latour, I. (2007). *Towards a methodology for educational modelling: a case in educational assessment*. Educational Technology & Society, 10 (1), 237-247.

[7] Peter James Rowlett. *Mathematics teaching & learning and XML*. MSOR Connections Feb 2005 Vol 5 No 1.

[8] George Valvis, Despina Polemi. *An XML-based data model for vulnerability assessment reports*. The IFIP Conference, October 26 – 28, 2005, Poznan, Poland.

[9] IEEE P1671, *Standard for Standard Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML*, July 2005.

[10] Yongwu Miao, Tim Sodhi, Francis Brouns, Peter Sloep, and Rob Koper. *Bridging the Gap between Practitioners and E-learning Standards: A Domain-specific Modeling Approach*, EC-TEL 2008, LNCS 5192, pp. 284–289, 2008.

[11] Suzanna Paul. *Comparative Assessment of the Effectiveness of Online Vs Paper Based*, Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, pp. 271 – 271, 2006.

[12] Bertil Roos. *From Distance Education to Online Learning: Formative Assessment in Higher Education*. EDEN 10th Anniversary Conference - Learning Without Limits: Developing the Next Generation of Education, Stockholm, Sweden 2001.

[13] Lembke, L. O., & Reys, B. J. *The development of, and interaction between, intuitive and school-taught ideas about percent*. Journal for Research in Mathematics Education, 25(3), 237, 1994.

[14] National Council of Teachers of Mathematics. (2000). *Principles and standards for school mathematics*. Reston, VA: Author.

[15] Silver, E. A. Herbst, P.G. *Theory in mathematics education scholarship*. In F. K. Lester (Ed.), Second handbook of research on mathematics teaching and learning (pp. 44-46). Charlotte, NC: Information Age Publishing, 2007.

[16] Baxter, G. P., Shavelson, R. J., Herman, S. J., Brown, K. A., & Valadez, J. R. (1993). Mathematics performance assessment: Technical quality and diverse student impact. *Journal for Research in Mathematics Education, 24*(3), 190-216.

[17] Blank, R. K., & Pechman, E. M. (1995). *State curriculum frameworks in mathematics and science: How are they changing across the states?* Washington, DC: Council of Chief State School Officers.

[18] Romberg, T. A., Zarinnia, E. A., & Williams, S. (1990). *Mandated school mathematics testing in the United States: A survey of state mathematics supervisors.* Madison, WI: National Center for Research in Mathematical Sciences Education.

[19] Ferrini-Mundy, J. & Johnson, L. (1996). Building the case for standards-based reform in mathematics education. In Bartels, S. (Ed.), *Implementing science education reform: Are we making an impact? AAAS Yearbook (36)*. Washington, DC: American Association for the Advancement of Science.

[20] Gay, S., & Thomas, M. (1993). Just because they got it right, does it mean they know it? In N. L. Webb & A. F. Coxford (Eds.), *Assessment in the mathematics classroom*. *1993 Yearbook* (pp. 130-134). Reston, VA: National Council of Teachers of Mathematics.

[21] Michael Mikusa, Paul S. Wang, David Chiu, Xun Lai, Xiao Zou. Web-based Mathematics Education Pilot Project, *Conference on Information Technology in Education*, Elizabethtown College Elizabethtown, PA, September 18, 2004.

[22] P. S. Wang, M. Mikusa, S. Al-shomrani, D. Chiu, X. Lai, and X. Zou. Features and Advantages of WME: a Web-based Mathematics Education System, *Proceedings of the IEEE Southeast Conference, IEEE*, Ft. Lauderdale, FL, April 8-10 2005, pp. 621-629.

[23] P. Wang, S. Gray, N. Kajiler, D. Lin, W. Liao, X. Zou. *IAMC* Architecture and Prototyping: A Progress Report, *Proceedings of ISSAC 2001, International Symposium on Symbolic and Algebraic Computation*, pp. 337-344, July, 2001.

[24] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou. WME: Towards a Web for Mathematics Education, *Proceedings of ISSAC, ACM Press*, August 2003, pp. 258-265.

[25] S. Al-shomrani and P. S. Wang. Building DMAD: A Distributed Mathematics Assessment Database for WME, *Proceedings of the IEEE Southeast Conference*, IEEE, Ft. Lauderdale, FL, April 8-10 2005, pp. 630-635.

[26] Wei Su, Lian Li, Paul Wang. Lesson Page Structure and Customization in WME, *IAMC 2005 Workshop, online Proceedings, July 24 2005*, Chinese Academy of Sciences, Beijing, China.

[27] David Chiu and Paul Wang. An Approach for Interoperable and Customizable Web-based Mathematics Education, *ICM Technical Report, ICM-200509-0006, Sept.*

2005. Also Proceedings, the Fifth IASTED INternational Conference on Web-based Education, Jan. 23-25, 2006, Puerto Vallarta, Mexico, pp. 80-87.

[28] Xun Lai and Paul Wang. An SVG Based Tool for Plane Geometry and Mathematics Education, *IAMC 2005 Workshop, online Proceedings, July 24 2005*, Chinese Academy of Sciences, Beijing, China.

[29] Xiao Zou. Support for Online Mathematics Education: MeML and WME Service, *Proceedings, IEEE Southeast Conference*, IEEE, Ft. Lauderdale, FL, April 8-10 2005, pp. 656-662.

[30] The Education Resources Information Center (ERIC), sponsored by the Institute of Education Sciences (IES) of the U.S. Department of Education. www.eric.ed.gov/. (Last visited: November 2008).

[31] The Eisenhower National Clearinghouse for Mathematics and Science Education www.enc.org/features/lessonplans/math/ (Last visited: October 2008)

[32] Proceedings of the IAMC 1999, 2001, 2002, and 2003 Workshops, icm.mcs.kent.edu/research/ iamc.html#iamcworkshop.

[33] Institute for Computational Mathematics, demos of mathematical computation icm.mcs.kent.edu/research/demo.html.

[34] icm.mcs.kent.edu/research/iamc/ (IAMC homepage), icm.mcs.kent.edu/research/iamcproject.html (IAMC project homepage).

[35] Math Forum, www.mathforum.org/. (Last visited: November 2008).

[36] Max Froumentin, Team Contact for the Math Working Group. *MathML*. www.w3.org/Math. (Last visited: November 2008)

[37] MathML International Conference 2000 and 2002, www.mathmlconference.org.

[38] The National Commission on Mathematics and Science Teaching for the 21st Century. "Before It's Too Late: A Report to the Nation", www.ed.gov/inits/Math/glenn/index.html, 2000.

[39] Annenberg/CPS http://www.learner.org/ (Last visited: October 2008)

[40] National Council of Teachers of Mathematics. Principles and Standards for School Mathematics, Reston, Va. www.nctm.org/standards, 2000.

[41] National Research Council. *Everybody Counts: A Report to the Nation on the Future of Mathematics Education*, National Academies Press, Washington, D.C., ISBN: 0309039770, books.nap.edu/books/0309039770/html/index.html, May 1989.

[42] PBS TeacherSource http://www.pbs.org/ (Last visited: November 2008).

[43] Online resources for mathematics education at the Ohio Resource Center for Mathematics, Science, and Reading. www.ohiorc.org/math/. (Last visited: November 2008).

[44] Center for Curriculum and Assessment. *Academic Content Standards: K-12 Mathematics*, Ohio Department of Education, Feb. 2002.

[45] Blackboard, www.blackboard.com/. (Last visited: November 2008).

[46] The Connexions: http://cnx.org/ (Last visited: November 2008).

[47] WebCT, www.webct.com/. (Last visited: November 2008).

[48] NWREL Assessment http://www.nwrel.org/assessment/ (Last visited: November 2008).

[49] Alberta Assessment Consortium – ACC http://www.aac.ab.ca/home.html. (Last visited: November 2008).

[50] P. Wang, M. Mikusa, S. Al-shomrani, X. Lai, X. Zou, D. Zeller "*WME: a Web-based Mathematics Education System for Teaching and Learning*", ICME 11-TSG 22 Theme 3 the l1the International Congress on Mathematical Education, Monterrey, Mexico, July 6 -13, 2008.

[51] Wei Su, Paul S. Wang, Lian Li *An On-line MathML Editing Tool for Web Applications* 2nd International Multi-Symposiums on Computer and Computational Sciences 2007 (IMSCCS|07), The University of Iowa, Iowa City, Iowa, USA, pp. 458-464.

[52] Wei Su, Paul.Wang, Lian Li, Guanyu Li, Yanjuan Zhao. *MathEdit, A Browser-based Visual Mathematics Expression Editor*, Proceeding of The 11th Asian Technology Conference in Mathematics, HongKong, December 2006, pp. 271-279.

[53] Xun Lai and Paul Wang. *GeoSVG: A Web-based Interactive Plane Geometry System for Mathematics Education*, Proceedings, the 2nd IASTED International Conference on EDUCATION AND TECHNOLOGY (ICET), July 17-19, 2006, pp. 5-10.

[54] Saleh Al-shomrani and Paul Wang. *Design and Implementation of an Assessment Database for Mathematics Education*, Proceedings, the 2nd IASTED International Conference on EDUCATION AND TECHNOLOGY (ICET), July 17-19, 2006, pp. 173-179.

[55] Xiao Zou and Paul Wang. *MEML: Supporting Structured, Interoperable and Dynamic Web-based Mathematics Education*, Proceedings, the 2nd IASTED International Conference on EDUCATION AND TECHNOLOGY (ICET), July 17-19, 2006, pp. 113-120.

[56] D. Chiu. *Customization and Interoperability in WME*, Proceedings of the IEEE Southeast Conference, IEEE, Ft. Lauderdale, FL, April 8-10 2005, pp. 636-640.

[57] David Chiu. *WME Site Organization and Customization Support*, Proceedings of the 2004 Conference on Information Technology in Education (ITE'04), Elizabethtown College, Elizabethtown, PA, September 18, 2004, pp. 110-115.

[58] David Chiu. *Web-based Mathematics Education with MathChat*, Proceedings of IEEE/ITCC'2004, April 5-7 2004, Las Vegas, Nevada, pp. 709-717.

[59] Paul S. Wang, Yi Zhou and Xiao Zou. *Web-based Mathematics Education: MeML Design and Implementation*, Proceedings of IEEE/ITCC'2004, April 5-7 2004, Las Vegas, Nevada, pp. 169-175.

[60] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou. *WME: Towards a Web for Mathematics Education*, Proc., ISSAC'2003, ACM Press, August 2003, pp. 258-265.

[61] Paul S. Wang, Norbert Kajler, Yi Zhou, and Xiao Zou. *Initial Design of A Web-Based Mathematics Education Framework*, (workshop paper at IAMC'2002).

[62] P. Wang, Q. Guo and W. Liao. *Mathematics over the Internet/Web: A Protocol-based Approach*, (Proc., International Conference on Parallel and Distributed

Processing Techniques and Applications (PDPTA'02), June 2002, pp. 2190-2196).

[63] Paul S. Wang, Yi Zhou, and Xiao Zou. *Mathematics Education Markup Language*, (poster and demonstration Proceedings of E-Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education October 15-19, 2002; Montréal, Canada).

[64] David Chiu, Yi Zhou, Xiao Zou, and Paul S. Wang. *Design, Implementation, and Processing Support of MeML*, IAMC'03

[65] Saleh Al-shomrani and Paul Wang "*DMAS: A Web-based Distributed Mathematics Assessment System*", International Conference on Learning 2008, University of Illinois at Chicago, USA, June 3-6, 2008.

[67] Questions that encourage written communication in math (from WASL): http://www.pasd.wednet.edu/school/mathWASL/ (Last visited: November 2008).

[68] At the Vermont Institute: (Last visited: November 2008). http://www.vermontinstitutes.org/assessment/data/math/index.htm

[69] From the Texas Education Agency: (Last visited: November 2008). http://www.tea.state.tx.us/student.assessment/resources/release/index.html

[70] Mathematics Classes from the MIT OpenCourseWare Project: http://ocw.mit.edu/OcwWeb/Global/all-courses.htm#Mathematics. (Last visited: November 2008).

[71] At the Mathforum at Drexel: http://mathforum.org/library/ed_topics/assessment/ (Last visited: November 2008).

[72] At the National Center for Educational Statistics (the US Education Department): http://nces.ed.gov/nationsreportcard/ITMRLS/search.asp?picksubj=Mathematics (Last visited: November 2008).

GLOSSARY

AJAX	**A**synchronous **J**avaScript **A**nd **X**ML
ACP	Answer Checking Protocol
CAS	Computer Algebra System
CGI	Common Gateway Interface
CS-Async	Client-Side and Server-Side Asynchronization Method
CSS	Cascading Style Sheets
CS-Sync	Client-Side and Server-Side Synchronization Method
CSV	Comma Separator Value
DFF	DMAS File Formats
DFFS	DMAS File Format Service
DMAD	A distributed database with *local databases* at different school sites.
DMAS	A Web-based Distributed Mathematics Assessment system.
DMASEngine	DMAS search engine
DMAS-MACS	A Web-Based Mathematical Answer/Expression Checking Service for DMAS
DOM	Document Object Model
DTD	Document Type Definition

FPQ	Free Parameters Question
FTP	File Transfer Protocol
geoSVG	A Web-Based Authoring tool for geometry in WME
HTTP	Hypertext Transfer Protocol
ICM/Kent	The Institute for Computational Mathematics at Kent State University
JSON	JavaScript Object Notation
MAML	Mathematics Assessment Markup Language
MathEdit	A Web-based Math editor in WME
MathML	Mathematical Markup Language
MeML	Mathematics Education Markup Language
MySQL	Relational database management system (RDMS)
PHP	Hypertext Preprocessor
REST	Representational State Transfer
SCP	Secure Copy Program
SFTP	Secure File Transfer Protocol
SMAD	School Mathematics Assessment Database
SQGF	Automatic Sample Question Generation Feature
SVG	Scalable Vector Graphics
TCP	Teacher Control Panel
TLP	Topic Lesson Page
TM	Topic Module

TMAD	Teachers Mathematics Assessment Database
SI	TSIM Student-Interface
TSIM	Teacher-Student Interaction Mechanism
TWAS	Traditional Web Assessment Systems
WME	A Web-based Mathematics Education System
XML	Extensible Markup Language
XML DOM	XML Document Object Model
XSL	The Extensible Stylesheet Language
XSLT	The Extensible Stylesheet Transformation Language