Mobile Interactive Problem Solving for Active Teaching and Learning

M. Muztaba Fuad, Debzani Deb and Shequi'lla Whitaker

Department of Computer Science
Winston-Salem State University
Winston-Salem, NC 27110, USA.
{fuadmo, debd}@wssu.edu

Abstract—**In traditional classroom setting, lecturing is the prominent method of transferring knowledge to students. However this passive mode of instruction does not keep student engaged and interested longer in the class. Although mobile devices are used extensively in everyday life and have a bad reputation of creating distraction during class, it can be used to make student's class participation more interactive and engaging. This paper presents one such approach, which is fundamentally different than existing ones by allowing students to participate in active learning during class session by using their mobile devices. This research envision interactive problem applications which will run in student mobile devices, which faculty can initiate remotely and where student can actively participate to solve problem in a hands-on manner. This paper presents the development of a Karnaugh map interactive mobile application and its incorporation into Mobile Response System.**

***Index Terms*—Active learning, Interactive problems, Hands-on activities.**

I. INTRODUCTION

In a traditional face-to-face classroom setting, if faculty is only using lecturing to deliver course material, they will loose attention of many students, minutes after the class has begun. In recent years, classes in Science, Technology, Engineering and Mathematics (STEM) courses faced a significant learning barrier, which is the problem of students' inability to keep engaged and interested in classroom. Particularly the proliferation of mobile devices in daily life and its use during the class makes this problem exacerbated. In this research, we explore the applicability of using mobile devices in the classroom and incorporation of interactive problem solving using those devices to increase class engagement and active learning for students. By allowing the students to solve hands-on problems in their preferred devices will create a friendly learning environment where students want to remain, be active and be skill full. This paper presents one such interactive problem (IP) application (app) designed to be deployed using the Mobile Response System (MRS) [1]. This app facilitates Karnaugh map (k-map) simplification for students by allowing them to interact with the problem at hand. Faculty can define the k-map problem using a server side tool and use the MRS backbone to initiate a hands-on session during class time in student's mobile devices. This paper presents ways to develop IP app and ways to incorporate that IP in the MRS software.

II. BACKGROUND

There has been a plethora of research work performed to address the lack of student engagement and motivation during class time. However, the research presented in this paper is different because along with providing a mobile interactive problem-solving platform it also provides a teaching methodology that goes hand-in-hand with this software tool.

A. *Teaching Methodology*

To improve student learning in the STEM disciplines, we need evidence based teaching and learning methods [2]-[4], which is based on constant monitoring of outcomes from assessment and resulting continuous improvement of the applied method. Study shows that in order to improve student learning in the STEM disciplines, traditional pedagogical approaches are not enough to transfer critical knowledge to students. A feedback driven, evidence based teaching and learning technique has to be devised and implemented in our classes to transfer knowledge to students. One of the key factors of such evidence based teaching methodology is for students to actively participate in class and receive instant feedback about their performance in such hands on activities. In that regard, evidence based teaching methodology was incorporated and assessed for couple of semesters. The findings [5] from that study and student's use of mobile devices during class time has encouraged us to developed this interactive problem solving teaching methodology. As several other [6]-[7] research works have shown that using technology during class can improve student's learning abilities, we are confident that the presented teaching tool will also have positive impact on student in-class engagement and problem solving skills.

B. *Mobile Response System*

To facilitate interactive problem solving in real-time manner, a comprehensive backend system is necessary. To satisfy requirements for interactive problem solving, a Mobile Response System is developed, which is independent of any interactive problem or its domain and transparent to the end user. MRS is a client-server software that allows the faculty to dynamically prompt the students with interactive problems synchronized with the lecture material in their mobile computing devices (Clients). Students are then able to actively interact with the problem through multiple steps and send their work back to the faculty computer (Server). MRS then performs grading of the exercises, by using the provided rubric. The other important features supported by MRS are immediate and context-sensitive feedback, anonymous question, polling, summarized grading etc. More information about MRS can be found in [1] and [8].

C. *Interactive Problem (IP)*

An IP is one, where students have to devise the answer following a set of steps and by following a particular algorithm/process. In each step, students have to make key choices that will have impact on the next step of the interaction. During these interaction steps, students can go back and forth and change their answer. This will allow them to see what is the affect of different selection on the result and how every piece fits together. Problems can be started bottom up or at the middle to give students different perspective on the problem and assess their problem solving skills. Only after the student traverse each of the steps or the allotted time to answer a problem runs out, the result of their interaction is sent back to the server as the answer of that student. Each problem has a rubric that not only grade correct answers but also partial answers to gauge student's problem solving skills and thinking models. Each IP has two different components. One at the server side to define the problem and other at the client-side to allow students to interact with the problem. In the server side, the intended problem and the rubric have to be defined by using the MRS standard protocol. On the client side, an app needs to be developed to allow corresponding interactive-ness. MRS provides corresponding Application Programmer Interface methods to convert user provided definition into the corresponding MRS format. Different parts of the format are as follows:

- *Question type*: It must match the Android package name of the client side interactive app and the package name of the server side program for that IP.
- *Question number*: Any number or word, as long as there is no other definition with the same question number.
- *Time to Answer*: Have to be expressed in seconds (which in turn is converted to milli-seconds for better time keeping) with the lower limit of 0 and with the upper limit of the maximum value allowed in a long variable.
- *Instruction/Description*: Any instruction or description associated with this IP. This is optional.
- *Problem Components*: Any number of components needed to render the corresponding IP app on the client.
- *Rubric Components*: Any number of components needed to process the answer of the student and what background information from the IP app should be monitored. The number of problem component and the number of rubric component can be different and has no upper value.

D. *Related Work*

Different studies [9]-[10] have found the benefits of using mobile devices in classrooms and in recent years, there has been a plethora of work [11]-[12] performed to incorporate mobile devices in classrooms. However, these works are traditional in nature and lack the interactive-ness needed in todays engineering classrooms. In Science and Engineering courses, students need to actively solve problems by interacting with the problem in a hands-on approach. Students cannot develop skills such as synthesizing a problem and developing critical thinking only by using multiple-choice or true-false questions. It is our belief that by presenting the problems as interactive entities, where students can actively play with the problem to solve it; we will be able to improve student's critical thinking and problem solving skills. Although there has been a lot of work on algorithm animation and visualization, there are hardly any [13] interactive tools for supporting k-map simplification as presented in this paper.

III. INTERACTIVE PROBLEM APP

To utilize MRS for facilitating interactive problem solving, one needs to follow several sequences of steps and some simple rules as shown in Figure 1 and described in the following sub-sections. Although the K-Map problem is used as an example to describe the process, the MRS software is developed independent of any problem domain. This allows it to be used by any one as long as they follow the described process and develop their own interactive problem and its rubric.

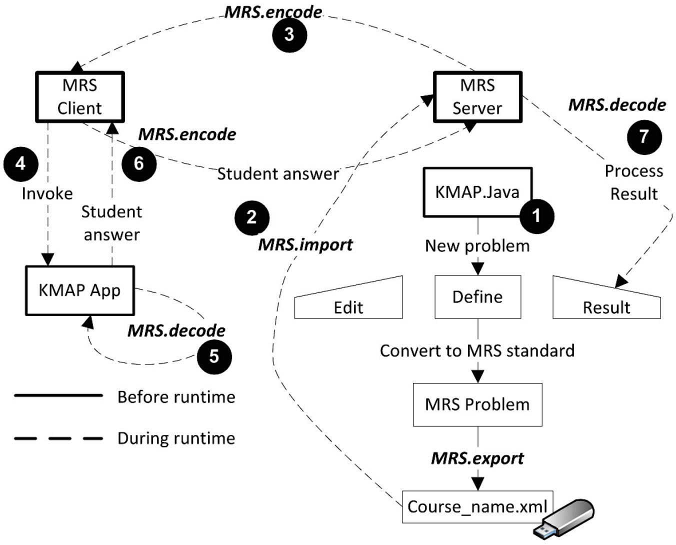
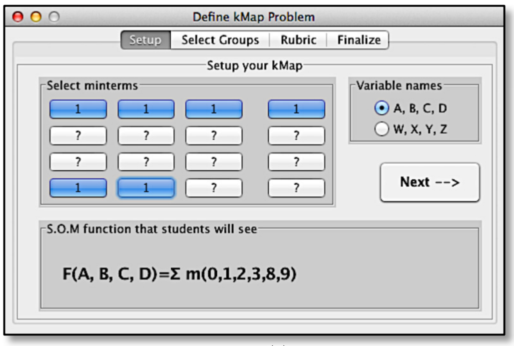


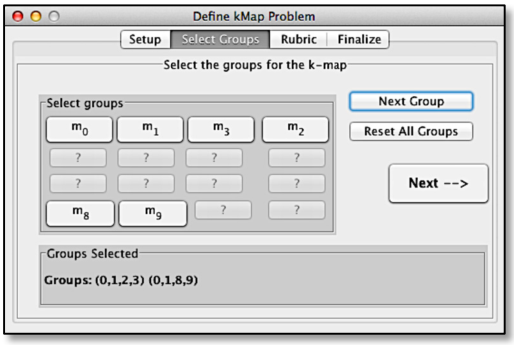
Figure 1. Component organization and workflow.

- *Step 1*: In this step, the IP has to be defined following the MRS protocol as described in Section II.C. If a server side program is written by the user to define the IP instead of defining it by hand, following rules needs to be followed:
    - The program should be in a package with the same name as the package of the IP app and the corresponding question type.
    - The program should have three methods, *define*, *edit* and *result* so that MRS can dynamically call them for corresponding functionality; i.e. *define* to define a new problem, *edit* to edit an existing definition and *result* to process student answers and produce corresponding graphical representation.
- *Step 2*: Once the IP is defined and saved in the storage as a XML representation, it can be imported to MRS by using the MRS GUI, which uses ***MRS.import*** to read the IP definition on the file. One file can have multiple types of IPs in it with unique question numbers.
- *Step 3*: Faculty uses the broadcast button in MRS to broadcast the IP definition to clients and MRS uses the ***MRS.encode*** to broadcast it to all available clients. In encoding and broadcasting, MRS does not include the rubric components of the definition.
- *Step 4*: Once the MRS client receives the encoded definition, it finds the corresponding IP app (by matching the question type with the app package name) and invokes it with the encoded definition as an Android Intent [14] with the key "QUESTION".
- *Step 5*: Once the IP app is invoked, it needs to get the encoded question with the key "QUESTION" and then use the ***MRS.decode*** method to convert it to the workable definition format.
- *Step 6*: Once student submit an answer or the time to answer the problem runs out, the IP app will quit and return the student response/result as Android Intent to MRS with the key "ANSWER". Once MRS client receives that Intent, it forwards that contained message back to the server using ***MRS.encode.***
- *Step 7*: MRS server locates the server side program to process the result and invokes the *result* method.

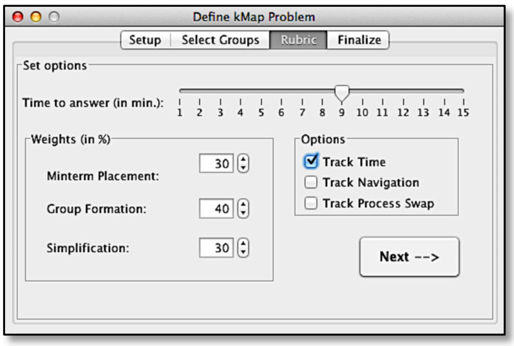
A. Server Side



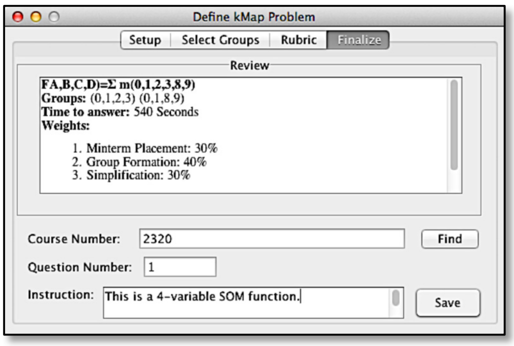
(a)



(b)



(c)



(d)

Figure 2. Defining the k-map IP.

For the k-map IP, at the server side a GUI-based program is designed for easy design of a problem. It follows a sequence of intuitive steps for the user to define a K-map IP. As shown in Figure 2 (a), faculty can choose their min-terms for the expression to be simplified and then choose the valid groups (Figure 2 (b)), set different parameters for problem solving and how it will be graded (Figure 2 (c)) and finally can review (Figure 2 (d)) the problem before saving it to the file. In each step, corresponding validity checks are made to make sure faculty is producing a correct k-map problem. For instance, when selecting groups, faculty will be prompted if the grouping does not follow any allowed rules of the algorithm.

Once the faculty saves the IP definition, it looks something like Figure 3. MRS uses a simple XML schema, which is space saving and faster to decode and manipulate. If faculty do not want to design a GUI, they can use any text editor to define their IP following the definition scheme and the XML schema.

```
<q_type>edu.wssu.kmap</q_type>
<q_num>2</q_num>
<q_time>540000</q_time>
<q_inst>Sample kmap problem</q_inst>
<comp_num>2</comp_num>
    <comp>SOM 0, 1, 2, 3, 8, 9</comp>
    <comp>Variable Names: A,B,C,D</comp>
<comp_num>7</comp_num>
    <comp>Groups |(0,1,2,3) (0,1,8,9)</comp>
    <comp>Placement weight: 30%</comp>
    <comp>Grouping weight: 40%</comp>
    <comp>Simplification weight: 30%</comp>
    <comp>Track student time: true</comp>
    <comp>Track student navigation: false</comp>
    <comp>Track process swap: false</comp>
```

Figure 3. IP definition as saved in the file.

Now, faculty can use the MRS server side GUI to import that problem definition and broadcast it to the available clients as shown in Figure 4. Once all student responses is received back from the clients, either a set of standard statistical tools (average, standard deviation, frequency etc.) within MRS will be used to produce the graphical representation of the student responses or the *result* method of the user definition program will be called to produce corresponding representation.

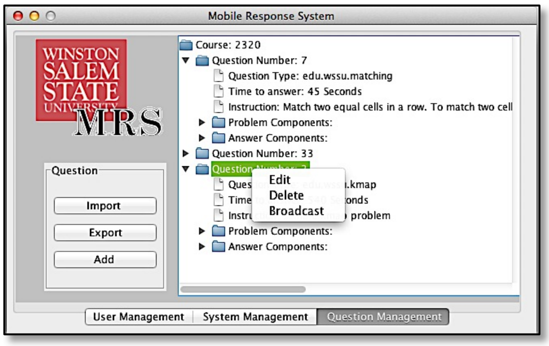


Figure 4. MRS GUI to manipulate IP definition.

B. Client Side

On the client side, students run the MRS client and use their credentials to log into the system. Once they do that, they are presented with the screen as shown in Figure 5. Students can also send anonymous questions or feedback using the client. Once a problem definition is received by the client, it finds the corresponding IP app and pass the definition to it. As shown in Figure 6, student is prompted with the given problem and the app allows the student to traverse back and forth to interact with the problem and solve it within the given time. At first, students can select cells corresponding to the equation (Figure 6(a)). This tests, whether students learned how the map is laid out and what every cell points to. Once this step is finished, students make groups with the cells following the minimization algorithm (Figure 6(b)). Students can select any cell and each group made will take a different colour (with the group number as superscript) to distinguish it from other groups. This will test whether student understands different characteristics of making groups. Next, students interact with each group and select variables from each group (Figure 6(c)) following the minimization algorithm. This tests whether students fully comprehend the algorithm. The app checks for algorithm validity (such as grouping) with student responses but since the rubric is not sent to the clients, can never tell the student whether his/her answer is correct or not. That way, students cannot figure out the correct answer by iterations. Only after all students submitted their answer, the faculty will be able to show them the correct answer. It is completely up to the developer of such IP app on how much interaction and how many screens they want the student's to encounter during the allotted time.

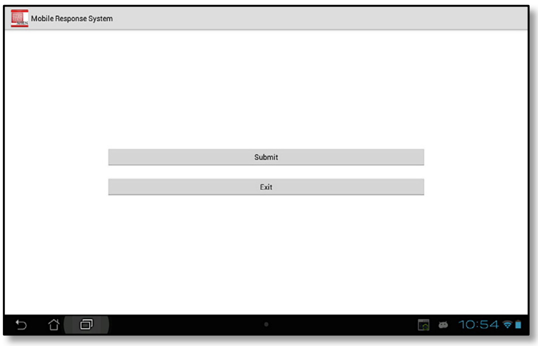
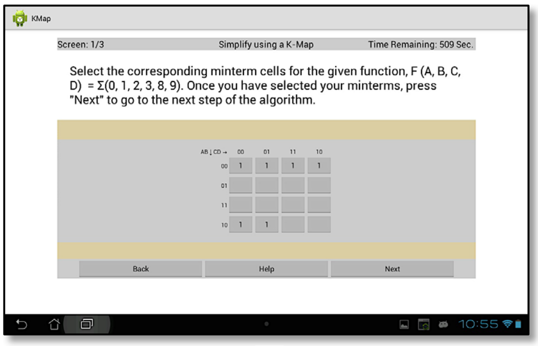
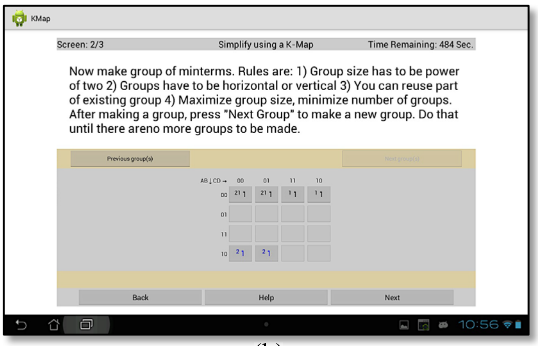


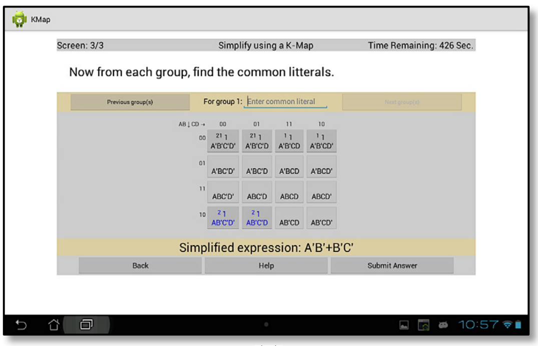
Figure 5. MRS Client.



(a)



(b)



(c)

Figure 6. IP interactions.

C. *Evaluation Plan*

One sophomore level hardware organization course (offered in both semesters) is selected to assess the system in Fall of 2014. This course is a good candidate for evaluation of this approach because it assess mathematics and problem solving skills and we can incorporate several kinds of interactions during the class to enhance these skills. Also, student performance and course evaluation data for last couple of years are available to compare with the new data after deploying the new methodology. Both formative and summative evaluation is planned with the help of data collection, pre and post tests, student surveys, performance data comparisons and focus groups.

IV. Conclusions

Interactive problem solving during class time will allow students to engage and participate more on the class activities and retain more of the lecture content. This paper presents an interactive problem app and its associated components and its incorporation into the Mobile Response System software. Although not presented, software quality matrices for the developed interactive problem app are measured and they fall within acceptable range. Evaluation plan is presented which will be implemented in the future.

Acknowledgment

This research was supported by National Science Foundation grant #1332531.

References

[1] M. M. Fuad and D. Deb, "Design and Development of a Mobile Classroom Response System for Interactive Problem Solving", Accepted for publication at the 26th International Conference on Software Engineering and Knowledge Engineering, Canada, 2014.
[2] D. Perry, "How The Brain Learns Best", *Instructor Magazine*, 2000.
[3] B. K. Saville, "Using Evidence-Based Teaching Methods to Improve Education", *Excellence in Teaching*, vol. IX, 2010, pp. 48-54.
[4] J. Fairweather, "Linking Evidence and Promising Practices in Science, Technology, Engineering, and Mathematics (STEM) Undergraduate Education: A Status Report for The National Academies Research Council Board of Science Education", Washington, DC, 2010.
[5] M. M. Fuad, D. Deb and J. Etim, "An Evidence Based Learning and Teaching Strategy for Computer Science Classrooms and its Extension into a Mobile Classroom Response System", Accepted for publication at the 14th IEEE International Conference on Advanced Learning Technologies, Athens, Greece, 2014.
[6] Mockus, L. and Edel-Malizia, S., "The Impact of Mobile Access on Motivation: Distance Education Student Perceptions", 17th Annual Sloan Consortium International Conference on Online Learning, 2011.
[7] Jones, A., & Issroff, K., "Motivation and mobile devices: exploring the role of appropriation and coping strategies", Research in Learning Technology, *Vol. 5*, No. 3, 2007.
[8] D. Deb and M. M. Fuad, "Use of mobile application to improve active learning and student participation in the computer science classroom", 45th ACM technical symposium on Computer science education, 2014.
[9] Roberts, J., Harvesting fragments of time. Mobile learning pilot project. Technical report, McGraw-Hill, 2003.
[10] Jeffrey R Young, Mobile College App: Turning iPhones Into 'Super-Clickers' for Classroom Feedback, Chronicle of higher education, 2008.
[11] Whattananarong, K., "An Experiment in the Use of Mobile Phones for Testing at King Mongkut's Institute of Technology North Bangkok", Intl. Conference on Making Education Reform Happen: Learning from the Asian Experience & Comparative Perspectives, 2004.
[12] Meisenberger M., Nischelwitzer A., "The Mobile Learning Engine (MLE) – a mobile computer-aided, multimedia based learning application", Multimedia Applications in Education Conference, 2004.
[13] Action Based Learning for Switching and Automata Theory, http://vlsi.colorado.edu/~mooni/N_ABLE/N_ABLE.html.
[14] Android Intent, http://developer.android.com/reference/android/content/Intent.html.