

# Using Speech Recognition in Learning Primary School Mathematics via Explain, Instruct and Facilitate Techniques

Ab Rahman Ahmad\*, Sami M. Halawani, Samir K. Boucetta

Faculty of Computing and Information Technology-Rabigh, King Abdulaziz University,  
Rabigh, KSA

Email: [hayinsuh@yahoo.com](mailto:hayinsuh@yahoo.com), [dr.halawani@gmail.com](mailto:dr.halawani@gmail.com), [sameerboucetta@gmail.com](mailto:sameerboucetta@gmail.com)

Received 24 February 2014; revised 20 March 2014; accepted 28 March 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The application of Information and Communication Technologies has transformed traditional Teaching and Learning in the past decade to computerized-based era. This evolution has resulted from the emergence of the digital system and has greatly impacted on the global education and socio-cultural development. Multimedia has been absorbed into the education sector for producing a new learning concept and a combination of educational and entertainment approach. This research is concerned with the application of Window Speech Recognition and Microsoft Visual Basic 2008 *Integrated/Interactive Development Environment* in Multimedia-Assisted Courseware prototype development for Primary School Mathematics contents, namely, single digits and the addition. The Teaching and Learning techniques—Explain, Instruct and Facilitate are proposed and these could be viewed as instructors' centered strategy, instructors'—learners' dual communication and learners' active participation. The prototype is called *M-EIF* and deployed only users' voices; hence the activation of Window Speech Recognition is required prior to a test run.

## Keywords

Explain, Instruct and Facilitate Techniques, Multimedia-Assisted Courseware, Primary School Mathematics, Visual Natural Language, Window Speech Recognition

---

## 1. Introduction

Teaching and Learning ( $T$  &  $L$ ) is an activity or process in connection with the dissemination of knowledge or

---

\*Corresponding author.

specific skills. It covers planning, management, delivery, supervision and evaluation in order to effectively disseminate knowledge. The model [1] divides *T & L* process into Teaching Objectives, Available Knowledge, Teaching Method and Performance Evaluation as shown in Figure 1.

The model in Figure 1 indicates that *T & L* includes a range of decisions and practices which might require personal contacts between instructors-learners; however, instructors' personality is not the central element in *T & L*. The use of technological devices, team teaching and non-graded instruction will definitely modify the nature of contacts between instructors-learners. Depending on the requirement of the *T&L* situations, particularly on the knowledge available, the future classroom will provide for rather different personal contact to the present conventional classroom. The model implies a greater emphasis on instructors' competence rather than personal charisma; however, it is useful to have these combinations.

The facilitation technique [2] is using questions. Previous research has shown that questioning is a key strategy that facilitators use to promote discussion in Problem-Based Learning (*PBL*). In this study, different types of questions is examined that experienced facilitators asked to promote discussion of teaching problems in professional development for science teachers. *PBL* sessions facilitated by three pairs of experienced facilitators are recorded. Data analysis showed that facilitators asked a set of questions to initiate and solicit ideas, to reframe ideas, to clarify ideas, to push for elaboration, to check for interpretation, and to connect to teachers' classroom practice. This study has implications for the development of *PBL* facilitators.

*T & L* strategies can be classified as three key elements: instructors' centered, concentration and learners' centralization learning. These strategies are sometimes intertwined to each other in *T&L* process. The 5E model [3]—Engage, Explore, Explain, Elaborate and Evaluate—is modified by inserting a conscious pause in a learning cycle. It is called as an express phase to assess and ensure that learners progress adequately through early phases of the cycle. Ultimately, this revised cycle enables learners to meet the standards addressed in a particular lesson by providing differentiated opportunities.

Various technologies have been used to convey knowledge and the use of checklist [4] is common in Teaching and Learning (*T & L*) conventional qualitative studies. It is pointed out that the problem must be translated into mathematical terms and mathematical language before it is completed as the concepts contained in the structure of the problem. Some difficulties in *T & L* may arise, however, problems and difficulties [5] [6] might be overcome by using Information and Communication Technology (*ICT*). *ICT* environment [7] is urged to become personalized for *T&L* and it should be a full multimedia with an almost perfect online for community. *T & L* might be delivered in various forms and modes, such as entertainment and games. This makes *T & L* process [8] more interesting, interactive and fun.

The study [9] discussed on teachers' educational beliefs, namely, constructivist and traditional as antecedent of computer use while controlling for the impact of technology-related variables in computer experience and general computer attitudes with demographical variables of sex and age. A multilevel modelling has been used in identifying differences in determinants of computer use in the classroom. For measuring primary teachers' use of computers to support *T & L* process, a modified version of the Class Use of Computers scale [10] was used. The study supports the hypothesis that teacher beliefs are significant determinants in explaining reasons for adopting computers in the classroom. The impacts of computer experience, general computer attitudes and gender have shown positive constructivist beliefs on the classroom use of computers as opposed to traditional beliefs.

Multimedia is viewed as a dynamic approach when absorbed into education sectors for producing a new *T&L* concept with a combination of educational and entertainment approach. The rapid development in mobile computing, digital memory, internet resources, audio, video transmission, virtual imaging and wireless communication have created new possibilities for the use of technology in *T & L*. Multimedia technology [11] which com-

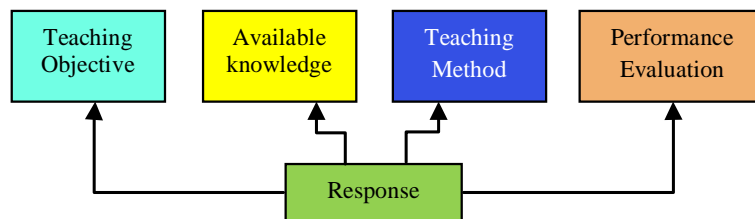


Figure 1. Robert Glaser Model.

bines computer technologies, compact disc players; video and audio systems yield a better enhanced interaction among end users. Interactive multimedia in the context of education has played a vital role in developing a *T & L* process towards a more dynamic and quality. Moreover, this technology creates a big and deep impact in the field of communication and education by representing numbers with pictures and animations. This is supported by computer abilities in presenting information and also *T&L* applications [12]. To date, *ICT* is often associated with education where the technology provides various facilities in *T & L* process and increase learners' interest in the subject being taught.

The goal in [13] is to acquire how multimedia video case studies can support the professionalization of primary school Mathematics educators. The use of multimedia is investigated to support educators in learning to mathematize, didactize and to learn how to use multimedia with students-teachers. The research study has an exploratory character; presents a framework for the use of multimedia as a tentative answer, grounded in the researchers' experiments and design activities. Finding from the study is of one course result in a six-step framework for working with multimedia cases.

Building an accurate sign language recognition system [14] is of a great importance in order to efficiently facilitating communication between normal and deaf communities. The study is the development of speech recognition for Arabic Sign Language using Window Speech Recognition (*WSR*). The grammar is developed and embedded in Microsoft Visual Basic 2008 (*MSVB2008*). The Arabic alphabets and single numbers with their respective signs and avatar images are stored in different directories and invoked upon receiving a word uttered by a user. The system is a visual natural language model that might be used for communication by both groups in Arabic speaking countries.

Automatic speech recognition has now become a great benefit both to normal people and those with various disabilities. The study in [15] described *T & L* in an application of *Talk Maths* which used the output from a commonly-used conventional automatic speech recognition system. It enables users to dictate mathematical expressions in a relatively straightforward way. These are then converted into electronic formats and embedded in a document to be displayed in an editor or web browser. The process is used for preparing Mathematic teaching materials for online tests. By this, the learning should be relatively straightforward for users whose do not have extensive knowledge on computer or mathematics. The way in which the spoken mathematical expressions are analyzed, converted and encoded is a novel approach.

The current research [14] [15] shows the Speech Recognition application in some disciplines and with right *T & L* pedagogy and techniques, this surely will enhance the education for everyone. Though mathematics is seen as an abstract of study such as quantity, structure, space and change, it is feasible to be transformed into a *WSR*-assisted courseware for *T & L*. The presence of an Integrated/Interactive Development Environment (*IDE*) provides comprehensive facilities to computer programmers for software or courseware development. This research is concerned with the development of Primary Mathematics courseware prototype using Explain, Instruct and Facilitate Techniques (*M-EIF*) via *WSR* and *MSVB2008* as *IDE* platform. The single digits and basic facts addition for these digits are used as the test bed materials for prototype development. The prototype could only be run by users' voice uttering the selected words.

## 2. Materials and Methods

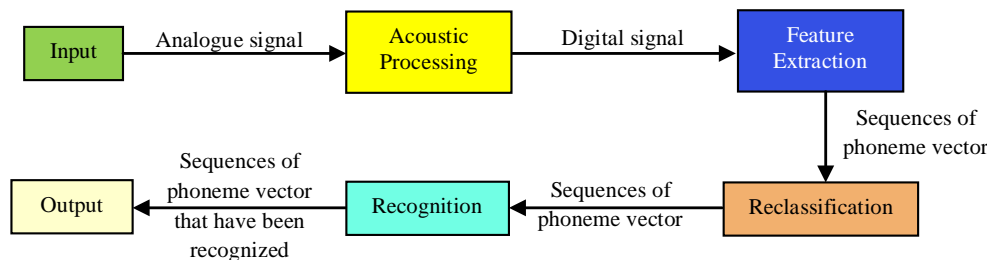
### 2.1. Windows Speech Recognition

In general, a speech recognition process [14] consists of processing the speech which is in acoustic, extracting feature and recognizing the speech. The process is based on a model as shown in **Figure 2**.

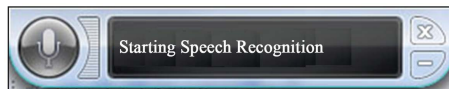
In this research, Windows 7 Speech Recognition [16] is applied since users' voice can be recognized automatically by the system. Users are required to perform the following training in order to acquire the maximum possible recognized speech. In the system, a voice can be used to control a computer by simply saying commands that a computer responds to and also dictating texts to it. A microphone should be connected to a computer prior to the execution the steps in setting up *WSR*.

The success of speech recognition depends on the quality of the microphone and the headset and desktop microphones are commonly used. Headset microphones are considered better since they are less prone to picking up extraneous sounds. A 30 minutes *WSR* training tutorial will assist users to exploit the commands used. **Figure 3** shows *WSR* pop-up widgets display for Starting Speech Recognition.

There is a unique voice profile used by *WSR* engine to recognize users' voice and spoken commands. Once



**Figure 2.** A speech recognition process.



**Figure 3.** A WSR pop-up widget.

WSR is utilized, users' voices profile gets more detailed and consequently improving computer's ability to understand the voices.

#### **The WSR implementation**

The implementation [17] written for C# provides an overview and examples in Windows Forms application by the following operations:

- 1) Initialize the speech recognizer.  
`Speech Recognizer SR = new Speech Recognizer().`
- 2) Create a WSR grammar.  
 The grammar is created by using the constructors and methods on the *Grammar Builder* and *Choices* classes. For a simple grammar, an element is added in a *Choices* object using the *Add* method. Then the *Grammar Builder* instance is created and using the *Append* method to insert the elements in *that* instance. Finally, the *Grammar Builder* instance is the initialized. A user must speak exactly one of the elements added by the *Choices* instance in order to attain the match between user speech and the grammar.
- 3) Load the grammar into the speech recognizer.  
 The grammar created in the previous operation must be loaded and passed into the speech recognizer by calling the *Load Grammar (Grammar)* method.
- 4) Register for speech recognition event notification.  
 The speech recognizer raises a number of the *Speech Recognized* events during its operation, when it accepts a user utterance with a grammar. A notification of this event is registered by appending an *Event Handler* instance via *SR\_Speech Recognized* the name written by a developer.
- 5) Create a handler for the speech recognition event.  
 A handler created for the *Speech Recognized* event displays the text of the recognized word or phrase using the *Result* property on the *Speech Recognized Event Args* parameter, e.

## **2.2. Primary School Mathematics**

Mathematics is the study of measurements, properties, and relationships of quantities and sets, using numbers and symbols. It is a group of related sciences, including algebra, geometry, and calculus. It concerns with the study of quantity, shape, and space and their interrelationships by using specialized notations. These notations called mathematical operations and the process is to look for solutions to problems or studies of some scientific field. In this research, Primary Mathematics contents taught in Elementary Schools, Saudi Arabia [18] are considered. **Figure 4** shows some materials taken from Class 1 book.

The basic topics in elementary mathematics are arithmetic and geometry. Elementary mathematics is used in everyday life activities such as dining, cooking, buying and selling things. It is also an essential first step on the path to understanding science. The contents presented are only covered basic digits 0 to 9 and addition operation for two single digits known as basic facts. By definition, the basic facts are any numbers or mathematical facts or ideas that can be instantly recalled without having to resort to a strategy to derive it. **Table 1** shows the details for addition operation basic facts for two single digits from 0 to 9.



Figure 4. Part of Mathematics for Grade 1 contents.

Table 1. The basic facts addition operation for digit 0 to 9.

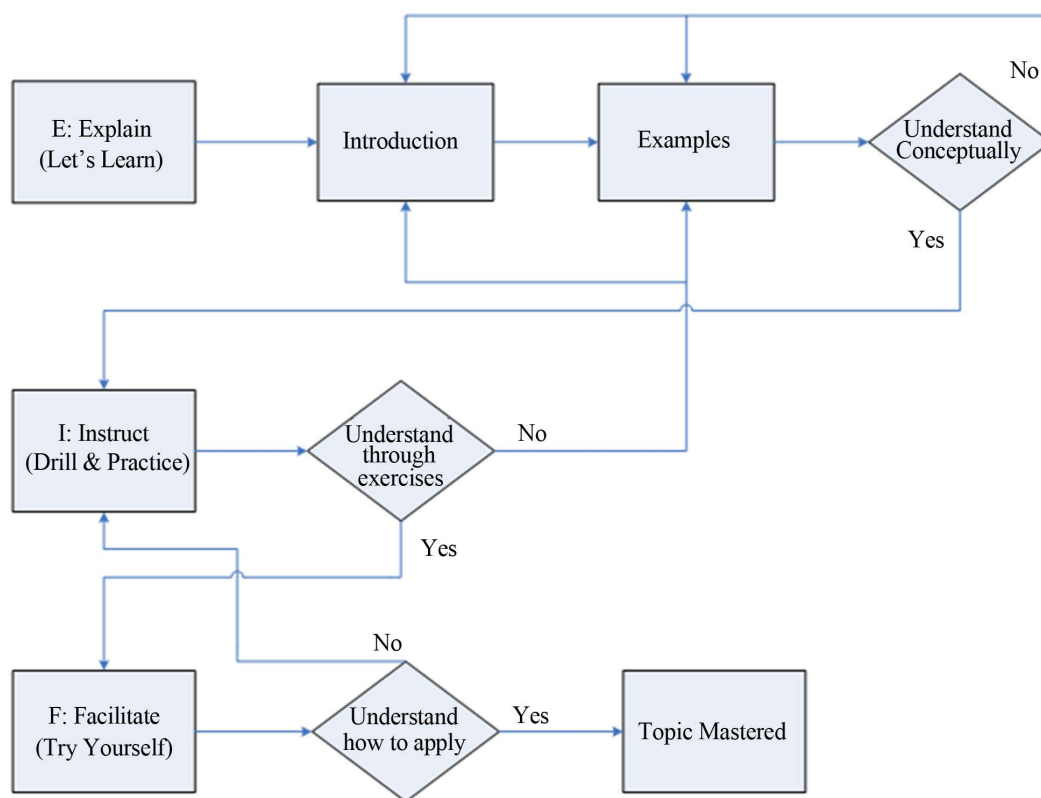
Addition operation for digit 0 to 9									
0 + 0 = 0	1 + 0 = 1	2 + 0 = 2	3 + 0 = 3	4 + 0 = 4	5 + 0 = 5	6 + 0 = 6	7 + 0 = 7	8 + 0 = 8	9 + 0 = 9
0 + 1 = 1	1 + 1 = 2	2 + 1 = 3	3 + 1 = 4	4 + 1 = 5	5 + 1 = 6	6 + 1 = 7	7 + 1 = 8	8 + 1 = 9	9 + 1 = 10
0 + 2 = 2	1 + 2 = 3	2 + 2 = 4	3 + 2 = 5	4 + 2 = 6	5 + 2 = 7	6 + 2 = 8	7 + 2 = 9	8 + 2 = 10	9 + 2 = 11
0 + 3 = 3	1 + 3 = 4	2 + 3 = 5	3 + 3 = 6	4 + 3 = 7	5 + 3 = 8	6 + 3 = 9	7 + 3 = 10	8 + 3 = 11	9 + 3 = 12
0 + 4 = 4	1 + 4 = 5	2 + 4 = 6	3 + 4 = 7	4 + 4 = 8	5 + 4 = 9	6 + 4 = 10	7 + 4 = 11	8 + 4 = 12	9 + 4 = 13
0 + 5 = 5	1 + 5 = 6	2 + 5 = 7	3 + 5 = 8	4 + 5 = 9	5 + 5 = 10	6 + 5 = 11	7 + 5 = 12	8 + 5 = 13	9 + 5 = 14
0 + 6 = 6	1 + 6 = 7	2 + 6 = 8	3 + 6 = 9	4 + 6 = 10	5 + 6 = 11	6 + 6 = 12	7 + 6 = 13	8 + 6 = 14	9 + 6 = 15
0 + 7 = 7	1 + 7 = 8	2 + 7 = 9	3 + 7 = 10	4 + 7 = 11	5 + 7 = 12	6 + 7 = 13	7 + 7 = 14	8 + 7 = 15	9 + 7 = 16
0 + 8 = 8	1 + 8 = 9	2 + 8 = 10	3 + 8 = 11	4 + 8 = 12	5 + 8 = 13	6 + 8 = 14	7 + 8 = 15	8 + 8 = 16	9 + 8 = 17
0 + 9 = 9	1 + 9 = 10	2 + 9 = 11	3 + 9 = 12	4 + 9 = 13	5 + 9 = 14	6 + 9 = 15	7 + 9 = 16	8 + 9 = 17	9 + 9 = 18

### 2.3. The Explain, Instruct and Facilitate (EIF) Techniques

Learning methods are frequently referred to as ways through which instructors deliver instructions and learners access these instructions. Several learning methods are described as traditional learning, e-Learning, blended learning, mobile learning, and personalized learning. These methods [19] accompanied with the advancements in technology and the paradigm shift from traditional learning to personalized learning methods. The proposed EIF technique, Explain (*E*), Instruct (*I*) and Facilitate (*F*) involves the development of theory and interactive courseware. Figure 5 shows that each stage has a different approach that involves a different role in *T* & *L*.

The Explain (*E*) stage is the process of explaining concepts for topics delivered. Many examples and other explanations are used to clarify in learning the concepts. Consequently, this stage can be viewed as presenting a text book for topics are being studied. For a courseware development, it is like an electronic book and the questions can be varied using a random function, especially involving numbers. It is useful for self-learning for those who haven't learned the concept presented. Instructors' presence might be required to clarify or clear any doubts.

The Instruct (*I*) stage is a dual communication where the session is benefited by answering the given questions. Instructors provide the questions according to levels, from easy to difficult. It is a test of knowledge and should be nurtured so that they will not hesitate to answer any questions. The courseware developed is seen as semi-interactive exercise book of students' involvement in response to questions displayed on a computer screen. The numbers are delivered randomly and answers provided by students will be verified by computer. For any



**Figure 5.** The *EIF* flow diagram.

difficulties the level Explain (*E*) can be referred again.

Learners' full participation is needed at the Facilitate (*F*) level. They are required to utter both questions and answers whilst instructors' tasks are to guide or act as facilitators. It is a full of interactive *T & L* session and a courseware developed at this stage requires knowledge for concepts of topics taught. In addition, the use of logic should be considered as computers assumed instructors jobs. At any higher level of difficulties, some tips for guiding learners should be provided. For the materials chosen in [Figure 4](#) and [Table 1](#), learners are required to pose questions and replied with answers which are verified by computers. [Table 2](#): shows a general *T & L* process to be used in the developed software using *EIF* Techniques.

A courseware development requires methods and techniques appropriate to *T & L* that can help to improve learners' understanding. The hardware and software specifications have to be determined as this has to be consistent with the requirements used for the development. Any cases relating to *T & L* should be studied to see the strategies are compatible with a computer-aided learning. In this research, the tools used are *WSR* and *MSVB2008* running on a standard personal computer.

## 2.4. The Algorithm

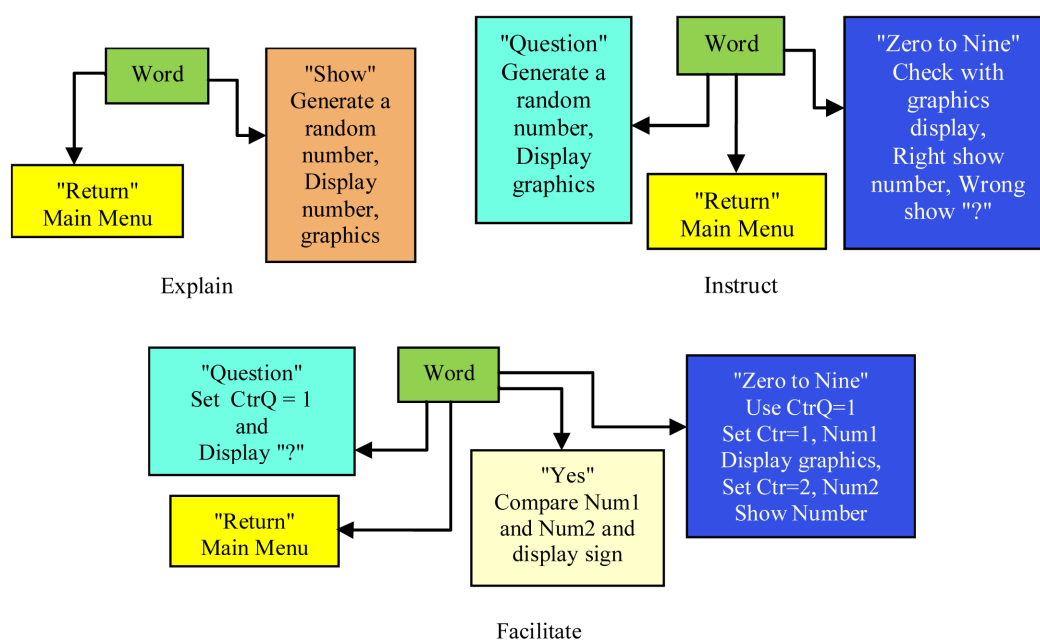
By using *WSR* and *MSVB2008*, the idea is to build a language grammar that allocates word phonemes for a user's voice to be recognized. The system is designed as such answers will only be accepted preceded by questions uttered. [Table 3](#) contains the suggested words list and these are used in the source codes in section 3.2 and the respective interface in Section 3.3. The flow diagrams in [Figure 6](#) and [Figure 7](#) used the words mentioned in [Table 3](#) and transferred into the source codes presented in Section 3.2.

## 3. Results and Discussions

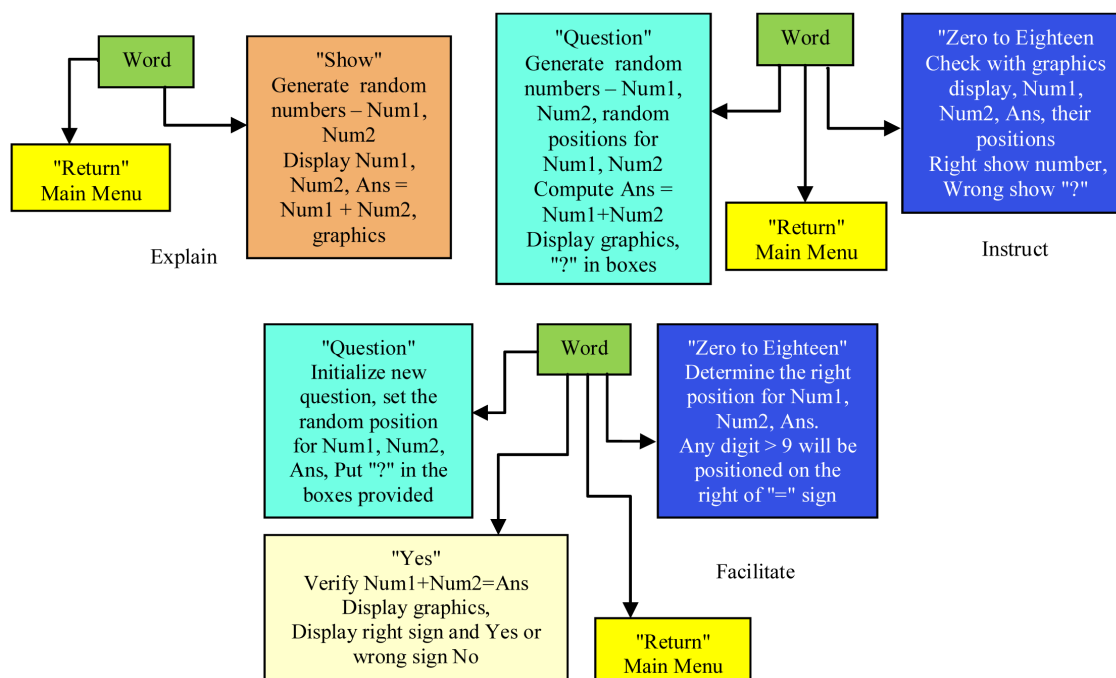
### 3.1. Creating *M-EIF* Speech Recognition Grammar

The idea is to build the language grammar that allocates word phonemes in [Table 3](#) for a user's voice to be rec-





**Figure 6.** The flow diagram for numbers.



**Figure 7.** The flow diagram for addition.

**Table 2.** *T & L* process.

<i>T &amp; L</i>		
Strategy	Approach	Technique
<ul style="list-style-type: none"> <li>Material Centered</li> <li>Teacher Centered</li> <li>Student Centered</li> </ul>	<ul style="list-style-type: none"> <li>From Simple to Complex</li> <li>From General to Specific</li> <li>Multiple Stage Understanding</li> </ul>	<ul style="list-style-type: none"> <li>Explain (<i>E</i>)</li> <li>Instruct (<i>I</i>)</li> <li>Facilitate (<i>F</i>)</li> </ul>

**Table 3.** The words list used in the source codes for each interface.

Interface		Words list
Main Menu		See, Say, Do, Learn, Check, Attempt, Numbers, Books, Flow, Exit
	Explain	Show, Return
Numbers	Instruct	Question, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Return
	Facilitate	Question, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Yes, Return
	Explain	Show, Return
Addition	Instruct	Question, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, Eleven, Twelve, Thirteen, Fourteen, Fifteen, Sixteen, Seventeen, Eighteen, Return
	Facilitate	Question, Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, Eleven, Twelve, Thirteen, Fourteen, Fifteen, Sixteen, Seventeen, Eighteen, Yes, Return
	EIF Flowchart	Return
	Numbers up to 100	Return
	Material used	Return

ognized. **Figure 8** shows the *M-EIF* architecture.

In this research, a speech recognition application in [17] [20] is adapted to perform the following operations and based on *MS-VB2008* structure. The following details are for the *M-EIF*:

- 1) The definition for *M\_ EIF* as a new Speech Recognition is defined as  
*Dim WithEvents M\_ EIF As New Recognition.SpeechRecognizer.*
- 2) The definitions for the grammar, alphabets, and loading them are presented as  
*Dim WordFacts As New Recognition.SrgsGrammar.SrgsDocument.*  
*Dim WordRule As New Recognition.SrgsGrammar.SrgsRule ("M\_ EIF").*  
*Dim WordList As New Recognition.SrgsGrammar.SrgsOneOf ("See", "Say", "Do", "Learn", "Check", "Attempt", "Numbers", "Books", "Flow", "Exit").*  
*WordRule.Add(WordList).*  
*WordFacts.Rules.Add (WordRule).*  
*WordFacts.Root = WordRule.*  
*M\_ EIF.LoadGrammar (New Recognition.Grammar (WordFacts)).*
- 3) The events, whether the defined *M\_ EIF* is recognized according to the grammar prescribed in Step 2, or otherwise and these require a subprogram defined as  
*Private Sub M\_ EIF \_SpeechRecognized (ByVal sender As Object, ByVal e As System. Speech. Recognition. Recognition EventArgs) Handles M\_ EIF.SpeechRecognized.*  
 which returns the value in *e* as *e.Result.Text*
- 4) The values in Step 2 are connected to the chosen location to display an interface, digit or picture for words designated in Step 3. For example, on uttering the word "See", the system executes *frmNumberShow.show*. The message "What was that?" in the widget is displayed when no matching is attained.

### 3.2. The Source Codes

The following source codes are based on the algorithm derived in Section 2.4. **Figure 9** and **Figure 10** show the source codes for *M-EIF* Main Menu and its Speech Recognizer (*SR*) procedure.

The source codes for other interfaces and their respective *SR* procedures are as follows. The words list used for each form is as given in **Table 3**.

**Number-Explain:** "See", **Figure 11** and **Figure 12**.

**Number-Instruct:** "Say", **Figure 13** and **Figure 14**.

**Number-Facilitate:** "Do", **Figure 15** and **Figure 16**.

**Addition-Explain:** "Learn", **Figure 17** and **Figure 18**.

**Addition-Instruct:** "Check", **Figure 19** and **Figure 20**.

**Addition-Facilitate:** "Attempt", **Figure 21** and **Figure 22**.



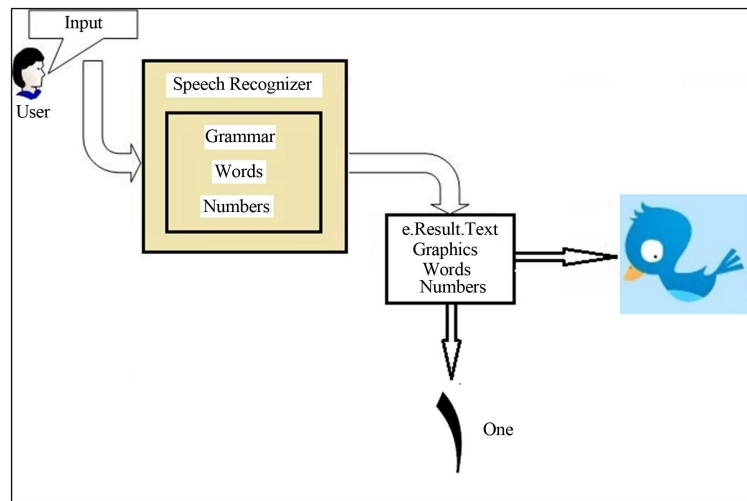


Figure 8. The M-EIF architecture.

```

Imports System.Speech

Public Class frmIntro

    Dim WithEvents M{EIF As New Recognition.SpeechRecognizer

    Private Sub frmIntro_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim WordFacts As New Recognition.SrgsGrammar.SrgsDocument
        Dim WordsRule As New Recognition.SrgsGrammar.SrgsRule("M{EIF")
        Dim WordsList As New Recognition.SrgsGrammar.SrgsOneOf("See", "Say", "Do", "Learn", "Check", "Flow", "Exit")

        WordsRule.Add(WordsList)
        WordFacts.Rules.Add(WordsRule)
        WordFacts.Root = WordsRule
        M{EIF.LoadGrammar(New Recognition.Grammar(WordFacts))
        lblWord.Visible = True
        lblWord.Text = "See Say Do Learn Check Attempt Numbers Books Flow Exit"
    End Sub
End Class

```

Figure 9. M-EIF main menu.

```

Private Sub M{EIF_SpeechRecognized(ByVal sender As Object, ByVal e As System.Speech.SpeechRecognitionEventArgs)
    Select Case e.Result.Text
        Case "See"
            frmNumberShow.Show()
        Case "Say"
            frmNumberTry.Show()
        Case "Do"
            frmNumberSay.Show()
        Case "Learn"
            frmAddito9.Show()
        Case "Check"
            frmInstrito9.Show()
        Case "Attempt"
            frmDolto9.Show()
        Case "Numbers"
            frmNumbers.Show()
        Case "Books"
            frmBooks.Show()
        Case "Flow"
            frmEIFChart.Show()
        Case "Exit"
            End
    End Select
    Me.Hide()
End Sub

```

Figure 10. M-EIF SR procedure.

```
Imports System.Speech

Public Class frmNumberShow

    Dim WithEvents ShowNumber As New Recognition.SpeechRecognizer
    Dim Num1 As Integer

    Private Sub frmNumberShow_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("ShowNumber")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Show", "Return")
        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        ShowNumber.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        clearbirds()
        lblWord.Text = "Show Return"

    End Sub

End Class
```

Figure 11. Number-explain main.

```
ShowNumber SpeechRecognized

Private Sub ShowNumber_SpeechRecognized(ByVal sender As Object, ByVal e As System.Speech.SpeechRecognitionEventArgs) Handles ShowNumber.SpeechRecognized

    Label2.Visible = True
    Label2.Text = e.Result.Text
    clearbirds()

    Select Case e.Result.Text
        Case "Return"
            frmIntro.Show()
            Me.Hide()
            clearbirds()
        Case "Show"
            Num1 = Rnd() * 10
            Showbirds(Num1)
            If Num1 = 0 Then
                Label2.Text = "Zero"
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNum1\Zero.jpg")
            ElseIf Num1 = 1 Then
                Label2.Text = "One"
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNum1\One.jpg")
            ElseIf Num1 = 2 Then
                Label2.Text = "Two"
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNum1\Two.jpg")
            Else
                Label2.Text = "Error"
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNum1\Error.jpg")
            End If
        Case Else
            Label2.Text = "Error"
            pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNum1\Error.jpg")
        End Select

End Sub
```

Figure 12. Number-explain SR.

```
Imports System.Speech

Public Class frmNumberTry

    Dim WithEvents TryNumber As New Recognition.SpeechRecognizer
    Dim Num1 As Integer

    Private Sub frmNumberTry_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("TryNumber")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Question", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Return")
        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        TryNumber.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        lblWord.Text = "Question Zero One Two Three Four Five Six Seven Eight Nine Return"
        clearbirds()

    End Sub

End Class
```

Figure 13. Number-instruct main.

```

TryNumber
SpeechRecognized

Private Sub TryNumber_SpeechRecognized(ByVal sender As Object, ByVal e As System.

    Label2.Text = e.Result.Text

    Select Case e.Result.Text
        Case "Return"
            frmIntro.Show()
            Me.Hide()
            clearbirds()
        Case "Question"
            clearbirds()
            Num1 = Rnd() * 10
            If Num1 > 0 Then
                Showbirds(Num1)
            End If
            pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\SayNumb
        Case "Zero"
            If Num1 = 0 Then
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\Say
            Else
                pbxArabic.BackgroundImage = System.Drawing.Image.FromFile("D:\Say
            End If
        Case "One"

```

Figure 14. Number-instruct SR.

```

Imports System.Speech

Public Class frmNumberSay

    Dim WithEvents SayNumber As New Recognition.SpeechRecognizer
    Dim Num1, Num2, Ctr, CtrQ As Integer

    Private Sub frmNumberSay_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) !

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("SayNumber")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Question", "Zero", "One", "T
            "Seven", "Eight", "Nine", "Ye:

        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        SayNumber.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        clearbirds()
        lblWord.Visible = True
        lblWord.Text = "Question Zero One Two Three Four Five Six Seven Eight Nine Yes Return"
        Ctr = 1
        CtrQ = 0

```

Figure 15. Number-facilitate main.

```

(frmNumberSay Events)
Load

Private Sub SayNumber_SpeechRecognized(ByVal sender As Object, ByVal e

    Label2.Text = e.Result.Text

    Select Case e.Result.Text
        Case "Question"
            CtrQ = 1
            clearbirds()
            Label2.Visible = True
            pbxArabic.Visible = True
            pbxArabic.BackgroundImage = System.Drawing.Image.FromFile
        Case "Yes"
            If Num1 = Num2 Then
                pbxCheck.BackgroundImage = System.Drawing.Image.FromFi
            Else
                Label2.Text = "No"
                pbxCheck.BackgroundImage = System.Drawing.Image.FromFi
            End If
            pbxCheck.Visible = True
            CtrQ = 0
        Case "Zero"
            If (CtrQ = 1) And (Ctr = 1) Then
                Num1 = 0

```

Figure 16. Number-facilitate SR.

```
Imports System.Speech

Public Class frmAddItto9

    Dim WithEvents ShowAdd As New Recognition.SpeechRecognizer
    Dim Num1, Num2, Ans As Integer

    Private Sub frmAddItto9_Load(ByVal sender As System.Object, ByVal e As System.E

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("ShowAdd")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Show", "Return")

        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        ShowAdd.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        lblWord.Text = "Show Return"
        clearallbirds()
        clearNumbers()

    End Sub
```

Figure 17. Addition-explain main.

ShowAdd ▼ ⚡ SpeechRecognized

```
Private Sub ShowAdd_SpeechRecognized(ByVal sender As Object, ByVal e As System.!)
    clearallbirds()

    Select Case e.Result.Text
        Case "Return"
            frmIntro.Show()
            Me.Hide()
            clearallbirds()
        Case "Show"
            Num1 = Rnd() * 10
            If Num1 > 0 Then
                Displaybird1(Num1)
            End If
            If Num1 = 0 Then
                lblNum1.Text = "Zero"
                pbxNum1.Visible = True
                pbxNum1.BackgroundImage = System.Drawing.Image.FromFile("D:\Say\
            ElseIf Num1 = 1 Then
                lblNum1.Text = "One"
                pbxNum1.BackgroundImage = System.Drawing.Image.FromFile("D:\Say\
            ElseIf Num1 = 2 Then
                lblNum1.Text = "Two"
```

Figure 18. Addition-explain SR.

```
Imports System.Speech

Public Class frmInstrItto9

    Dim WithEvents TryAdd As New Recognition.SpeechRecognizer
    Dim Num1, Num2, Ans1, Ans2, Ans3, Ans, QType As Integer

    Private Sub frmInstrItto9_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("TryAdd")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Question", "Zero", "One", "Two", "Three", "Four", "Five", "Six", _
            "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", _
            "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Return")

        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        TryAdd.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        lblWord.Text = "Question Zero One Two Three Four Five Six Seven Eight Nine Ten Eleven Twelve Thirteen Fourteen Fifteen Sixteen
        clearallbirds()
        clearNumbers()

    End Sub
```

Figure 19. Addition-instruct main.

```

TryAdd

Private Sub TryAdd_SpeechRecognized(ByVal sender As Object, ByVal e As System.Speech.SpeechEventArgs) Handles MyBase.SpeechRecognized
    Select Case e.Result.Text
        Case "Question"
            clearallbirds()
            Ans1 = 0
            Ans2 = 0
            Ans3 = 0
            Num1 = Rnd() * 10
            Num2 = Rnd() * 10
            Ans = Num1 + Num2
            QType = Rnd() * 3 + 1
            If QType = 1 Then
                If Num1 > 0 Then
                    Displaybird1(Num1)
                End If
                If Num2 > 0 Then
                    Displaybird2(Num2)
                End If
            ElseIf QType = 2 Then
                If Num2 > 0 Then
                    Displaybird2(Num2)
                End If
            End If
    End Select
End Sub

```

Figure 20. Addition-instruct SR.

```

Imports System.Speech

Public Class frmDoItto9

    Dim WithEvents DoAdd As New Recognition.SpeechRecognizer
    Dim Num1, Num2, Ans1, Ans2, Ans3, Ans As Integer
    Dim QPlace, QPlace1, QPlace2, QPlace3 As Integer

    Private Sub frmDoItto9_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        Dim NumberGrammar As New Recognition.SrgsGrammar.SrgsDocument
        Dim NumbersRule As New Recognition.SrgsGrammar.SrgsRule("DoAdd")
        Dim NumbersList As New Recognition.SrgsGrammar.SrgsOneOf("Question", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Yes", "Return")

        NumbersRule.Add(NumbersList)
        NumberGrammar.Rules.Add(NumbersRule)
        NumberGrammar.Root = NumbersRule
        DoAdd.LoadGrammar(New Recognition.Grammar(NumberGrammar))
        Randomize()
        clearallbirds()
        PutQuestions()
        lblWord.Visible = True
    End Sub
End Class

```

Figure 21. Addition-facilitate main.

```

DoAdd

Private Sub DoAdd_SpeechRecognized(ByVal sender As Object, ByVal e As System.Speech.SpeechEventArgs) Handles MyBase.SpeechRecognized
    Select Case e.Result.Text
        Case "Question"
            clearallbirds()
            QReset()
            PutQuestions()
            pbxCheck.Visible = False
        Case "Yes"
            If ((Ans1 = 1) And (Ans2 = 1) And (Ans3 = 1)) Then
                Displaybird1(Num1)
                Displaybird2(Num2)
                Displaybird3(Ans)
                If (Num1 + Num2) = Ans Then
                    pbxCheck.BackgroundImage = System.Drawing.Image.FromFile("D:\...")
                Else
                    pbxCheck.BackgroundImage = System.Drawing.Image.FromFile("D:\...")
                End If
                pbxCheck.Visible = True
            End If
        Case "Zero"
            If (QPlace1 = 1) And (Ans1 = 0) Then
                pbxNum1.BackgroundImage = System.Drawing.Image.FromFile("D:\SayN...")
            End If
    End Select
End Sub

```

Figure 22. Addition-facilitate SR.



### 3.3. The *M-EIF* Testing

The aim of this research is to develop *T & L* prototype in basic Mathematics, *M-EIF* based on techniques called *EIF*, Explain (*E*), Instruct (*I*) and Facilitate (*F*) using *WSR* and *MSVB2008*. The contents used are part of Elementary School Mathematics Curriculum, Saudi Arabia. **Figure 23** shows the First Form upon executing *M-EIF*. The prototype only accepts users' voice, hence *WSR* are required to be installed prior to the testing. The other forms with the details are as follows.

#### 1) *Number: Explain*—"See"

**Figure 24** is shown upon uttering the word "See". By uttering the word "Show", the Number-Explain form randomly displays a digit from 0 to 9, the corresponding graphics birds and word depending on the digit that appeared. **Figure 25** and **Figure 26** show some sample output "Return" forces the lessons back to *M-EIF* Main Menu, **Figure 23**.

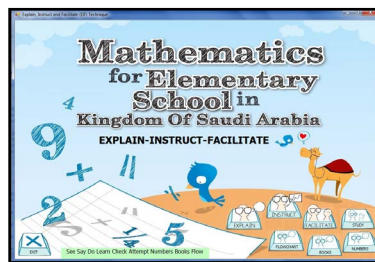


Figure 23. *M-EIF* first form.

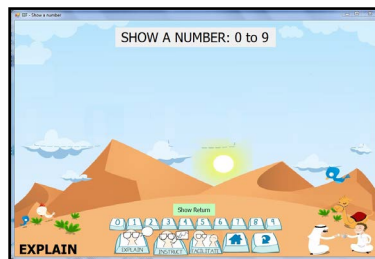


Figure 24. Number-explain main.



Figure 25. Number-explain "Zero".



Figure 26. Number-explain "Nine".



## 2) *Number-Instruct*—“Say”

Upon uttering the word “Say”, the Number-Instruct form as in **Figure 27** appears. The word “Question” must be uttered next for a computer to show randomly a digit ranging from 0 to 9. **Figure 28** shows a sample output. And for a right answered, the display is as **Figure 29**, otherwise the “?” is unchanged.

## 3) *Number: Facilitate*—“Do”

The screen displays the interface as in **Figure 30**, upon uttering the word “Do”. Users have to utter “Question” and a number (0 - 9), then the graphics bird is displayed. For the answer, they have to count the number of birds and say the number again, then “Yes” to validate. **Figures 31-34** shows the details for the right answer where Q and A stand for Question and Answer respectively. **Figures 35-37** is the display the wrong answer.



Figure 27. Number-instruct main.



Figure 28. Number-instruct “Question”.



Figure 29. Number-instruct “Seven”.



Figure 30. Number-facilitate main.



Figure 31. Number-facilitate—“Question”.



Figure 32. Number-facilitate (Q)—“Seven”.



Figure 33. Number-facilitate (A)—“Seven”.



Figure 34. Number-facilitate—“Yes”.



Figure 35. Number-facilitate (Q)—“Six”.



Figure 36. Number-facilitate (A)—“Five”.



Figure 37. Number-facilitate (A)—“Yes”.

#### 4) Addition: Explain—“Learn”

The word “Learn” will show the main form, **Figures 38-41** shows the random outputs by uttering the word “Show”.

#### 5) Addition: Instruct—“Check”

By uttering the word “Check”, the display as in **Figure 42** will be shown. The word “Question” will randomly show the position of bird graphics between the three different locations, *i.e.*, **Figures 43-48** are a right combination for answers to a random question in **Figure 43**.

#### 6) Addition: Facilitate—“Attempt”

**Figure 49** shows the Main form for Addition-Facilitate upon uttering the word “Attempt”. From **Figure 49**, users are required to utter the first number and the computer will choose the location. For any number bigger than 9, the number will be inserted on the right hand side of the equation. Then, the rest has to be completed in turn with the final word is “Yes” to validate the answer. **Figures 50-53** shows the details for correct answer whilst **Figure 54** and **Figure 55** are for the wrong and correct answer respectively.

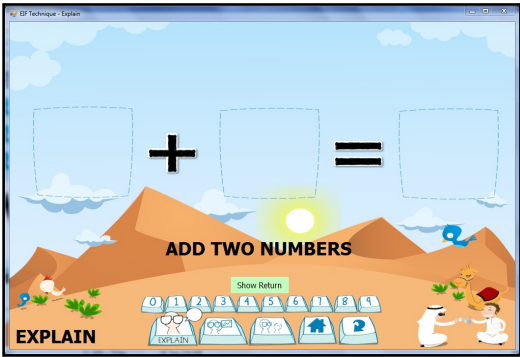


Figure 38. Addition-explain main.

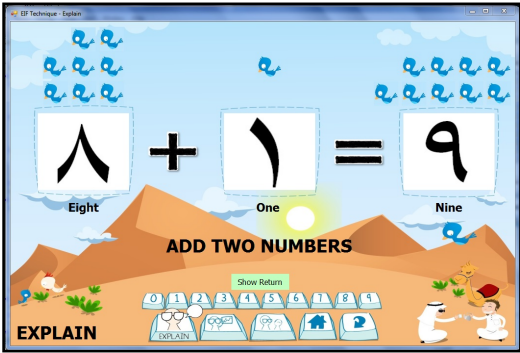


Figure 39. Addition-explain "Show".

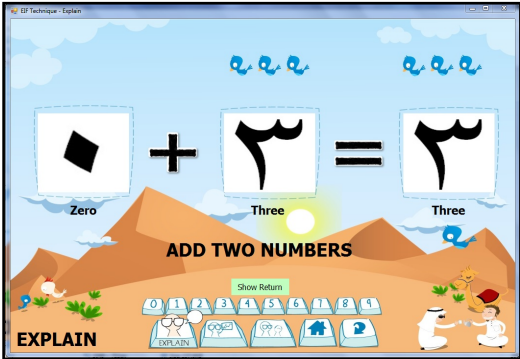


Figure 40. Addition-explain "Show".

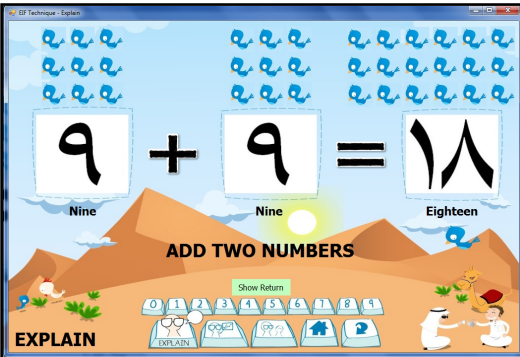


Figure 41. Addition-explain "Show".

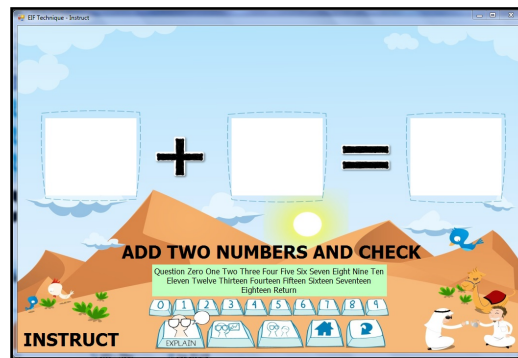


Figure 42. Addition-instruct main.

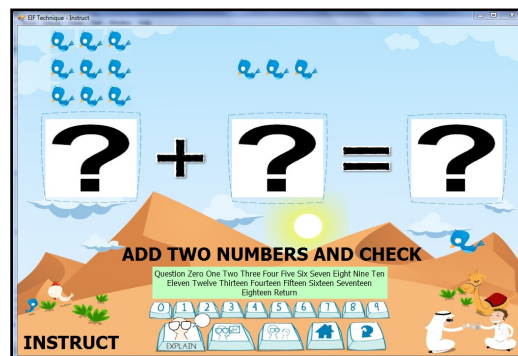


Figure 43. Addition-instruct "Question"

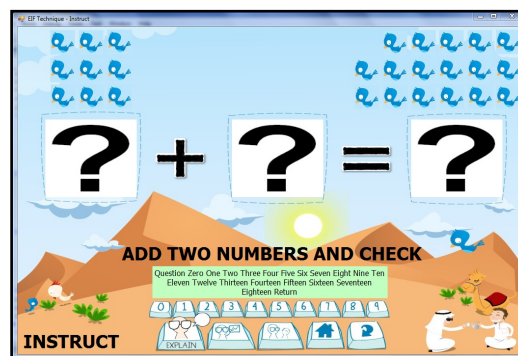


Figure 44. Addition-instruct "Question".

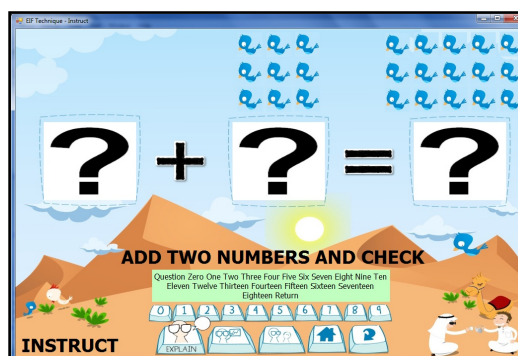


Figure 45. Addition-instruct "Question".



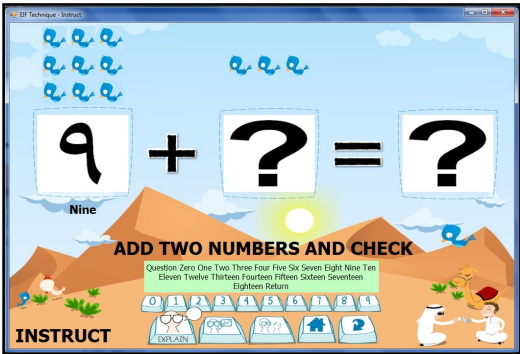


Figure 46. Addition-instruct “Nine”.

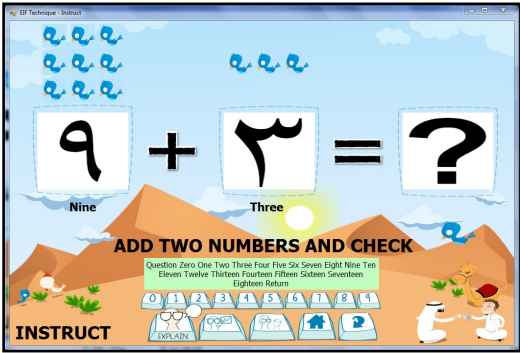


Figure 47. Addition-instruct “Three”.

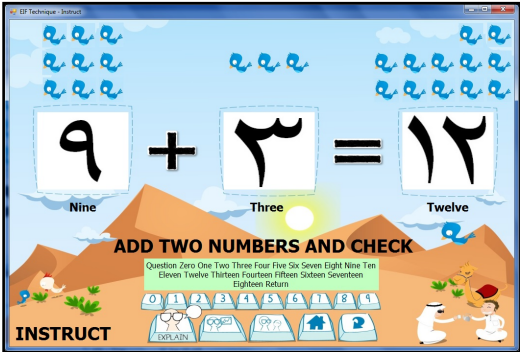


Figure 48. Addition-instruct “Twelve”.

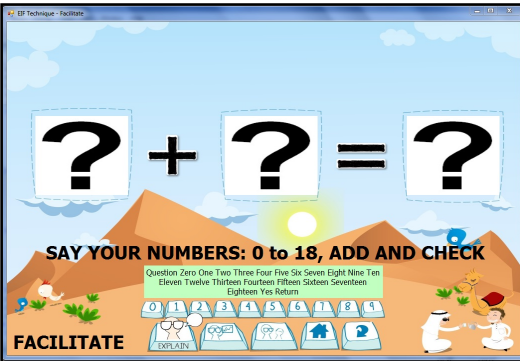


Figure 49. Addition-facilitate main.



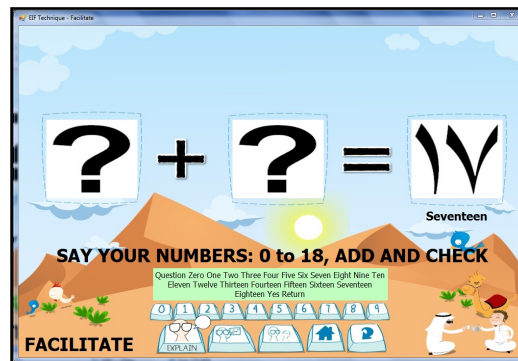


Figure 50. Addition-facilitate "Seventeen".

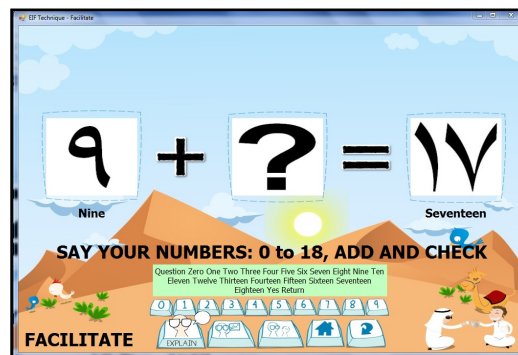


Figure 51. Addition-facilitate "Nine".

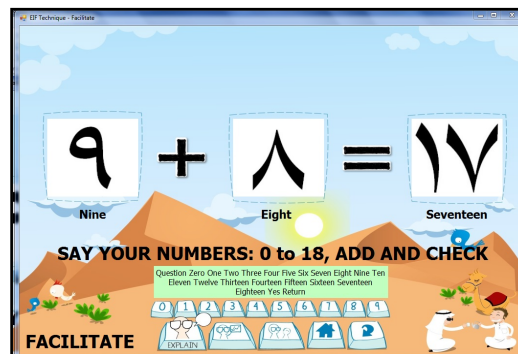


Figure 52. Addition-facilitate "Eight".

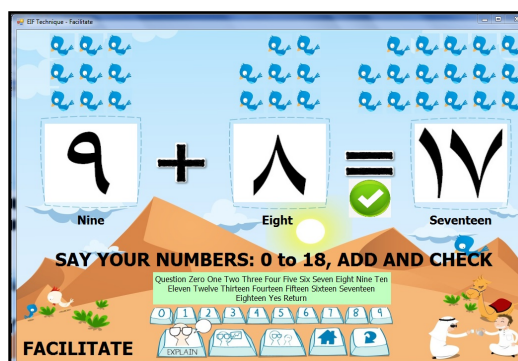


Figure 53. Addition-facilitate "Yes".

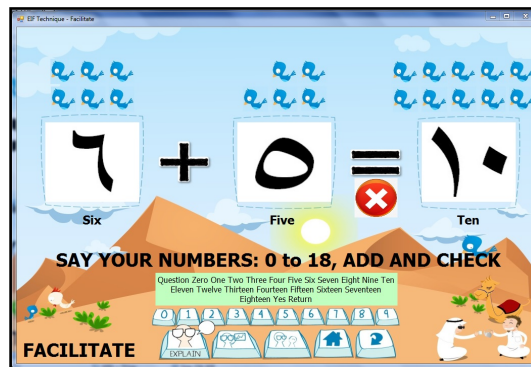


Figure 54. Addition-facilitate "Yes".



Figure 55. Addition-facilitate "Yes".

#### 4. Conclusions and Recommendations

The rapid change in mathematics education in as many decades implies a wind of change in society. Starting from behavioral theory the 1950-1970, Cognitive Science (1970-1980) and from 1980 to the present theory, the constructivism has been on the trust. Besides, the e-learning system becomes an alternative educational tool that promises to play a prominent role in years ahead. An interactive approach in *T&L Mathematics* using Explain, Instruct and Facilitate (*EIF*) technique is expected to open a new wave in mathematical education. The prototype used the multimedia technology to support the viability and significant of the approach. The *EIF* technique could be deployed in a developed multimedia courseware and implemented at schools for *T & L*. Learners as beginners on Mathematics or otherwise will benefit from the courseware whilst instructors act an aide to facilitate them.

One of the criteria that are essential in developing multimedia courseware is attractive interfaces and graphic designs. These features should be compatible with the local environment and user friendly. The consistency between learning objectives and content of instruction are required for conveying the right pedagogy with the chosen materials. In this research, *M-EIF* is successfully developed with the contents consist of single digits (0 to 9) and the basic facts for addition operation with the graphics specially designed for the local environment. The concept of interactive techniques has been much emphasized via *EIF*. *M-EIF* could be used to assist learners in mathematics and also English. Regardless of their competencies, *M-EIF* provides random examples and questions in enriching *T & L*.

*M-EIF* has been developed for some topics in Primary Mathematics with simple graphics representation and voices using *WSR*. To develop better courseware, the expertise in Mathematics or chosen subjects, educational psychology, programming, graphics and animations are much required and work as a group. By this, then the rest of the contents either simple or abstract such as mathematical proofs could be done subsequently for the entire curriculum and followed by some tests in schools to acquire the feedbacks. The *EIF* techniques could also be applied in the higher levels of education, at secondary schools and university to judge its effectiveness. In addition, this technique could also be implemented and tested in other courses. With the availability of financial

resources and manpower, further improvement on *M-EIF* is certainly viable. The use of web technology might be considered in facilitating *T & L* to anyone, in everywhere and anytime. *M-EIF* might be regarded as the opening for the development of more sophisticated Multimedia-*WSR* assisted courseware in the future.

## Acknowledgements

This Project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. 27-004/430. The authors, therefore, acknowledge with thanks DSR technical and financial support.

## References

- [1] Glaser, R. (1976) Components of a Psychology of Instruction: Toward a Science of Design. *Review of Educational Research*, **46**, 1-24. <http://www.jstor.org/stable/1169916>  
<http://dx.doi.org/10.3102/00346543046001001>
- [2] Zhang, M., Lundeborg, M., McConnell, T.J., Koehler, M.J. and Eberhardt, J. (2010) Using Questioning to Facilitate Discussion of Science Teaching Problems in Teacher Professional Development, *Interdisciplinary Journal of Problem-Based Learning*, **4**, 57-82. <http://dx.doi.org/10.7771/1541-5015.1097>
- [3] Duran, E., Duran, L., Haney, J. and Scheuermann, A. (2011) A Learning Cycle for All Students: Modifying the 5E Instructional Model to Address the Needs of all Learners. *The Science Teacher*, **78**, 56-60. [www.scilinks.org](http://www.scilinks.org)
- [4] Frels, R.K., Sharma, B., Onwuegbuzie, A.J., Leech, N.L. and Stark, M.D. (2011) The Use of a Checklist and Qualitative Notebooks for an Interactive Process of Teaching and Learning Qualitative Research. *The Journal of Effective Teaching*, **11**, 62-79. [http://uncw.edu/cte/et/articles/Vol11\\_1/index.htm](http://uncw.edu/cte/et/articles/Vol11_1/index.htm)
- [5] Al Ghamdi, Y.A.S. (1987) The Effectiveness of Using Microcomputers in Learning Algebraic Precedence Conventions. Doctoral Dissertation, Florida State University, Florida.
- [6] Alessi, S.M. and Trollip, S.R. (2000) Multimedia for learning: Methods and Development. 3rd Edition, Allyn and Bacon, Boston.
- [7] Coelho, A.C.C. (2010) Interactive Textbooks and Student's Learning, *e-TEALS: An E-Journal of Teacher Education and Applied Language Studies*, **1**, 13-43.
- [8] Funkhouser, C. (1993) The Influence of Problem Solving Software on Student Attitudes about Mathematics. *Journal of Research on Computing in Education*, **25**, 339-346.
- [9] Hermans, R., Tondeur, J., van Braak, J. and Valcke, M. (2008) The Impact of Primary School Teachers' Educational Beliefs on the Classroom Use of Computers. *Computers & Education*, **51**, 1499-1509. <http://dx.doi.org/10.1016/j.compedu.2008.02.001>
- [10] van Braak, J., Tondeur, J. and Valcke, M. (2004) Explaining Different Types of Computer Use among Primary School Teachers. *European Journal of Psychology of Education*, **19**, 407-422. <http://dx.doi.org/10.1007/BF03173218>
- [11] Harun, J. and Tasir, Z. (2000) Pengenalan Kepada Multimedia. Venton Publishing, Kuala Lumpur.
- [12] Rashid, A.R.A. (2000) Wawasan Dan Agenda Pendidikan. Utusan Publications and Distributors Sdn. Bhd., Kuala Lumpur.
- [13] Dolk, M., den Hertog, J. and Gravemeijer, K. (2002) Using Multimedia Cases for Educating the Primary School Mathematics Teacher Educator: A Design Study. *International Journal of Educational Research*, **37**, 161-178, [http://dx.doi.org/10.1016/S0883-0355\(02\)00058-7](http://dx.doi.org/10.1016/S0883-0355(02)00058-7)
- [14] Halawani, S.M., Daman, D., Kari, S. and Ahmad, A.R. (2013) An Avatar Based Translation System from Arabic Speech to Arabic Sign Language for Deaf People. *International Journal of Computer Science & Network Security*, **13**, 43-52. [http://paper.ijcsns.org/07\\_book/201312/20131207.pdf](http://paper.ijcsns.org/07_book/201312/20131207.pdf)
- [15] Wigmore, A., Hunter, G., Pflügel, E., Denholm-Price, J. and Binelli, V. (2009) Using Automatic Speech Recognition to Dictate Mathematical Expressions: The Development of the "TalkMaths" Application at Kingston University. *Journal of Computers in Mathematics and Science Teaching*, **28**, 177-189. <http://www.editlib.org/p/30301>
- [16] Windows Speech Recognition Engine (2014) Microsoft Windows 7 Professional Operating Systems.
- [17] Windows Speech Recognition (2014) <http://msdn.microsoft.com/en-us/library/>
- [18] Mathematics for Grade 1 (2011) First Semester. Ministry of Education, KSA.
- [19] Sohlberg, M.M., Ehlhardt, L. and Kennedy, M. (2005) Instructional Techniques in Cognitive Rehabilitation: A Preliminary Report. *Seminars in Speech and Language*, Theime Medical Publishers Inc., **26**, 268-279. <http://dx.doi.org/10.1055/s-2005-922105>
- [20] Speech Recognition in Visual Studio 2008 (2014) [http://social.msdn.microsoft.com/Profile/bruno\\_yu/activity](http://social.msdn.microsoft.com/Profile/bruno_yu/activity)