

*English as a Second Language; Engaging the Early Years through Interactive Touch
Application Software*

Catherine A. Todd*¹, Keerthi Prasad*¹, Abeer Rahman*¹, Mohamed Redwan*¹,
Sarah Bennett*²

*¹University of Wollongong in Dubai, UAE, *² Deira International School, UAE

0229

The Asian Conference on Language Learning 2013

Official Conference Proceedings 2013

Abstract

An interactive touch application has been developed that targets early years of primary school education (kindergarten students) for improvement in English literacy and linguistic skills. The application services remote learning in English as a second language and is developed in accordance with school curricula and learning objectives, in collaboration with Deira International School, Dubai. In the UAE, GCC and other cultural contexts that require English literacy for wider community inclusion and where parental assistance may be lacking, the software provides an invaluable aid both inside and outside the classroom, for development in reading, writing and audition. Interactive activities that offer skill development and assessment are modularized, providing tasks in listening, reading and writing with phonetics and spelling challenges, combined with audio-visual prompts. Tasks and levels are assigned automatically as determined by the software, based on real-time competency assessment of individual students; tailoring the task and its difficulty to each child's needs. Fuzzy logic is applied for task/level determination. Handwriting character recognition is embedded for skill assessment in the writing module. Parents and teachers are able to view graphical results of a child's progress, as system administrators. The software is developed in C++ on Windows 8, utilizing Microsoft Visual Studio Express 2012 SDK. Compared with existing solutions, the software offers many new features and benefits. It offers a holistic approach with an expanse of syllabi-based tasks and automatically assesses each student individually, providing a tailored, engaging yet comprehensive package, toward universal education and community engagement.

Key-Words: - Innovative Language Teaching, E-Learning Methodologies, Educational Technologies, Touch Screen, Software Development, Computers for Education

iafor
The International Academic Forum
www.iafor.org

1 Introduction

This software provides an immersive application for young school children learning English as a Second Language (ESL); to provide them an opportunity to learn and develop their literacy and linguistic skills, where parental and/or instructional support may be lacking, or to target their development in a specific skill deficit. Multicultural cities exist globally, to a large extent driven by technology, and Dubai is a prime example, where children from different nationalities coexist. In this context, English provides a vital medium for communication, and hence inclusion and integration into society. Without basic skills in English, a child is at a disadvantage. Thus the objective of this software is to aid children from all nationalities to gain a good, basic understanding of English from the very beginning of their education; the vital years of learning and development.

The software focuses on three core elements via the modules: reading (phonetics), listening and writing. Within these modules, students progress through a series of levels to learn, develop and test their competencies. The activities within these modules are presented via a fun, interactive application on a touch-screen platform with a user-friendly Graphical User Interface (GUI), to further engage the students for self-learning. Implementation is done on Windows 8 operating system with LG touch-screen technology, with modules developed using Microsoft SDK and programming in C#, with database querying in SQL to Microsoft Access. The software offers instructional support, wherein teachers and parents can provide input and feedback, ascertaining a child's individual progress within each skill-set. Through fuzzy-logic algorithms, the software automatically assesses the competencies and progress of a student and places the student within their own activity level. HCR is implemented in the writing module, to automatically detect the child's response. Records can be automatically generated for instructors. It is the first software with the three modules to be based on current teaching curricula and tested within a cross-cultural, varied skill level classroom environment. The over-arching purpose of this software is not solely for assessment, but for a fun, portable learning environment accessible to children from across the world for their education and social inclusion. It is applicable to children from any educational institution, or those without access, to gain the basic skills in ESL.

2 Existing Solutions

Many applications are available in the current online market to assist in the acquisition of skills in English at an early age; these include Kids ABC Phonics (Intellijoy, 2013), Kids Learn to Read (Intellijoy, 2012) and Licking Letters (Tenlin Studios LLC, 2013). All three software applications offer letter and sound blending for word construction. Kids ABC Phonics aims to teach children the letter sounds and the basics of blending of these letters. The application is divided into four sections where the child is guided through the learning processes of letter sounds efficiently. Namely; learning letter sounds, building letter blocks, popping letter bubbles as a game functionality and making words, where the child is required to demonstrate the basics of phonics blending learned from the previous sections. This application assists the child in listening and learning the sounds of the alphabets. Writing is not included. Kids Learn to Read is an interactive game-based application for the Android OS, which invites preschool children to practice the blending of sounds together to read and spell simple words, such as "dog" and "sun". This application uses two main methods; with the

help of an interactive character named Tommy the Turtle, the child is guided to learn the blending of letter sounds into words. Here the child constructs words, determines how each letter is sounded out, how a word is formed and pronounced. In the second method, the child is encouraged to make words by turning real-time blocks with letters on each side until the word is correctly spelled. This application encourages a child to progress through each level, with positive reinforcement. Licking Letters is developed for Apple products running on iOS3.1.3 or later. It aids the child to learn how to spell words through an interactive user interface and with the help of a virtual character named Hoppy the Frog. Words to spell increase in the level of difficulty, with a total of 150 words to spell, starting with 3-lettered words and gradually increasing the difficulty to 5-lettered words, with an option to spell words with letters in or out of order. With each correctly spelled word, the child is rewarded with coins that can be redeemed to collect outfits to accessorize the virtual character, for motivation. Each time three words are spelled, the child is rewarded with a mini-game within the level to gain extra coins. Game difficulty can be set manually depending on the child's skill level. This program aims to help the child to spell simple words as well as learn new words.

A benefit of the developed software is that the application is specifically tailored to the UK curriculum followed by Deira International School, Dubai. The advantage that this software has over existing programs is that it is open source, curriculum-specific, and offers an advanced grading system through semi-automatic classification of difficulty levels and embedded activities. HCR is implemented in the Writing Module, with fuzzy logic for automatic grading and activity offerings within the Reading and Listening Modules.

3 Applied Methodologies, Algorithm Implementation and Output

The following is the Main Page GUI, with the three modules available for selection; Listening, Reading and Writing (Figure 3.1). Upon button press, the respective module is loaded.



Figure 3.1 A screenshot of the Main Page containing the modules

3.1 Listening Module Design and Development

Algorithm development and testing progressed from the conversion of a hard-limit algorithm, based on count taken of correct responses, and the time taken for each activity, to classify a child's progress, to a dynamic algorithm based on fuzzy logic. Program functionality is discussed below, for each function provided within the Listening Module (Figure 3.2).



Figure 3.2 Listening module GUI upon pressing the start button; also showing user performance (View Progress is selected).

Upon the loading of the listening module form, a connection is established to the database “studentDb” and its table, Listening1. Once the connection is established, a data table, dataresult, is created to hold the required data such as the word, time taken for the activity, the number of incorrect attempts, the activity set number and the difficulty level the selected word was from; for the recording purpose as the user proceeds through the application. Columns are created to the data table to store the values for each activity with their data type: Word, Time Taken, Number of Incorrect Attempts, Activity Set Number and Difficulty Level. This table is set as hidden by default and can be viewed as the child progresses by checking the check box “View Progress”.

The Start button is used to initialize the listening module. Upon clicking the start button, the speaker button, the print button, the ignore button are enabled while disabling the start button. The first activity set is loaded into the application and is displayed on the screen. This label can be hidden by the admin by checking the check box Hide label. The Speaker button is used to prompt the audio of the word displayed. Upon clicking the speaker button, the counter variable buttoncount is incremented by one upon each click; to keep track of how many times the speaker button has been clicked. The timer is initiated is set to increment every one second. An IF statement is introduced to check whether the button is clicked more than once, hence avoiding the reset of the timer. If the statement is validated, the current time is captured from the system clock and is saved to the variable “startTime” as well as the Yes Button (buttonyes) and No button (buttonno) is enabled respectively. A new sound player, simpleSound, is created. Based upon the check, simpleSound locates the sound file of the respective name and then plays the audio file.

The No button is used mainly to record the number of incorrect attempts by the user per activity as well as per activity set. Every time the user has an incorrect attempt at the activity, both the number of the incorrect activity (`incorr_num`) and the activity set (`total_incorr_num`) are incremented. The number of times the no button has been pressed by the admin is recorded and shown on the screen using the label “counterno”. For every tick by the Activity Timer (`activitytime`), the elapsed time is calculated by subtracting the time from the system clock at the given point of time, extracting the seconds from the elapsed time and saving it into the variable “timecount”. An additional one second is added to avoid the lag of one second which was noticed during application testing. The “timecount” variable is later used for Yes button functionality.

In the click event of the Yes button, the fuzzy logic algorithm is initiated. Fuzzy logic is essentially the classification of dynamically changing data. The fuzzy logic algorithm was developed based on existing work (Kim et al., 2001; Kao and Shih, 2008; Hsieh et al., 2012; Yusof et al., 2012). Key indicators of the performance for membership functions include time taken for an activity and the number of attempts at the activity. The fuzzy logic is executed depending on user input such as time taken by the user per activity and number of incorrect attempts per activity based on five main conditions executed based on his/her performance basis. Upon yes button press, the timer is paused to record time taken by the child for the activity, and a new row (dr) is created in the data table `dataresult` to store the values such as the word selected, the time taken by the user to complete the activity, the number of incorrect attempts, the activity set number and the difficulty level the selected word of the activity. A dynamic variable named `average` is used to store the calculated ratio between total incorrect clicks and total clicks. This value demonstrates the average number of wrong user attempts per word. This value is used to compare the number of incorrect attempts by the user for a given activity. The average time taken by the user per activity is calculated with ratio between the total time taken by the user for the activity set (`totaltimecount`) and the number of correct attempts by the user. With these two methods it can be determined whether the user is doing better than average on two different measurements, the time taken and the number of incorrect attempts; both of which are dynamic. These two calculations are used to ascertain which activity from which difficulty level will be chosen next. After the activity is chosen, these two values are then used to recalculate the dynamic limits for the next word. Five membership conditions (outlined below) are introduced to select an activity depending upon the various fuzzy logic rules which are executed within IF-THEN statements.

Average Clicks $_{(0,1)}$ = $(\text{total_incorr_click} / \text{total_click}) \times 10$, for $\text{total_incorr_clicks} \leq \text{total_clicks}$

Average Time Ratio $_{(0,x)}$ = $(\text{totaltimecount} / \text{total_corr_click})$, for $\text{totaltimecount} \geq \text{total_corr_click}$

The five membership functions are:

```
{if (incorr_num < average && timecount < goodtime),  
if (incorr_num > average && timecount < goodtime),  
if (timecount > goodtime && timecount < badtime && incorr_num < average),  
if (timecount > goodtime && timecount < badtime && incorr_num > average),  
if (badactiveset > 3)} THEN select a random activity for the user
```

Once the nested loop has been executed, the data rows (dr) is filled with the respective value from the variables, prevword, activesetnum, incorr_num and difflevel is added to the data table dataresult; for recording purposes. Depending upon the user performance, the good time and the bad time would be incremented or decremented by 0.1 seconds (Liukkonen, 2009). Only if the criteria in the nested loop that the time is greater than one second are met, the statements in loop are executed. If timeratio is less than goodtime then goodtime and badtime is decremented by 0.1 second. Else, if timeratio is greater than goodtime then goodtime and badtime is incremented by 0.1 second. Values of timecount and incorr_num are reset to one second and zero respectively. The yes button (buttonyes) and no button (buttonno) are disabled after statement execution.

The print function is called for the click even of the Print button (buttonprint) to print the datatable dataresult into an excel sheet for the record purpose of the admin up to the activities the user has completed. The moment the Print button is triggered, a Dialog box is expected to open with the file type of excel format (.xls) with the default name "Result". When the user has clicked on the "OK" button of the dialog box to print the report, the print function is called to convert the data table-dataresult into an excel sheet. The Ignore button is an optional button purely for admin button to ignore the words in the activity set of the type "High Frequency Words". It contains the same functions as described in the "Yes Button" code functionality, except it does not record the activity set, word, time/attempts for that particular word.

3.2 Reading Module

This module is similar to the listening module (common functions), but without the word appearing and with a few differences in functionality. On the click event of the Next button, the button count is incremented each time the button is pressed. The activity is enabled for every one second. To avoid resetting of the timer, it starts only when the button is pressed once. The "Yes Button" and the "No Button" are enabled and the activity word is displayed on the screen (refer Figure 3.3).

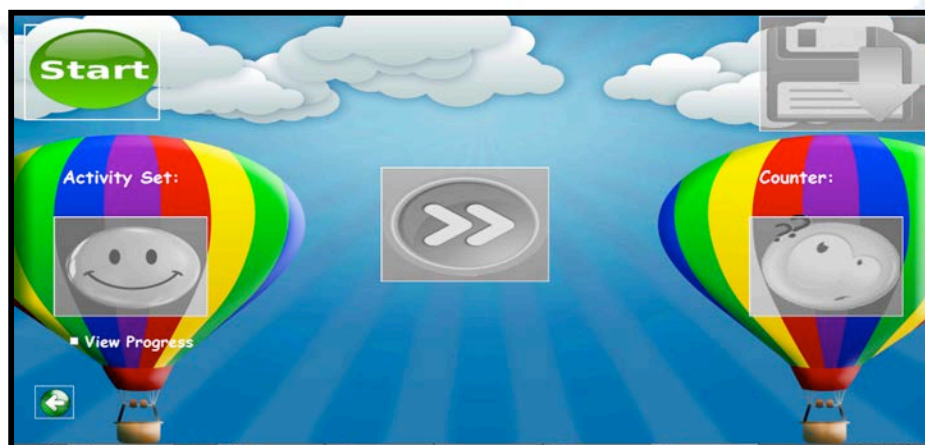


Figure 3.3. Illustrating the Form Load

3.3 Writing Module

HCR is implemented in the Writing Module in a simple, efficient way to aid in the development of motor skills for primary level children (aged 4 to 5 years). At this stage, the children are just beginning to be able to emulate letter shapes and crudely draw alphabetical characters. Accordingly, this module has the child progress through the alphabet, requiring him/her to write out each character. The GUI for this module is shown in Figure 3.4. Upon completion of the writing and selection of the next button, automatic recognition is invoked, where a message ‘Well Done’ appears if the character is correct, or ‘Try Again’ if it is incorrect.

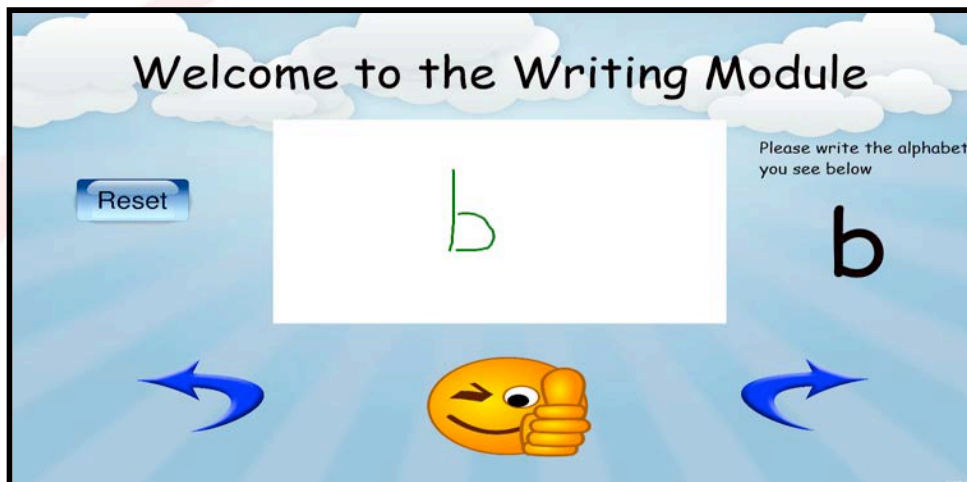


Figure 3.4. Figure illustrating how a child may write the letter “b”, using 2 strokes

Notable classes and controls for the Writing Module are discussed in brief. The class InkManager is part of the Windows.UI.Input.Inking Library, and it is used for the managing of stroke inputs, manipulation (erasing in this situation), and processing (HCR). This is done on a per stroke basis. InkManager is used for this application because it allows for a degree of freedom in recognizing what a character is and is not. Since an early years child uses the application, a child’s lack of developed motor skills is considered. Thus with InkManager, the distance between two strokes can be used in order to dictate when a new letter is written. For example, a child may not be able to write the letter “b” in one fluid movement yet, and therefore, may use two strokes. The Distance function is used to calculate the shortest distance between two lines using a basic Pythagorean equation, represented by the algorithm: $d = \text{Math.Sqrt}(\text{Math.Pow}((x2 - x1), 2) + \text{Math.Pow}((y2 - y1), 2))$. This calculation is used in the InkCanvas_PointerMoved function (implementing Equation 1) in order to determine when a new stroke should be recognized. The reset_Click_1 event happens when the Reset button is pressed (reset). When the button is pressed, the alphabet variable is set back to the letter a. The use of InkManager allows for this. The class Canvas Control is used to accept the strokes used for the InkManager class. Each stroke is saved as a child of the Canvas object. When the application is loaded, Event Handlers are initiated for the Canvas control to recognize when the mouse pointer is pressed, released, moved, and when the pointer leaves the allocated area.

Available strokes are retrieved from the InkManager and each stroke is passed through the Handwriting Recognizer. The character is then recognized and added to the results variable, which stores the values as a continuous string. The inkManager

mode is changed to Erasing, and the strokes are deleted. The mode is then switched back to Inking (drawing mode). The children of the panelcanvas object are also deleted. The results variable is then compared to the alphabet string to see if the user has written in the correct character. If both match, a message box appears confirming a correct match and alphabet is updated to the next character. If both variables do not match, a separate message box appears, prompting the user to try again. When the Back button is pressed, the strokes are erased. As soon as the strokes are erased, the alphabet variable is checked, and the user is taken back one letter. When the Next button is pressed, the strokes are erased in the same way as the ok button, the alphabet variable is then checked and the user is taken forward one letter. No character recognition takes place here.

4 User Testing and Results

Functionality testing was carried out extensively for the Reading, Listening and Writing Modules. For the Reading and Listening Modules, the five categories of performance were evaluated, from Good to Bad, based on time taken for each word (logged) and the number of incorrect attempts. Records of the activity set, difficulty levels and written comments were also evaluated for expected versus actual results. These results were attained through use of the software by Ms Sarah Bennett, in the classroom, for a variety of child skill levels. Each module performed well, moving the child between activities and sets, based on their performance indicators. Distance function computation within the Writing Module, as described previously, with the HCR algorithm, enabled classification of the user's response on the canvas, within the writing module. Although the child's handwriting was sometimes recognizable by the instructor as correct, although it differed from the boundaries for classification by the automated HCR program, there was an override function available, wherein the instructor could manually override the automatic grading, if there was a disparity. The HCR algorithm was tested against several deviations of input, and misclassification infrequently occurred. Where misclassification did exist, the writing was scrutinized by the instructor and did not fit within the acceptable limits of the HCR.

4.1 User Testing

The testing was carried out on a particular kindergarten class of around 20 children, taught by Ms. Sarah Bennett. The development team had set up the touchscreen monitor at a study corner, connected to the laptop where the application was stored. The children were offered turns, one after the other, to come and take a lesson in their most favorite module. With the presence of Ms. Sarah Bennett, the testing of the chosen module was conducted and the development team recorded the user interaction as well as the answers through videos. After 14-15 children took their turns and the modules had been tested, difficulties and complications were noted and addressed. Testing spanned for two hours and at the conclusion, feedback were taken. The children who tried the application were asked regarding their experience and the feedback received was positive. Most of them praised the colorful background and attractive icons which gave them a feeling of playing a game. They also highlighted the fact that they found it fun to learn the lessons as they intuitively knew what buttons to press at what time. Ms. Sarah Bennett was satisfied with the overall user testing and mentioned that the children actually enjoyed their interaction with the application. Feedback for additional functionality was addressed. A manual override in writing module to progress towards the next word. Within the Reading and Listening modules, chang-

ing label names, adding count for incorrect attempts, replacing icons. Steps pursued in order to avoid any sort of technical crash during the running of the application. A pause button functionality which can be used in case the application needs to be stopped for urgent reasons. A method to avoid occurrence of high-frequency words in the listening module from the database.

The second phase of the testing was conducted in the same class of children as the first. Children who were not a part of the first phase were selected and given turns, one after the other. And complying with it, different combination of children, categorized as good, average and weak, were selected. They now had to undertake all the three modules in the application at a single turn. Hence a realistic assessment environment was simulated. With the presence of Ms. Sarah Bennett, the user testing commenced and the development team observed and took videos for further analysis. Around 9-10 children went through the lessons and it spanned over two hours. Fixes suggested by the expert were implemented for this phase. Feedback from the children were positive in this scenario as well. They pointed out that they learnt new words and would actually remember them from now on. Furthermore, the children noted that they did not get bored during the testing duration and would love to do it again. However, some of them complained that they could not trace out alphabets at a single try because of the friction in the touchscreen monitor. At the end of the testing, it was decided in coordination with DIS that the application, touchscreen and a user manual was provided to Ms. Sarah Bennett for use, further analysis and reporting.

Questionnaires were handed to teachers at DIS in response to using the application. Questions and answers were as follows. 1.) What do you feel this application brings to the classroom? 2.) Any difference you've found using this technology in comparison to the traditional method in books from a teachers point of view? e.g.) reducing paper work, Automated syllabus? 3.) As a teacher, do you find any difference in the progress level of the child in using the application than the traditional methods used in the classroom? 4.) With a scope for future development, do you think this at the current stage this application would significantly aid the methods of teaching the curriculum? If yes, how? Responses were mostly positive: a bright and colorful application. Children enjoy the visual stimulus; it is a great activity and does not feel like assessment for the children. A more interactive approach is required; children want to play and see this more of a fun activity than being assessed. This way is far more efficient but data needs to be easily accessible and backed up. Children are retaining the words quicker and get to recognize the CVC (consonant/vowel/consonant) words by sight after they reappear and already been sounded before. Parents would also find it useful and informative (great that it is randomized). A quick and easy to access and was good to use as a practice/revision activity but not for recording guidance.

5 Conclusions and Future Work

The application has been developed as a supportive software tool, to assist English instructors and parents with teaching ESL to early years; providing assistive, interactive and automated computer-based activities to develop skills in English Literacy and assess the child in real-time. Although several applications exist for learning the basic skills in English for children at a young age, this software provides several benefits

over existing solutions, such as advancement in activity sets based on the child's real-time performance and the application of artificial intelligence (fuzzy logic) at the back-end to achieve this via automatic assessment. Three modules are offered to help with Reading, Writing and Listening. The software significantly reduces the manual role that instructors and parents play in administering, monitoring and storing results, as well as results retrieval. The software is also based on the UK curricula and has been exhaustively tested by the developers for functionality, as well as an expert early years teacher, with children from a variety of linguistic abilities. Test results show that the software is working soundly, and is accepted by the children and teachers that have been using it in the classroom environment. Future developments may include recording from the Writing Module automatically; currently this is a manual function, although the file for entry is provided for the instructor on the installation CD. User-driven skipping through activity sets, for manual entry of the desired start-activity (other than the first, which is the default), would also be beneficial to the instructor. Removal of high frequency words from the database, upon user-initiation would offer more flexibility to the administrator. Speech to text recognition is another identified extension of the work.

References

- Hsieh T.C, Wang, T, Su C.Y and Lee M.C. (2012). A Fuzzy Logic-based Personalized Learning System for Supporting Adaptive English Learning. *Educational Technology & Society*. 15 (1), pp. 273-288.
- Intellijoy. (2013). Kids ABC Phonics. Available: <https://play.google.com/store/apps/details?id=com.intellijoy.andreoid.phonics&feature=top-paid>. Last accessed 21 January 2013.
- Intellijoy. (2012). Kids Learn to Read (Preschool). Available: <https://play.google.com/store/apps/details?id=com.intellijoy.android.reading&feature=top-paid>. Last accessed 21 January 2013.
- Kim S.W, Kim K Lee J.H and Cho D.D (2001). Application Of Fuzzy Logic To Vehicle Classification Algorithm In Loop/Piezo-Sensor Fusion Systems. *Asian Journal of Control*, 3, pp. 64-68.
- Kao, C.C and Shih, A.J. (2008). Design and tuning of a fuzzy logic controller for micro-hole electrical discharge machining. *Journal of Manufacturing Processes*, 10, pp. 61-73.
- Liukkonen, K.T.N. (2009). Human Reaction Times as a Response to Delays in Control Systems. *Information Processing Science*. 1 (2), pp. 21-34.
- Tenlin Studios, LLC. (2013). Licking Letters. Available: <https://itunes.apple.com/us/app/licking-letters/id431550762?mt=8>. Last accessed 21 January 2013.
- Yusof, N, Ahmad N.B, Othman M.S and Nyen, Y.C. (2012). Fuzzy Inference System - Theory and Applications. A Concise Fuzzy Rule Base to Reason Student Performance Based on Rough-Fuzzy Approach. In: Azeem, M.F Fuzzy Inference System - Theory and Applications. Croatia: InTech. pp. 504.

