Web Based Intelligent Tutoring System in Sudoku for Logic Training

(Volume 1 of 1)

Student Name : Hui Wing Sze
Student No. : 
Programme Code : BScCS
Supervisor : Dr Kwok, Lam For
1st Reader : Dr Lee, Ka Chun Kenneth
2nd Reader : Dr Yu, Yuen Tak
I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Web based Intelligent Tutoring System in Sudoku for Logic Training

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: ________________________________  Signature: ________________________________
Student ID: ________________________________  Date: ________________________________
ABSTRACT

A cast number of players is unaware of logic skills benefit from enjoying the international trended puzzle game – Sudoku. In this project, we proposed an Intelligent Tutoring System (ITS) that aims to develop the logic ability of players by solving the Sudoku Puzzle steps by steps. All of logical rules used and its deductive influence in solving the puzzle are reviewed or given by hints. The expert contains the domain knowledge on Sudoku, which is used to uncover all the cell values. The tutor model provides hints, which is most valuable, to the learner. The student model keeps tracks of information on the learner performance. It also summarizes and evaluates the performance of student.
ACKNOWLEDGEMENT

I would like to convey my deepest and grateful thanks to my supervisor Dr. L. F. Kwok for his continuous support and help in completing the whole Final Year Project. Based on his previous experiences on information technological researches, he provided lots of precious advices in the development of Intelligent Tutoring System. Additionally, his expertise in the educational system helped and guided me in designing and implementing the significant requirements for the project.

Additionally, I would like to express my thanks to my classmates Peggy, Keith, Eric, Sharon, Jan, Kate, Edmond, Kato and Kandace for their precious advices and encouragement in the completion of the project.
# TABLE OF CONTENT

STUDENT FINAL YEAR PROJECT DECLARATION ......................................................................................... I

ABSTRACT ............................................................................................................................................... II

ACKNOWLEDGEMENT ............................................................................................................................. III

TABLE OF CONTENT ............................................................................................................................... IV

LIST OF FIGURES ..................................................................................................................................... VIII

LIST OF TABLES ........................................................................................................................................ X

INTRODUCTION .......................................................................................................................................... 1

1.1 FUNDAMENTAL INFORMATION OF PROJECT.................................................................................. 1

1.1.1 Logic................................................................................................................................................ 1

1.1.2 Sudoku ............................................................................................................................................. 2

1.1.3 Logical Deduction through Sudoku ............................................................................................... 3

1.1.4 Intelligent Tutoring System................................................................................................................ 4

1.2 PROJECT ORIGINATION .................................................................................................................... 5

1.2.1 Overview of Yu(2007) Project.............................................................................................................. 5

1.2.2 Improvements of Yu(2007) Project ....................................................................................................... 7

1.3 OBJECTIVES OF THE PROJECT ......................................................................................................... 8

1.4 CHALLENGES OF THE PROJECT ...................................................................................................... 9

1.5 REPORT ORGANIZATION .................................................................................................................. 10

REVIEWS ON LOGIC, SUDOKU & ITS ................................................................................................. 11

2.1 LOGIC DEVELOPMENT .................................................................................................................... 11

2.1.1 Validity of a Deductive Argument .................................................................................................. 11

2.1.2 Development of Logical Mind ......................................................................................................... 12

2.2 LOGIC TRAINING BY SUDOKU ......................................................................................................... 12

2.2.1 Acquisition of Sudoku Solving Knowledge ..................................................................................... 12

2.2.2 Logical Rules in Solving Sudoku Puzzle ......................................................................................... 13

2.2.3 Logical Rules Deduction of Sudoku in Student Mind ................................................................... 13

2.3 INTERPRETATION OF LOGICAL RULES .......................................................................................... 14

2.3.1 Interpretation of Single In Group ..................................................................................................... 14

2.3.2 Interpretation of Hidden Single ....................................................................................................... 16
3.5.1 Recorded Information ................................................................. 47
3.5.2 Evaluation Strategy ................................................................... 48
3.6 KNOWLEDGE REPRESENTATION FOR EXPERT MODEL .................. 49
  3.6.1 Fundamental Facts on Rule Based Knowledge Representation ......... 49
  3.6.2 Rules Based Representation of Logical Rules .............................. 50

DESIGN OF THE WEB BASED ITS IN SUDOKU ................................................. 53

4.1 DESIGN OF APPLICATION INTERFACE ........................................... 53
  4.1.1 Overview the Design of Application Interface ............................. 53
  4.1.2 Design of the Game Grid ......................................................... 55
  4.1.3 Game Information and Settings Panel ...................................... 56
  4.1.4 Instructions and Solving Procedure Box .................................... 58

4.2 DESIGN OF EXPERT MODEL ........................................................... 59
  4.2.1 Control Mechanism of Expert Model ....................................... 60
  4.2.2 Solving Sudoku Ability of Expert Model .................................. 62

4.3 DESIGN OF TUTOR MODEL ............................................................. 64
  4.3.1 Operations design of tutor model .......................................... 65
  4.3.2 Instruction Provision Mechanism ........................................... 65
  4.3.3 Initialization for Automatic Instruction .................................... 67

4.4 DESIGN OF STUDENT MODEL ......................................................... 69
  4.4.1 Operations design of the student model ................................... 69
  4.4.2 Formulation of Graphs ........................................................... 71

4.5 DESIGN OF DATABASE ................................................................. 73

IMPLEMENTATION OF THE WEB BASED ITS IN SUDOKU .............................. 76

5.1 DEVELOPMENT TOOLS AND ENVIRONMENT ................................ 76
5.2 IMPLEMENTATION OF LOGICAL RULES IN EXPERT MODEL ............ 77
  5.2.1 Algorithm in implementing SG ............................................. 78
  5.2.2 Algorithm in implementing NS ............................................. 79
  5.2.3 Algorithm in implementing HS ............................................. 80
  5.2.4 Algorithm in implementing IR ............................................. 80
  5.2.5 Algorithm in implementing NP and HP ................................... 82

5.3 IMPLEMENTATION OF AUTOMATIC INSTRUCTIONS PROVISION IN TUTOR MODEL ............................................................ 83
5.4 IMPLEMENTATION THE EVALUATION GRAPHS IN THE STUDENT MODEL ............................................................ 84
5.5 IMPLEMENTATION OF WEB PAGE ................................................ 85
5.6 COMMUNICATIONS BETWEEN APPLET AND DATABASE ........................ 86

DISCUSSION AND EVALUATION OF ITS IN SUDOKU ..................................... 88

6.1 DISCUSSION OF A PERFECT ITS IN SUDOKU ................................... 88
6.2 Effectiveness of ITS Adaptation in This Project ................................................................. 88
6.3 Discussion on Insufficient Adaptation of ITS ................................................................. 89
   6.3.1 Lack of Advance Rules in Expert Model ................................................................. 90
   6.3.2 Unreal Tailor-Made Instructions in Tutor Model .................................................... 90
   6.3.3 Inefficient in Time Management for Automatic Instructions ................................. 91
   6.3.4 Incomprehensive Evaluation for Student Model .................................................... 91
6.4 Testing of the Application ......................................................................................... 92
   6.4.1 Background of Students .......................................................................................... 92
   6.4.2 Testing procedures .................................................................................................. 93
   6.4.3 Testing Results Evaluation .................................................................................... 93
6.5 Further Testing for Evaluation ................................................................................. 95
   6.5.1 Test the learning function of application ............................................................... 95
   6.5.2 Test the mental state against logical learning ....................................................... 96

CONCLUSION OF SUDOKU IN ITS ...................................................................................... 97

7.1 Achievements of Web Based Sudoku in ITS ............................................................... 97
   7.1.1 Project Objectives Fulfillment ................................................................................. 97
   7.2.2 Successfully Implemented Functions ...................................................................... 99
7.2 Limitations of Web Based Sudoku in ITS ................................................................. 100
7.3 Further Enhancement of Proposed Project ................................................................... 101

REFERENCES .................................................................................................................. 102

APPENDIX A DESIGN CLASS DIAGRAM .............................................................................. 104

APPENDIX B MONTHLY LOG ............................................................................................ 105
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A partially complete Sudoku puzzle &amp; solved Sudoku puzzle</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Logical Rules Deduction in a Sudoku puzzle</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Game board interface of ITS in Sudoku for Logic Training</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Design of ITS in Sudoku for Logic Training</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Categories of logical rules</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Sample Sudoku puzzle that applied SIG</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>Sample Sudoku puzzle that applied HS</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Sample Sudoku puzzle that applied HS</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Sample Sudoku puzzle that applied IR</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Sample Sudoku puzzle that applied NP</td>
<td>20</td>
</tr>
<tr>
<td>2.7</td>
<td>Sample Sudoku puzzle that applied HP</td>
<td>21</td>
</tr>
<tr>
<td>2.8</td>
<td>Application interface of Sudoku Online</td>
<td>23</td>
</tr>
<tr>
<td>2.9</td>
<td>Application interface of Sudoku hint</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>Application interface of Sudoku Tiger</td>
<td>25</td>
</tr>
<tr>
<td>2.11</td>
<td>CAI instructions provision flow</td>
<td>27</td>
</tr>
<tr>
<td>2.12</td>
<td>Components of Expert System</td>
<td>28</td>
</tr>
<tr>
<td>2.13</td>
<td>Procedures of construction in knowledge base</td>
<td>29</td>
</tr>
<tr>
<td>2.14</td>
<td>Questionnaires for the measurement of student emotion</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Overall Architecture of Web based ITS in Sudoku</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Conceptual application interface design</td>
<td>39</td>
</tr>
<tr>
<td>3.3</td>
<td>A typical semi-solved Sudoku conceptual representation in Expert Model</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>A typical example of Sudoku skill matrix</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Interface design of the ITS in Sudoku</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Enlarged image of part of the game grid</td>
<td>55</td>
</tr>
<tr>
<td>4.3</td>
<td>Warnings for handling apparent errors from students</td>
<td>56</td>
</tr>
<tr>
<td>4.4</td>
<td>Detailed designs of game information and settings panel</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>Detailed descriptions of instructions and solving procedure box</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>Components of Expert Model</td>
<td>59</td>
</tr>
<tr>
<td>4.7</td>
<td>Overall control mechanism of Expert Model</td>
<td>60</td>
</tr>
<tr>
<td>4.8</td>
<td>A list of Sudoku steps matrix</td>
<td>61</td>
</tr>
<tr>
<td>4.9</td>
<td>Sudoku skill matrix of the application</td>
<td>62</td>
</tr>
<tr>
<td>4.10</td>
<td>Control mechanism of Sudoku puzzle solving</td>
<td>63</td>
</tr>
<tr>
<td>4.11</td>
<td>Components of Tutor Model</td>
<td>64</td>
</tr>
<tr>
<td>4.12</td>
<td>Sample instructions provision format</td>
<td>66</td>
</tr>
<tr>
<td>4.13</td>
<td>Number of solutions per time band</td>
<td>67</td>
</tr>
</tbody>
</table>
FIGURE 4.14 COMPONENTS OF TUTOR MODEL................................................................. 69
FIGURE 4.15 REGISTRATION FORM OF THE APPLICATION ........................................... 70
FIGURE 4.16 SAMPLE OF PERFORMANCE CHART ...................................................... 71
FIGURE 4.17 SAMPLE OF HINTS USAGE CHART ....................................................... 72
FIGURE 4.18 SAMPLES OF RULES; MISCONCEPTIONS AND HINTS RELATION CHART .............................................................................................................................. 72
FIGURE 4.19 DESIGN OF DATABASE FOR THE SUDOKU IN ITS .................................. 74
FIGURE 4.20 DESIGN OF DATABASE CONNECTION ....................................................... 75
FIGURE 5.1 ALGORITHM OF SG.................................................................................... 78
FIGURE 5.2 ALGORITHMS OF NS ............................................................................... 79
FIGURE 5.3 ALGORITHMS OF HS ............................................................................... 80
FIGURE 5.4 ALGORITHM OF IR.................................................................................... 81
FIGURE 5.5 ALGORITHM IN SEARCHING FOR POSSIBLE VALUES PAIRS ..................... 82
FIGURE 5.6 ALGORITHMS OF NP AND HP ................................................................. 83
FIGURE 5.7 ALGORITHM OF AUTOMATIC INSTRUCTIONS PROVISION ...................... 84
FIGURE 5.8 SAMPLE CODING FOR GRAPH GENERATION .......................................... 85
FIGURE 5.9 LOGIN PAGE OF THE PROPOSED PROJECT ............................................ 86
FIGURE 5.10 THE doPost FUNCTION OF THE SERVLET FOR HANDLING RETRIEVAL ....... 87
FIGURE 6.1 RESULTS OF TESTING ............................................................................... 94
FIGURE 6.2 AVERAGE PREDICTED LEARNING CURVE OF STUDENT .............................. 96
LIST OF TABLES

TABLE 1.1 FUNDAMENTAL TERMS OF SUDOKU ................................................................. 2
TABLE 1.2 FUNCTIONS SUPPORTED BY ITS IN SUDOKU FOR LOGIC TRAINING .............. 6
TABLE 2.1 DEDUCTIVE ARGUMENTS OF SIG ................................................................. 15
TABLE 2.2 DEDUCTIVE ARGUMENTS OF HS ................................................................. 17
TABLE 2.3 DEDUCTIVE ARGUMENTS OF NS ................................................................. 18
TABLE 2.4 DEDUCTIVE ARGUMENTS OF IR ................................................................. 19
TABLE 2.5 DEDUCTIVE ARGUMENTS OF NP ................................................................. 21
TABLE 2.6 DEDUCTIVE ARGUMENTS OF HP ................................................................. 22
TABLE 2.7 FRAME BASED REPRESENTATION OF A BOOK ............................................ 31
TABLE 2.8 ABILITIES IN MEASURING COGNITIVE STATE ........................................... 34
TABLE 3.1 FUNDAMENTAL TERMS OF SUDOKU PUZZLES ............................................ 49
TABLE 3.2 RULE BASED KNOWLEDGE REPRESENTATION OF LOGICAL RULES ............. 51
TABLE 4.1 SAMPLE STORAGE OF STRATEGY GRAPH .................................................. 61
TABLE 5.1 APPLICATION DEVELOPMENT ENVIRONMENT ............................................. 76
TABLE 5.2 DEVELOPMENT TOOLS OF APPLICATION .................................................. 77
TABLE 6.1 CHARACTERISTICS OF PROPOSED ITS SYSTEM ......................................... 89
CHAPTER ONE
INTRODUCTION

Sudoku has long been considered as a logical training game. With the rapid growth of computing technology and artificial intelligent, Intelligent Tutoring System is proposed in educational aspects. The proposed project is the combination of the Intelligent Tutoring System and the logical training in Sudoku. In this Chapter, some fundamental concepts of important terms in the project are introduced. Afterwards, the background and challenges of the project are described.

1.1 FUNDAMENTAL INFORMATION OF PROJECT

In this Section, the notion of logic and Sudoku are described. Afterwards, the deduction of logic rules from Sudoku is discussed. Additionally, Intelligent Tutoring System (ITS), which instructs students in logical development, is introduced.

1.1.1 Logic

Logic was introduced by Aristotle, who built up the fundamental knowledge of logic in philosophy. It was considered as the systematic approach to determine the correctness of an argument. Based on the modern psychology researches, the logical ability to draw the conclusion is developed through experiences, examples and instructions (Munson, Conway and Black 2004). Logic is divided into four categories including formal logic, informal logic, symbolic logic and mathematical logic. In this project, Sudoku, a kind of symbolic logical training tool, is proposed. Although Sudoku contains numeral values, it considers as symbolic logic instead of mathematical logic because there is no mathematic relationship between the numeral values.
1.1.2 Sudoku

Sudoku, which means a single letter, is a widely known logic based number placement puzzle. It was invented by an American architect, Howard Garns, in 1979 with original names “Number in Place”. It was popular since 1986 when the name “Sudoku” is given by the Japanese puzzle company Nikoli and became an international trend since 2005. Sudoku is a grid of 9X9 cells in which each row, column and 3X3 box can only accommodate digits from 1 to 9 exactly once. Table 1.1 demonstrates the fundamental terms applied in this project.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>A series of 9 cells placed in the same horizontal line in the grid</td>
</tr>
<tr>
<td>Column</td>
<td>A series of 9 cells placed in the same vertical line in the grid</td>
</tr>
<tr>
<td>Box</td>
<td>Any one of the nine squares covered by 3x3 cells that is equally divided into a grid</td>
</tr>
<tr>
<td>Scope</td>
<td>It refers to row, column or box.</td>
</tr>
<tr>
<td>Cell</td>
<td>It is the place that accommodate a value inside the grid</td>
</tr>
<tr>
<td>Grid</td>
<td>9x9 cells combined into a grid</td>
</tr>
<tr>
<td>Definite Value</td>
<td>The correct value of a cell</td>
</tr>
</tbody>
</table>

*Table 1.1 Fundamental Terms of Sudoku*

The puzzle provides a partially complete grid to players as clues for solving the puzzle. Examples of Sudoku puzzle are shown in Figure 1.1. According to the studies and researches conducted by Herzberg and Murty (2007), Sudoku is claimed to be an effective logical development tool.

![Figure 1.1 A partially complete Sudoku puzzle & solved Sudoku puzzle.](image-url)
1.1.3 Logical Deduction through Sudoku

A number of logic deduction rules have actually been applied during the Sudoku puzzle solving process. However, people are usually unaware of rule deduction through logical games. This circumstance is interpreted by the observations of Leighton (2004), “reasoning works behind the scenes, coordinating ideas, premises, or beliefs in the pursuit of conclusions. These conclusions may sometimes find their way to the surface but, rather, stay beneath the surface and function internally as antecedent conditions that feed into chains of productions for problem solving.” Figure 1.2 shows a typical example for the interpretation of logical deduction through Sudoku.

![Figure 1.2 Logical Rules Deduction in a Sudoku Puzzle](image)

When we examine the value of cell shaded in the figure, values 2, 4, 6, 7, 8 and 9 are not in our consideration, since they are already given in the same scope as the target examined cell. Therefore, the possible values of cell are only 1, 3 or 5. This strategy
in determining the value of a cell has already undergone a logical rule deduction procedure. It is because the possible values of the target cell are determined based on the constraint of Sudoku puzzle (Every digit in the same scope can use once exactly). In this project, an intelligent tutoring system is employed to facilities the logic development of players. The intelligent tutoring system is introduced in the next subsection.

1.1.4 Intelligent Tutoring System

Computer-Assisted Instructions (CAI) is recognized as the predecessor of Intelligent Tutoring System (ITS). It refers to any computing system that acts as a coach to teach students with specific knowledge domain by predefined instructions and feedback. In 1980s, Jonassen (2000) suggested that most of the existing CAIs were drilled-and practical systems, which simply provide predefined feedback for checking the accuracy for student responses.

Meanwhile, applications were normally equipped with decision making strategies based on the rapidly developing Artificial Intelligent (AI) technology. Expert model, which supports the decision making ability of the system, was applied in the application. As a result, the most sophisticated form of CAI - Intelligent Tutoring System was proposed by Sleeman and Brown in 1982. It does not only record and analyze the performance of student, but also provide real time feedback and instructions for students based on their learning progress and ability.

There are three major challenges have to overcome in order to educate student knowledge in a specific domain. Firstly, the preceptor should be capable of tackling all the problems of the specified knowledge domain. Secondly, the preceptor should provide effective and efficient instructions with respect to student’s ability and learning state. Thirdly, the preceptor should record and evaluate student performance for providing suitable instructions to a student. These three essential functions are the main duties of the expert model, tutor model and student model of ITS respectively.
Chapter One

Introduction

The thought of applying ITS in Sudoku to support the logical development was first proposed by Yu(2007). In section 1.2, the need for developing an improved version of ITS in Sudoku for Logic Training is discussed.

1.2 PROJECT ORIGINATION

The idea of Web Based Intelligent Tutoring System in Sudoku for Logic Training was originated from the Intelligent Tutoring System in Sudoku for Logic Training (Yu, 2007). In this section, a brief overview of Yu’s (2007) project is described. Meanwhile, the improvements of the project are discussed.

1.2.1 Overview of Yu(2007) Project

Figure 1.3 demonstrates the system interface proposed by Yu (2007). There are three main areas inside the application interface. They are the game board area, game information panel and instruction box. Game board area provides an area for students to work on Sudoku puzzles. Game information panel provides data of the game. Instruction box displays the hints generated by the system under student’s request.
Chapter One

Introduction

This project aims to develop an application which enhances the logical mind of students. The design pattern of the system is based on a typical ITS. The functions supported by this application are summarized in Table 1.2.

![Figure 1.3 Game Board Interface of ITS in Sudoku for Logic Training](image)

<table>
<thead>
<tr>
<th>Module</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert Model</td>
<td>Supports puzzle solving with three well tested declaration rules and one elimination rule</td>
</tr>
<tr>
<td></td>
<td>Solves the Sudoku with pre-defined rule priority</td>
</tr>
<tr>
<td>Tutor Model</td>
<td>Provides immediate feedback for every solving step</td>
</tr>
<tr>
<td></td>
<td>Provides real time instructions when request by students</td>
</tr>
<tr>
<td>Student Model</td>
<td>Stores student profile</td>
</tr>
<tr>
<td></td>
<td>Records game progress information</td>
</tr>
</tbody>
</table>

Table 1.2 Functions supported by ITS in Sudoku for Logic Training
Apart from the functions listed, the application also provides an interface which allows students to interact with the system. Additionally, it contains a file configuration system which supports the storage of student performance and information on the client side.

The overall architecture of the application extracted from Yu’s (2007) report is shown in Figure 1.4. It has successfully implemented the concept of ITS since the three basic modulus including expert model, tutor model and student model are interacted with each other. Meanwhile, functions supported by each module are well defined and categorized.

![Figure 1.4 Design of ITS in Sudoku for Logic Training](image)

After a brief review on the Yu’s (2007) project, some improvements are found to be worth in developing a new application under the original concept.

### 1.2.2 Improvements of Yu(2007) Project

As discussed (Section 1.2.1), Yu’s (2007) project is based on the concept of a typical ITS model. He has proposed a java application to facilitate the purpose of the project. However, some improvements are worth implementing.
To begin with, a java application is not convenience for distribution and maintenance. Configurations such as the jar file of the application are required before students can use it. Additionally, new Sudoku puzzles are required to send to the students before they can enjoy some updated game. Moreover, the data collection for the study of the student model is also ineffective and inefficient as the files are required to send back to the programmer for further research.

In addition, ITS model in Sudoku is not fully developed. Firstly, the expert model is only capable of solving the puzzle through four out of nine well defined logical rules. Secondly, the tutor model does not act like initiative and sensitive as human coach since no instruction is given based on student learning progress. Lastly, the student model does not support any evaluation of student learning performance.

The inconveniences and unsatisfied functions of ITS model result in the proposal of Web Based Intelligent Tutoring System in Sudoku for Logic Training.

1.3 Objectives of the Project

Based on the problems of ITS in Sudoku for Logic Training described above (Section 1.2.2), it is necessary to develop an improved version of the existing project. The main objective of the proposed project is to provide a platform for students to develop the logic mind in solving Sudoku puzzle based on ITS model.

Apart from the main objectives, there are several additional objectives to be achieved in implementing this proposal. In other words, these additional objectives are the scope of the proposed project. They are

1. Enrich my knowledge on the deduction of the logic, Sudoku rules and intelligent tutoring system
2. Derive high level requirements for logic training in Sudoku based on the concept of an intelligent tutoring system
3. Design an application with functions that facilitate the logic training in Sudoku and based on ITS model
4. Implement ITS model in the selected environment and with well defined algorithms
5. Summarize the characteristics of a good ITS. Perform testing and evaluate how ITS model is adapted in the proposed application

The proposed project also provides both Chinese and English interface. The challenges of implementing these improvements are discussed in subsection 1.4.

**1.4 CHALLENGES OF THE PROJECT**

Since the collection of information in existing application is inconvenience and configurations are required in using the application, the proposed application aims to solve these problems by converting existing java application into a java applet which launches in the website. However, it is a huge reconstruction. Only logical rules performing functions and rules interpretation functions are reusable. A control layer should be included to control the communications and interactions between the expert model, tutor model and student model since the organization of classes in the previous project is confusing. They are not implemented with a good organization of ITS. Additionally, the presentation of information originally stored in the file configuration system should be reconstructed and recorded in the database.

Moreover, the two additional elimination rules are complex in algorithm. It is because the implementation of these two rules, Naked Single and Hidden Single, are not straight forwards. A cast number of testing is required to ensure the correctness and accuracy of the rules.

Furthermore, only a few numbers of existing tutor model are capable of providing real time and automatic instructions for students. Lots of researches and studies have to be done to ensure the time providing the automatic instructions are as sensitive and initiative as the human coach.
Finally, there are only a few numbers of researches proposed a complete, efficient and effective evaluation method for the student model. Therefore, almost none of the researches are suitable in evaluating the logic development in Sudoku through ITS. In order words, lots of testing should be done after the implementation of the project to ensure the effectiveness of developing logic mind through the proposed application.

1.5 REPORT ORGANIZATION

After the introduction of the proposed application, we found that many challenges are required to tackle. Chapter 2 of this report provides a detailed literature review on ITS and some existing application of Sudoku. In Chapter 3, the conceptual schema of the entire application is described. Afterwards, the design and game flow of the project is discussed in Chapter 4. Then, Chapter 5 describes the detailed implementation of the system components including the webpage, database and applet application. Chapter 6 interprets the results of the system and discusses on the insufficiency of the system. Finally, Chapter 7 describes the achievements, limitations and further enhancement of the project.
CHAPTER TWO

REVIEWS ON LOGIC, SUDOKU & ITS

In this chapter, a detailed and comprehensive literature review is conducted via Logic, Sudoku and ITS. These subjects are the important components of the proposed application. To begin with, logic development is discussed.

2.1 LOGIC DEVELOPMENT

The main intention of logic is to judge the correctness of an argument. In other words, logic is developed through the understanding of an argument. On the side, a number of researches and studies had been conducted about the development of logical mind.

2.1.1 Validity of a Deductive Argument

Argument is divided into several categories but the structures of them are the same. Argument consists of a conclusion and one or more premises. Premise is a claim to support the conclusion with reasons. The logical development in Sudoku is based on the judgment on the correctness of deductive argument. Deductive argument is considered as valid if and only if both the conclusion and its premises are valid. In other words, for a valid deductive argument, if all the premises are true, the conclusion must also be true and vice versa (Musgrave, 1989).

Logic is the innate abilities for all human beings and is indivisible from our daily life. Psychologists suggested that logical mind can be developed and improved.
2.1.2 Development of Logical Mind

In the late 19th Century, Plato introduced that human can easily apply and be taught the inferential rules (Fong and Nisbett 1993). Afterwards, Nisbett and his colleagues made use of what Plato had claimed for his research in finding the most efficient and sufficient way to give inferential rules for instructions (Nisbett, Fong, Lehman and Cheng 1987). Then, Lehman further studied the correlation between undergraduate education and reasoning ability (Lehman and Nisbett 1990). He discovered that practicing mathematics improves the reasoning abilities, especially condition reasoning. Consequently, logic is developed through experiences in which the problem solving instructions stimulate and enrich the logical mind. In other words, training and practices through examples and instructions are the essential components for the development of logic in human mind. In the next section, how logic can be developed by Sudoku is discussed.

2.2 LOGIC TRAINING BY SUDOKU

According to previous chapter (Section 1.1.3), it is generally accepted that Sudoku is a tool to develop logic. In this section, the source of Sudoku knowledge and categories of rules are indicated.

2.2.1 Acquisition of Sudoku Solving Knowledge

Since the popularity of Sudoku, media like magazines and web pages provide a cast number of methods and strategies in solving Sudoku puzzle. Undoubtedly, logical rules deduced from different media are varied. They are actually of same concept but different wordings. The knowledge acquisition of logic rules, premises and conclusion in this project are based on Sudoku Tiger and The Time Sudoku Book (2005). To conclude, there are altogether nine logic rules, which divide into three groups.
2.2.2 Logical Rules in Solving Sudoku Puzzle

It is undeniable that logical rules are formed based on the constraint of the problem. In Sudoku, the only constraint is to assign the 9x9 cells with digit one to nine exactly once in each row, each column and each box. Figure 2.1 reveals the logical rules that can be applied in solving a Sudoku puzzle.

![Figure 2.1 Categories of Logical Rules](image)

The most well known rules are the declaration rules and elimination rules. Declaration rules are used to determine a cell value while elimination rules are used to eliminate the impossible value of a cell. These two types of rules are the concentration of the project. Meanwhile, the figure demonstrates the difficulty of the rules. Students who use different priority of rules in solving a Sudoku puzzle result in different solving procedures. Therefore, many possibilities arise during the solving procedures of the Sudoku.

Most of the Sudoku puzzle can be successfully solved by the declaration rules and elimination rules. However, how the logical mind of the rules is deduced during the game? The answer of this question is described in the next subsection.

2.2.3 Logical Rules Deduction of Sudoku in Student Mind
As discussed, deductive arguments are the significant material for the improvement of logic mind of student. Nonetheless, studying deductive arguments by rote is not effective and efficient.

Students should experience deductive arguments through numbers of trials and errors in order to develop the reasoning skills. The quality of training will be increased if deductive arguments are provided immediately when students get mistakes.

Although the resources for solving Sudoku are varied, the discussions on reasoning steps (premises and conclusion) to solve the puzzle are rare. In other words, students are difficult to recover the deduction inferences. Even worse, some of them are unaware of the involvement of logic rules in solving a puzzle. The presentation of deductive arguments and interpretation of logic rules are illustrated in the section 2.3.

2.3 Interpretation of Logical Rules

In this section, the declaration rules and elimination rules are interpreted. For each of the logic rules, an example is provided to illustrate the application of the rule. Additionally, the premises which draw the conclusion steps by steps are also described.

2.3.1 Interpretation of Single In Group

Single In Group (SIG) is the simplest logic rule among others. This rule is formed based on the constraint that cells in the same scope should only accommodate digit from one to nine exactly once. Take the row highlighted in figure 2.2 as an example.
Digits 1, 2, 3, 4, 5, 6, 7, 8 have already accommodated by the cells of row 2. Therefore, the remaining cell (cell (2, B)) should accommodate digit 9. Similarly, the rule is also capable of solving the puzzle in column and box if they under the same precondition (There are already eight cells in the same scope having a declared value). Based on the information of Figure 2.2, sample deductive arguments of SIG are formed and shown in table 2.1.

<table>
<thead>
<tr>
<th>Effective Scope of SIG</th>
<th>Deductive Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
<td><strong>Premises 1</strong>: Each cell should accommodate a digit from 1-9</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 2</strong>: Digits 1-9 should only appear once in each scope</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 3</strong>: Digits 1, 2, 3, 4, 5, 6, 7, 8 are given in row 2</td>
</tr>
<tr>
<td></td>
<td><strong>Conclusion</strong>: Digit 9 should be in cell(2,B) of row 2</td>
</tr>
<tr>
<td><strong>Column</strong></td>
<td><strong>Premises 1</strong>: Each cell should accommodate a digit from 1-9</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 2</strong>: Digits 1-9 should only appear once in each scope</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 3</strong>: Digits 1, 2, 3, 4, 5, 7, 8, 9 are given in column D</td>
</tr>
<tr>
<td></td>
<td><strong>Conclusion</strong>: Digit 9 should be in cell(5,D) of column D</td>
</tr>
<tr>
<td><strong>Box</strong></td>
<td><strong>Premises 1</strong>: Each cell should accommodate a digit from 1-9</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 2</strong>: Digits 1-9 should only appear once in each scope</td>
</tr>
<tr>
<td></td>
<td><strong>Premises 3</strong>: Digits 1, 2, 3, 4, 5, 6, 7, 8 are given in box 9</td>
</tr>
<tr>
<td></td>
<td><strong>Conclusion</strong>: Digit 9 should be in cell(8,H) of box 9</td>
</tr>
</tbody>
</table>

*Table 2.1 Deductive Arguments of SIG*
2.3.2 Interpretation of Hidden Single

Hidden Single (HS) is another logic rule for declaration. It works whenever a digit is only possible for one cell in the same scope. Take figure 2.3 as an example to illustrate the rules.

In the highlighted box, there are three unsolved cells including cell (4, A), cell (4, B) and cell (6,A). Because of the cell shaded (cell (9, A)) contains a digit 9. Therefore, only one cell (cell (4, B)) can accommodate digit 9 in the box. Table 2.2 provides premises and conclusion of HS in different effective scope. The premises and conclusion of box scope in table 2.2 are based on the example puzzle of Figure 2.3.
Chapter Two

Reviews on Logic, Sudoku & ITS

<table>
<thead>
<tr>
<th>Effective Scope of HS</th>
<th>Deductive Arguments</th>
</tr>
</thead>
</table>
| Row                   | Premises 1: Each cell should accommodate a digit from 1-9  
                       | Premises 2: Digits 1-9 should only appear once in each scope  
                       | Premises 3: Digits 2,3,4 are possible unsolved cells of row 1  
                       | Premises 4: Only cell(1,A) where the digit 3 can accommodate  
                       | Conclusion: Digit 3 should be placed in cell(1, A)  
| Column                | Premises 1: Each cell should accommodate a digit from 1-9  
                       | Premises 2: Digits 1-9 should only appear once in each scope  
                       | Premises 3: Digits 1,3,5 are possible unsolved cells of column H  
                       | Premises 4: Only cell(6,H) where the digit 5 can accommodate  
                       | Conclusion: Digit 5 should be placed in cell(6, H)  
| Box                   | Premises 1: Each cell should accommodate a digit from 1-9  
                       | Premises 2: Digits 1-9 should only appear once in each scope  
                       | Premises 3: Digits 1,8,9 are possible unsolved cells of box 4  
                       | Premises 4: Only cell(4,B) where the digit 9 can accommodate  
                       | Conclusion: Digit 9 should be placed in cell(4, B)  

Table 2.2 Deductive Arguments of HS

2.3.3 Interpretation of Naked Single

Naked Single (NS) is considered as the most difficult declaration rule among others. However, this rule is simple if elimination tools are provided. Figure 2.4 demonstrates a typical Sudoku puzzle, and it is used to interpret the application of NS.

![Figure 2.4 Sample Sudoku Puzzle that applied HS](image-url)
Shaded Cell (5, E) is the Naked Single; since there is only one possible value can be placed in this cell. To find out the only possible digit of the cell, the digits appeared in the same scope with it is considered. Firstly, it is found that in Row 5, digits 2, 6, 7 are occupied. Secondly, digits 3, 4, 8, 9 are appeared in Column E. Thirdly, digits 3, 5, 7 have been used in Box 5. In other words, digits 2, 3, 4, 5, 6, 7, 8, 9 are accommodated in the cells of same scope. Only digit 1 can be accommodated by Cell (5, E). Table 2.3 illustrates deductive arguments of NS. Since NS is applied based on the information of three scopes, no effective scope is considered.

<table>
<thead>
<tr>
<th>Deductive Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premises 1: Each cell should accommodate a digit from 1-9</td>
</tr>
<tr>
<td>Premises 2: Digits 1-9 should only appear once in each scope</td>
</tr>
<tr>
<td>Premises 3: Digits 2, 6, 7 appear in row 5</td>
</tr>
<tr>
<td>Premises 4: Digits 3, 4, 8, 9 appear in column E</td>
</tr>
<tr>
<td>Premises 5: Digits 3, 5, 7 appear in box 5</td>
</tr>
<tr>
<td>Premises 6: Digit 1 is the only possible digit for cell (5, E)</td>
</tr>
<tr>
<td>Conclusion: Digit 1 should place in Cell (5, E)</td>
</tr>
</tbody>
</table>

**Table 2.3 Deductive Arguments of NS**

### 2.3.4 Interpretation of Intersection Reduction

Intersection Reduction (IR) is the first elimination rule that we have discussed so far. It is impossible to solve all the Sudoku puzzles by applying declaration rules alone. Intersection reduction is a significant logic rule in solving most of the non-trivial Sudoku puzzles. Figure 2.5 shows a sample puzzle for the interpretation of this useful technique.
Possible digits for the unsolved cells in row 2 include 1, 6, 7, 8 and 9. However, digit 7 is only possible for cell (2, G) and cell (2, H) which belong to the same box (Box 3). Therefore, digit 7 should either be the definite value of cell (2, G) or cell (2, H). In other words, unsolved cells in Box 3 excluding cell (2, G) and cell (2, H) are impossible to accommodate digit 7. Thereby, digit 7 is eliminated its possibility from cell (3, G) and cell (3, H). Table 2.4 describes the deductive arguments of IR.

<table>
<thead>
<tr>
<th>Effective Scope of IR</th>
<th>Deductive Arguments</th>
</tr>
</thead>
</table>
| Row                   | Premises 1: Each cell should accommodate a digit from 1-9  
Premises 2: Digits 1-9 should only appear once in each scope  
Premises 3: Digit 7 is one of the possible digits for unsolved cells in Box 3  
Premises 4: Digit 7 only occurs in cell (2, G) and cell (2, H) in Row 2  
Conclusion: Digit 7 cannot be accommodated by other rows of columns within Box 3 |
| Column                | Premises 1: Each cell should accommodate a digit from 1-9  
Premises 2: Digits 1-9 should only appear once in each scope  
Premises 3: Digit 6 is one of the possible digits for unsolved cells in Box 4  
Premises 4: Digit 6 only occurs in cell (5, A) and cell (6, A) in Column A  
Conclusion: Digit 6 cannot be accommodated by other rows of column within Box 4 |
2.3.5 Interpretation of Naked Pairs

Naked Pairs (NP) is an elimination technique that is considered to be the simplest to trace it out. It is because one of the conditions of applying NP is that pairs of possible values should be found in the same scope. Figure 2.6 shows a sample Sudoku puzzle in which NP can be applied.

![Sample Sudoku Puzzle that applied NP](image)

NP can be applied only if there are two possible values for a pair of cells in the same row or same column. In the sample provided, Cell (5, G) and Cell (5, H), which are of the same row (Row 5), only contains digits 3 and 4 as their possible value. It indicates that digit 3 is the definite value of Cell (5, G) and digit 3 is the definite value of Cell (5, H) or vice versa. In other words, other unsolved cells of the Row 5 are impossible to accommodate digit 3 and 4. Therefore, in this case, the digit 4 should eliminate its possibility in cell (5, A), cell (5, C) and cell (5, D). Table 2.5 proposes the premises and conclusion for develop the logical mind by NP.
Effective Scope of NP | Deductive Arguments
--- | ---
Row | Premises 1: Each cell should accommodate a digit from 1-9  
Premises 2: Digits 1-9 should appear once in each scope  
Premises 3: Digits 3 and 4 are possible in cell (5, G) and cell (5, H) of Row 5  
Conclusion: Digit 3 and 4 can be eliminated their possibilities from cells in Row 5 other than cell (5, G) and cell (5, H)
Column | Premises 1: Each cell should accommodate a digit from 1-9  
Premises 2: Digits 1-9 should appear once in each scope  
Premises 3: Digits 1 and 4 are possible in cell (5, D) and cell (8, D) of Column D  
Conclusion: Digit 1 and 4 can be eliminated their possibilities from cells in Column D other than cell (5, D) and cell (8, D)

**Table 2.5 Deductive Arguments of NP**

### 2.3.6 Interpretation of Hidden Pairs

Hidden Pairs (HP) is the most difficult elimination rule among those will be implemented in this project. HP works if hidden pairs are found in a row or column. Figure 2.7 demonstrates a sample Sudoku puzzle in which hidden pairs (Shaded) are found.

![Figure 2.7 Sample Sudoku Puzzle that applied HP](image)

---

21
Hidden pairs are formed when a pair of cells contains at two same possible digits and there are no other unsolved cells in the same row or column containing those pairs of digits. Cell (2, C) and Cell (4, C) are a typical sample of hidden pairs. In this case, the digit involved should be digits 2 and 4. Since digits 2 and 4 can only be accommodated either in Cell (2, C) or Cell (4, C). Therefore, other digits of these two cells, i.e., digits 1 and 3 in Cell (2, C) should be eliminated. Table 2.6 shows the deductive argument of this concept.

<table>
<thead>
<tr>
<th>Effective Scope of HP</th>
<th>Deductive Arguments</th>
</tr>
</thead>
</table>
| Row                   | **Premises 1:** Each cell should accommodate a digit from 1-9  
**Premises 2:** Digits 1-9 should only appear once in each scope  
**Premises 3:** Digits 3 and 4 are only possible in cell (5, G) and cell (5, H) of Row 5  
**Conclusion:** Digit 1 and 2 can be eliminated their possibilities from cell (5, G) |
| Column                | **Premises 1:** Each cell should accommodate a digit from 1-9  
**Premises 2:** Digits 1-9 should only appear once in each scope  
**Premises 3:** Digits 2 and 4 are only possible in cell (2, C) and cell (4, C) of Column C  
**Conclusion:** Digit 1 and 3 can be eliminated their possibilities from cell (2, C) |

*Table 2.6 Deductive Arguments of HP*

After the discussion of the logical rules, existing applications are reviewed. Nowadays, there is still no existing Sudoku application, which provides deductive arguments as instructions to guide the students in the process of solving Sudoku. Some of the applications even solve the Sudoku puzzle using the brute force algorithm in which there is no relation with logic. In the next section, only application provides tutoring tools are discussed.

### 2.4 Commentaries of Existing Applications

Because of the popularity of Sudoku, the number of Sudoku Applications is increasing significantly. In this section, the existing Sudoku applications, which search from Google, are reviewed.
2.4.1 Sudoku Online

Sudoku Online is a Sudoku solving application in which instructions are provided during the game. The characteristic of this application is the categories of help provided during the game. In figure 2.8, help from the system can be triggered in the menu bar. Firstly, “Possibles Easy”, “Possibles Medium” “Possibles Hard” and “Possibles Tough” determined the proportion of possible values that display in unsolved cells. Students are not allowed to edit the possible values in the cell. The systems will be automatic update the possible values if the student put a correct answer for cells. Secondly, by pressing “Hint” and “Clue” options, two types of instructions are given. “Hint” option provides the row and column index of a potential solved cell. “Clue” option stimulates student thinking by asking a question. Both “Hint” and “Clue” options in this application are only one-level. Thirdly, if “Solve” or “Solve One cell” options are selected, the answer of cell will appear in the game board interface and no interpretation is described. After the Sudoku puzzle is complete, the time used in solving the Sudoku puzzle is recorded.

![Figure 2.8 Application Interface of Sudoku Online](image-url)
2.4.2 Sudoku Hints

An application interface of Sudoku Hints is displayed in Figure 2.9. In this application, possible values of the undetermined cells are provided. Similar to Sudoku online, these values are not allowed to modify by students. Additionally, three levels of instructions are provided during the game progress. The first one is displayed when “Hint” button is pressed. It provides the box index of the potential cell to be solved. In the second level, value that can be defined in cells of the box is provided. In the third stage, the solving step is provided. It directly gives the answer of the cell. When the Sudoku puzzle is complete, the application will not record any information of game progress and student profile.

![Figure 2.9 Application Interface of Sudoku Hint](image)
2.4.3 Sudoku Tiger

Sudoku Tiger (see Figure 2.10) provides all-round functions in Sudoku. To begin with, the possibility of the cell is displayed under student request. The possible values are generated by the application. In addition, the instructions provided by Sudoku Tiger are based on the logical rules deduced from Sudoku. In figure 2.10, the technique which can apply to solve the cell is provided. For student convenience, the potential cell is highlighted. The hints request in this application is exactly same as requesting answer for the solving puzzle. It is because the answer for the highlighted cell is also given in the hints box. During the game solving procedure, some information including the time and difficulty is records. After the completeness of the game, no information can be traced out.

![Figure 2.10 Application Interface of Sudoku Tiger](image-url)
2.4.4 Conclusion drawn for existing Sudoku Applications

It is found that our focus of review is based on determining what concepts of the tutor model and student model have been applied in the existing application. Although the reviewed applications are not implemented based on the concept of ITS, the coaching strategies of existing applications can be determined.

There are lots of advantages and crucial functions that are important in the existing application. To begin with, the applications provide instructions to students during the learning progress. In the reviewed applications, the instructions are provided under request and are divided into two types. One of them is to minimize the converge area on potentially unsolved cells. Another one is to provide the direct answer of the potentially unsolved cell. For the former one, instructions are given in terms of questions such as “Where does 7 go in box one” or sentence like “Look at row 1 column 1”. The latter one provides an answer of a potentially unsolved cell in a more straightforward method, i.e. “Put digit 7 into box 1 column 1 and row 1”. In addition, the applications record basic information and progress of the games. The data stored includes time to solve a puzzle, the number of cells to be solved and the difficulty level of the game, etc.

However, the properties provided by the existing applications can be improved to achieve the creation of a better learning environment for students. There is a room for improvement in the tutoring strategies. Firstly, instruction provision cannot only be initiated by students. It is significant to implement an application which acts as sensitive as human coach. This can be achieved by providing instructions when the application senses the student is under a bottleneck or his/her performance is out of expectation. Secondly, the instructions given by existing applications are too simple. Deductive argument is suggested because it is a significant representation of rules in stimulating the logical mind of students. Thirdly, the information recorded by the existing applications is insufficient. Other variables such as the number of instruction request and solving steps of student should also be recorded for evaluating student performance.
As discussed, ITS is an important model for the development of learning application. Thereby, the structure of ITS is reviewed in the next section.

2.5 STRUCTURE OF ITS

Adopting ITS in educational application improves the performance of learning. It provides real time instructions at any instance during the progress of the game, which stimulates the student mind in training of specific ability. In this project, ITS is employed in the logic training application by solving Sudoku. Logical mind development can be achieved by applying different educational and tutoring techniques.

CAI, which is the predecessor of ITS, is considered as the coach to provide instructions for students to solve a particular problem. Figure 2.11 demonstrates the flow of instruction provision of CAI. The strategy of CAI is to provide instruction to solve the problem before a question is given. Instruction is given to rectify the concept of student until the student answer correctly.

CAI is only drilled and practical in learning application that is of simple knowledge such as mathematical operations (Conati, 2009). Additionally, the solving procedures of a problem cannot be determined. Instead, only the correctness of the final answer can be recorded by CAI (Conati, 2009). Therefore, if we would like to develop learning applications with more complex knowledge domain, a more well-defined model should be investigated.
Sleeman and Brown proposed ITS in 1982, which is an educational system that based on CAI and two additional models including expert model and student model. Tutor model of ITS substitutes the instruction provision of CAI, which focus on the relationship between instructions and learning performance. Expert Model is the embedded knowledge domain of the application which solves the Sudoku with predefined strategies. Student model is the database of student profile and performance. Additionally, it evaluates the information and affects the instruction provision strategy of the tutor model.

With the above basis, ITS should provide tailor-made instructions for students and be capable of solving all the problems instead of those pre-defined problems. Additionally, the instructions provided should base on the student pervious or even instant performance. In the next section, the configurations of the expert model are described.

2.6 CONFIGURATIONS OF EXPERT MODEL IN ITS

Expert model, also called knowledge based system, is responsible for the decision making in the knowledge domain. Figure 2.12 (Yu, 2007) shows the significant components of the expert model. They are the knowledge base and inference engine.

![Figure 2.12 Components of Expert System](image)
Knowledge base contains heuristics, which construct the expert system. It consists of rules to facilitate the decision making process. Inference engine facilitates the conclusion formation process. It draws the conclusion depends on facts and heuristics in the knowledge base.

In this section, knowledge based construction procedures are reviewed.

### 2.6.1 Knowledge Based Construction Procedures

It is undeniable that knowledge base construction is a complex task in implementing the expert system. The difficulties do not only in the conversion from heuristics to codes, but also in the selection and collection of accurate knowledge. Figure 2.13 (Yu, 2007) demonstrates the procedures for the formation of the knowledge base.

![Figure 2.13 Procedures of construction in knowledge base](image)

The first step for the construction of the knowledge base is the knowledge acquisition. Nowadays, knowledge from different aspects can be easily found and collected through the web searching. The main objective of this step is to analyze and extract the information which is the concern of the project. Additionally, the correctness and accuracy of the information are crucial. It is a must to ensure the knowledge selected is from expertise.
The second step is the representation of knowledge. The representations of knowledge are summarized into heuristics, which finally implanted into the knowledge base of the expert system. There are several ways in representing knowledge such as the frame based representation and rule based representation. These knowledge representation strategies are the easiest and most straightforward.

The last step is knowledge utilization that only the selected knowledge is defined in this step. This knowledge is applied and as the basis for the inference engine. Since knowledge acquisition and knowledge utilization methods are based on different knowledge domain. In the next subsection, knowledge representation methods are described.

2.6.2 Knowledge Representation

As discussed, there are several ways to represent knowledge. In this section, only frame based representation and rule based representation are discussed since they are the most straightforward.

Based on the observation of Ingizio (1991), knowledge is divided into two general categories. They are prior knowledge and inferred knowledge. Prior knowledge is said to be “the facts and rules that are known about a specific domain prior to any consultation session with the expert system” (Ingizio, 1991) while inferred knowledge is considered as “the facts and rules concerning a specific case that are derived during, and at the conclusion of, a consultation with the expert system” (Ingizio, 1991). These two types of knowledge are applied in the expert system to satisfy its purpose. In other words, expert system applies the inferred knowledge are derived from prior knowledge. In Sudoku, prior knowledge refers to the logical rules while inferred knowledge is the steps matrix of the expert model.

To begin with, frames based knowledge is reviewed. It is a simple structure for the object representation in a typical situation. Slots and fillers are utilized in defining an
object. Slot refers to the data of an object while the filler are the values or procedures of slot. Table 2.7 (Yu, 2007) illustrates the frame based representation of a book.

<table>
<thead>
<tr>
<th>Slots</th>
<th>Fillers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Expert System</td>
</tr>
<tr>
<td>Authors</td>
<td>Joseph C. and Gary D.</td>
</tr>
<tr>
<td>Edition</td>
<td>Fourth Edition</td>
</tr>
<tr>
<td>Publisher</td>
<td>Course Technology</td>
</tr>
</tbody>
</table>

*Table 2.7 Frame Based Representation of a Book*

Based on the observations of Giarratano J. and Riley G. (2005), frame based knowledge are practical in presenting causal and generic knowledge since their information is arranged by object attributes, cause and effect. Therefore, rule based representation is better in logic representation.

Rule based representation is commonly used for representing logical knowledge in the expert system. As it is representing in terms of conditional ways, it would be more closely related to logic. Therefore, it is more straightforward and understandable. As discussed, since there are many logical rules deduced from Sudoku, rule based representation should be more suitable than the frame based representation in this application. Apart from a well defined expert model, tutor model is also significant criteria for the successful in an ITS.

### 2.7 Configurations of Tutor Model in ITS

Tutor Model is to correct the mismatch between student knowledge and expert knowledge. A good tutor model should imitate the human coach. However, a number of researchers claimed that ITS is incapable of modulating all the possible ways to solve the problem and the “canned” text instructions cannot provide the same sensitivity as the human tutor (Jonassen, 2000). In this section, the tutoring technique for human coach in a specific domain is reviewed.
2.7.1 Reviews on tutoring techniques

Since ITS proposes a tutor model especially for instructing students in knowledge learning progress. Ideal case of tutor model acts as sensitive and initiative as human coach. Therefore, tutoring techniques for human instructors are reviewed. There are three basic techniques that are important in teaching including explaining the subject matter, asking questions, listening to students. Explaining the subject matter is a significant part of teaching. It provides the general knowledge for the domain. In this case, Sudoku game constraints and logical rules are the general knowledge provided for problem solving. Diagrams and examples can be used in explaining the problem in a more effective way.

Asking questions is another crucial strategy. It is a way to stimulate the student self-thinking. In the proposed application, the functions of asking questions are accomplished by the instruction provision during the game. Additionally, the questions and instructions given should be in different levels, which finally lead the student into the in-depth knowledge of Sudoku.

The last strategy that can be applied in teaching is listening to students’ response. However, the time waited to provide instructions to students is a complicated measurement. A quick instruction provision may mistakenly measure the ability of students. On the other hands, if an instruction given is slow, student may get upset.

2.7.2 Emotional Considerations in Tutoring

According to the conference by Conati(2009), emotion of students should also take into account in teaching. However, there is still no existing research on the comprehensive emotional measurement. Conati(2009) proposed a questionnaire that can serve the purpose. A sample extracted from Conati(2009) is shown in figure 2.14.
However, the information recorded in the questionnaire cannot provide a continuous emotional measurement during the game progress. Compare with the expert model, development of the tutor model of ITS is less considered. Student model, which records and evaluates student performance, develops in an even slower pace. The configurations of the student model in ITS are described in next section.

2.8 Configurations of Student Model in ITS

Student Model looks upon as the student performance keeper and evaluator. Student’s performance such as the strategies applied is stored in the student model diagnosis. There are several techniques used in evaluating student performance and his/her learning ability. However, Liu and Wu (2009) claimed that there was still no universal framework to explain how the student model applied in ITS. In this section, knowledge learning representation and measurement of the learning progress are described.
2.8.1 Knowledge Learning Representation

There are three typical models to analyze the learning progress of student. They are overlay model, differential model and perturbation model. Among three of them, overlay model is most common in the student model.

To begin with, overlay model is described. This model aims to demonstrate the unstudied knowledge of students. Another common model is the differential model. The objective of this model is to illustrate the knowledge explosion of students. However, the overlay model and differential model do not state the incorrect knowledge that students applied. Perturbation model, which shows the incorrect strategies applied in solving the problem, is described. The student knowledge out of the expert knowledge represents the incorrect strategies or mal rules. Apart from the knowledge learning measurement, the measurement of cognitive state during the game progress is discussed in the next subsection.

2.8.2 Cognitive State of Student

Cognitive state of student is difficult to measure. The discussion in this section is based on the research done by Liu, Tang and Li (2008). Table 2.6 demonstrates the abilities that are important in understanding the cognitive state of students.

<table>
<thead>
<tr>
<th>Abilities</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Induction</td>
<td>Ability of understanding from examples and general knowledge</td>
</tr>
<tr>
<td>Memorization</td>
<td>Ability of recalling the past memories such as experiences and concept</td>
</tr>
<tr>
<td>Discovering</td>
<td>Ability of gaining knowledge that is new to students based on the knowledge in which tutor has been taught</td>
</tr>
<tr>
<td>Differentiation</td>
<td>Ability of differentiate different concepts or rules of knowledge</td>
</tr>
<tr>
<td>Analogizing</td>
<td>Ability of deterring the relationship between different objects</td>
</tr>
</tbody>
</table>

*Table 2.8 Abilities in measuring cognitive state*

It is suggested that the understanding of a knowledge domain of a student is determined by the above abilities.
In the recent development of ITS, student model of ITS is a hot topic for researching. However, until now, there are still no concrete and comprehensive studies on the student model, especially in the relationship between the records stored in the student model and instruction provision in the tutor model.

After the reviews of the expert model, tutor model and student model in the previous section, some recent development of ITS is discussed.

2.9 RECENT DEVELOPMENT OF ITS

ITS surpasses other educational systems such as Computer-Aided Learning (CAL), Computer-Aided Instruction (CAI), Computer-Based Training (CBT) and Web-Based Training (WBT) etc. The success of the ITS is because of its dynamicity and adaptability towards the learning of individuals (Nwaocha, N., 2009). Until now, researches in ITS is still a hot topic and submitted in the international ITS conference. In this section, two recent researches are discussed. The first one is the mixed initiative Sudoku in Its which the other one is the new challenges and directions of ITS.

2.9.1 Mixed-Initiative Sudoku ITS

Caine and Cohen (2007) proposed a mixed initiative ITS in 2007 for solving Sudoku puzzle. For mixed initiative system, interaction between players and system is significantly important. The players take the initiative to solve the Sudoku puzzle. The system only takes the initiative for an instructor whenever players make any mistake. In the system, the teaching consultant calculates and evaluates the most effective strategy for instructions with accordance to the individuals (Caine A. & Cohen R., 2007).

For the mixed initiative Sudoku ITS system, the tutor model has met the criteria that the ITS should dynamic and adaptive to each individual. However, the tutor model in this project cannot act as actively and sensitively as the human tutor. In other words,
it provides suitable instructions to players but is not with suitable time. Tutor model should automate to sense the difficulty that student faced and give appropriate instructions.

Nowadays, for a successfully intelligent tutoring system, importance in evaluating the student performance and ways to provide the most suitable instructions for students outweigh the level of intelligence of the system.

2.9.2 New Challenges and Directions of ITS

In the 21st International Joint Conference in Artificial Intelligence 2009, Conati proposed the importance of student emotion be the new directions of the tutor model in future ITS. Intelligent tutoring system should be more human to handle the emotion of student. For example, ITS considers what to do if the student is feeling upset and how to improve the learning if they are with positive energy. In addition, she also states the usage of example during learning as the proportion of example in learning may affect the ability of student to solve the problem (University of British Columbia, 2009).

From the reviewed of the recent development of ITS, it is simply to conclude the difficulties in the implementation of the ITS and it will be discussed in the next section.

2.10 Difficulties of ITS

After reviewing some recent development of ITS, it is not difficult to find out the main problem in developing a successful ITS. They are the implementation and design of the tutor model and student model in facilitating the learning. The challenges and difficulties are summarized in this section.
2.10.1 Performing Human Tutoring System

It is a complicated job to construct a tutor model with the same sensitively of a human instructor. Strategy, timing, benefit and emotion are the most important components to in constructing the tutor model to be the human coach. To formulate these components is not an easy task. This project achieves three quarters of components including strategy, timing and benefit that student gained when an instruction is given.

2.10.2 Measuring Student Performance

The measurement of student performance for particular individuals is difficult. Most of the paper has suggested that the measurement is formulated and calculated in terms of the strategy which students applied and strategy can be applied to solve the problem. However, if there are list of strategies is available for players, it is complicated for the system alone to decide which strategy that players actually use. Furthermore, student normally applied the strategy without knowing them. Therefore, there should be some guessing or variations in the calculation. As a result, the calculated performance should be 100% correct.

This chapter has reviewed a number of existing researches and studies. The high level requirements that are suitable for the proposed project will be described in the chapter three.
CHAPTER THREE
CONCEPTAL MODEL OF ITS IN SUDOKU

After reviewing what Sudoku, Logic and Intelligent Tutoring System is, the conceptual model of the entire project is discussed in this chapter. The functional requirements are proposed specifically in terms of Sudoku application and logical developing operations in this chapter. Additionally, the conceptual representation applied in the application to facilitate logic training and functions of ITS is also described.

3.1 OVERALL CONCEPTUAL DESIGN OF THE APPLICATION

The conceptual communication of different components for web based ITS in Sudoku is shown in figure 3.1.

![Figure 3.1 Overall Architecture of Web Based ITS in Sudoku](image)

The three basic modulus including expert model, tutor model and student model communicate and interact with each other base on the ITS concept. These models are the embedded system, and they communicate with the students through the
application interface. Any significant information and data such as student profile and his/her solving steps are stored in the database whenever a game is finished. More detailed information of this overall conceptual schema is discussed in the following section.

3.2 REQUIREMENTS OF APPLICATION INTERFACE

Sudoku application interface facilitates the interactions between students and the embedded ITS. Specially, it triggers and provides instructing information from the tutor model. Requirements of the application interface are based on the information supplementary about the instructing strategies and instruction provision during game progress. Figure 3.2 demonstrates the conceptual interface design of the application. In this figure, students are capable of performing three functions in the application interface.

To begin with, the system allows the student or his/her coach to set the priority of rules for the Sudoku puzzle solving procedure. This raises the flexibility of the system since the strategies applied on each step is based on the student preferences. In other words, expert model is capable of solving the Sudoku puzzle in various ways. Meanwhile, the system provides more understandable instructions for students because the application and the student are using the same logic set.
In addition, instructions in terms of logical argument are given whenever student’s request or student performance is out of expectation during the solving progress. Given instructions do not only solve the bottleneck for students in solving the puzzle, but also stimulate and develop the logical mind of students. In this system, students are allowed to enable or disable the automatic instruction provision function. The time for given an automated instruction depends on student previous performance. This function makes the tutoring procedures as initiative as the human coach. All the instructions provided are displayed in the instruction’s box.

3.3 Conceptual Representation of Expert Model

Conceptually, expert model should be capable of solving all practical Sudoku puzzles. Additionally, it should apply the strategies with similar direction of student in solving Sudoku puzzles. This can be achieved by setting the priority of rules before the game gets started. The priority of rule defaults in terms of ascending order of its difficulty. Expert model uses the priority set of rules to solve the Sudoku puzzles to perform similar strategies and directions as what students do in reality. To achieve the overall purpose of the expert model, conceptual representation of the expert model is required and is described in this section. Some fundamental terms in expert model is defined.

3.3.1 Definitions of fundamental terms in Expert Model

There are three main components for a typical expert model. They are strategy graph, steps matrix and skills matrix. In this subsection, the functions of each component are described.

Firstly, strategy graph is used to figure out the possibility of a group of potential answers. For example, in an English language learning system, there are four choices provided for students to select the correct answer. In this case, strategy graph will store each of the choices and eliminate the choice one by one according to the decision from the expert model.
Secondly, steps matrix is used to store all the procedures from that instant to completion of the problem. A simple example can be extracted from a mathematical calculation such as “(1+2) X 3/9 =?” Suppose the student has already finished the first steps in calculating 1+2 = 3. At this moment, the steps matrix generated will be the other two procedures which are (1+2) X 3 = 9 and (1+2) X 3/9 =1.

Thirdly, skills matrix records the skill or skills to solve a particular problem. To cite a simple example, a mathematical proof can be solved through various methods such as proof by contradiction or mathematical induction. Methods that can be applied to solve this mathematical proof are recorded in a skills matrix.

After the description of the important terms in the expert model, the strategy graph, steps matrix and skills matrix applied in Sudoku is discussed in the following subsection (Sub-section 3.3.2, 3.3.3 & 3.3.4).

### 3.3.2 Sudoku Strategy Graph

Expert model is initialized by the Sudoku Strategy Graph (SSG) once students start the application. The important clues (given cells) are from the database and are defined in the SSG. SSG is applied for the representation of instant Sudoku solving state. Figure 3.3 shows a typical Sudoku Strategy Graph for a semi-solved Sudoku puzzle. Sudoku Strategy Graph is straightforward and simple to understand.
In the expert model, every cell is divided into nine parts and each part represents the digit from one to nine respectively. Each possible value within a cell is deleted either the value is determined within the cell of its same scope (column, row and box) or the value is found to be impossible by different logical rules. Within the system, the SSG does not only responsible for recording the instant game progress of the student, but also providing information of the instant game states. Expert model makes use of the SSG solving the Sudoku puzzle in various ways based on the rules’ priority according to student preferences. As discussed, the possible ways of solving a cell is determined by the Sudoku Steps Matrix, which described in the next section.

### 3.3.3 Sudoku Steps Matrix

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.3.</td>
<td>.4.</td>
<td>.6.</td>
<td>.7.</td>
<td>3</td>
<td>9</td>
<td>.2.</td>
<td>.4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2.3.</td>
<td>.4.</td>
<td>.6.</td>
<td>.7.</td>
<td>1</td>
<td>2.3.</td>
<td>.4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>2.3.</td>
<td>.4.</td>
<td>.6.</td>
<td>.8.</td>
<td>7</td>
<td>2.3.</td>
<td>.4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>1.6.</td>
<td>.4.</td>
<td>.8.</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>1.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.2.</td>
<td>1.3.</td>
<td>.4.</td>
<td>.5.</td>
<td>.6.</td>
<td>1.2.</td>
<td>7</td>
<td>1.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1.2.</td>
<td>4.5.</td>
<td>.6.</td>
<td>.7.</td>
<td>8</td>
<td>2.3.</td>
<td>4.5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>1.2.</td>
<td>4.5.</td>
<td>.6.</td>
<td>.7.</td>
<td>3</td>
<td>4.3.</td>
<td>4.5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>2.3.</td>
<td>.4.</td>
<td>.5.</td>
<td>.6.</td>
<td>2.3.</td>
<td>4.5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>2.3.</td>
<td>.4.</td>
<td>.5.</td>
<td>.6.</td>
<td>3</td>
<td>2.3.</td>
<td>4.5.</td>
</tr>
</tbody>
</table>

*Figure 3.3 A typical Semi-Solved Sudoku Conceptual Representation in Expert Model*
Sudoku Steps Matrix aims to record the sequence of steps that follow to complete the Sudoku puzzle. Sudoku steps are stored in two dimensions. One is breadth steps while the other is depth steps. Breadth steps refer to the steps that students apply at the instant. Depth steps refer to the sequence of steps that are applied in order to solve the puzzle based on the rules’ priority of students’ preference. Apart from the Sudoku steps matrix, skills matrix is also applied in ITS for Sudoku.

### 3.3.4 Sudoku Skills Matrix

Expert model should draw up the Sudoku Skill Matrix (SSM) to determine the skills can be used to solve a cell. A typical Sudoku skills matrix is shown in Figure 3.4.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1,3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1,2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1,2</td>
<td>3</td>
<td>1,2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1,3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1,2</td>
<td>5,6</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>2,4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1,3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2,3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1,2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*Figure 3.4 A typical example of Sudoku Skills Matrix*

Zero in the cell indicates that the cells cannot be solved instantly. Number from one to six determines the index of logical rules can be applied to solve the particular cell.

After the description of the major components in the expert model, another significant model, which is the tutor model, is discussed.
3.4 Conceptual Representation of Tutor Model

Undoubtedly, tutor model is the communication bridge between the expert model and student. Expert model performs checking on the accuracy of student input whenever students declare the cell value of the Sudoku puzzle. Tutor model performs its functionalities whenever students make any mistake or his/her performance it out of expectation. In this section, the requirements of the tutor model are described.

3.4.1 Descriptions of functions in the tutor model

The most important objective of a tutor model is performing like a human coach. There are three main functions of a tutor model including the instruction provision to students, initiation and sensitivity of human coach and explanation of the generated answer. In this subsection, the basic ideas of these functions are described.

To begin with, instruction provision to students is described. Instructions refer to the hints that guide the student to draw the conclusion by themselves instead of providing answers. For example, in a mathematical calculation “1+2 X 3 =?” a sample instructions provided can be similar to “Perform multiplication before addition”.

Additionally, it is important for a tutor model acts as initiative and sensitive as a human coach. Initiative indicates that the tutor model should be capable of providing instruction to students automatically. Sensitive means the tutor model should provide instructions to the student when the model senses the student is under a bottleneck. Therefore, tutor model can be as initiative and sensitive as a human tutor if it gives automatic instructions when it senses that the student faces a bottleneck.
Moreover, explanation of problems that students cannot be solved on their own is considered as a significant learning procedure. The explanation of the problem should be steps in steps and guide the student to the completion of the problem. For example, in a mathematical calculation “1+2 \times 3 = ?” The explanation is similar to the followings:

“Firstly, perform multiplication before addition”
“Secondly, perform multiplication 2 \times 3 = 6”
“Thirdly, perform addition 1+2 \times 3 = 1 + 6 = 7”
“Therefore, the final answer should be 7”

After the descriptions of the functions in the tutor model, we found that the functions are actually the techniques described in section 2.7.1 that could be applied in tutoring a student. The functions applied for tutor model in Sudoku are described in next subsections.

### 3.4.2 Applying teaching strategies in the tutor model

In the proposed application, information and logic rules of Sudoku are first explained in the web page. Examples are used for the interpretation of logical rules. In this way, a basic concept of logic rules and it application is provided for students.

After that, the instruction is provided during the puzzle solving procedures. Instructions are provided in three levels. They are stated as follows:

**Level 1: Row Hints**

It is the first level of the instructions provided to inform students that the row of a potential cell that can be solved.

**Level 2: Column Hints**

Column Hints is the second level of the instruction given to tell the student about column row of a potential cell that can be solved.

**Level 3: Rule Hints**

Rule Hints suggested that the potential logical rules that can be applied in solving the Sudoku puzzle
If the student cannot solve a particular cell after the provision of instructions, tutor model explains the procedures to solve the cell using premises and collusion, which is an effective and efficient way to stimulate and develop the logical mind of student. In the proposed project, we considered this explanation as the level 4 instructions and stated as follows:

Level 4: Answer Hints

Request answer hints indicate that the system will solve the next steps for students. The instructions given are in terms of premises and conclusion to teach the student to solve a specific cell step by step.

Last but not the least; the proposed tutor model is as sensitive and initiative as a human coach. Therefore, the tutor model provides automatic instructions if students do not make any progress after a predefined time. In this case appropriate waiting time is required to determine whether students understand the teaching material. A long waiting time may make students upset while a short waiting time cannot measure the actual understanding of students. In this project, an automatic instruction provision function is established. The waiting time for instructions is adjusted based on the previous performance of students.

To conclude, in the proposed tutor system, instructions together with premises and conclusion are given for the interpretation of a particular scenario that students would like to overcome. By continuous interpretation of logic rules in different scenarios, students understand how to apply logical rules in different aspects effectively.

After discussing the ways to instruct students in logic training, the learning of the student should be recorded for further analysis and evaluation, which is a significant objective of the student model.
3.5 Conceptual Representation of Student Model

Student model stores the student performance and evaluates the development progress of logical mind of students. The information recorded in the database is useful in evaluation. Below described the information required for the evaluation process.

3.5.1 Recorded Information

There are several categories of information that would be recorded during the game. The following information is recorded in the database and is used for evaluation.

To begin with, number of hints requested by students indicates how much knowledge the system has been taught the student. It also determines the understanding and ability of students to apply different logic rules. In other words, number of hints requested by students is proportional to how students are familiar with the logical rules. If students are proficient in apply logical rules, they are capable of solving the Sudoku puzzle without any assistance.

In addition, time consumed to solve the puzzle provides evidence on the strength of student in logic thinking. Obviously, it also provides information on how students familiar with the logic rules.

Moreover, number of wrong declaration of cell value gives the information of misconception that the student made in solving the Sudoku puzzle. It can be used to trace out the scenario that the student most likely to make mistakes. It helps to guide the student to correct the mal-rules developed in his/ her mind.

Furthermore, percentage of the student applied each rule successfully determines which rule the student is most familiar with. It also provides information about the learning progress of student.
All the information is stored and applied together in the evaluation progress to formulate the most suitable evaluation method for measuring student logic developing process. The method in evaluating student learning is discussed in the next subsection.

### 3.5.2 Evaluation Strategy

As discussed, information recorded is for the evaluation of student performance. In this project, three types of graphs are proposed to show the student performance.

The first one is the performance chart. The objective of the performance chart is to calculate the percentage that student can apply a specific rule successfully. Expert knowledge is assumed to be one hundred percents. This graph illustrates the percentage of knowledge of student. The graph actually shows the differences between acquired knowledge and expert knowledge, which is the concept of the overlay model.

The second one is the hints usage chart. It is applied to check the usage of different instructions. If all the cells are solved with the use of answer hints, the percentage of the hints usage would results in one hundred percents.

The last one is the combination of the above two graph together with the misconception parameter. In Sudoku, there may be some mal-rules that would be applied to solve the puzzle correctly only in some scenario. Meanwhile, some of the mal rules violate the concept of the proposed logic rules. Therefore, misconceptions that student made should be recorded. Instructions, knowledge of logical rules and misconceptions curves in the same graph provides the performance of students in a particular game.

After the description of the conceptual representation of the ITS model, the knowledge representation of logic rules is defined in the next section.
3.6 Knowledge Representation for Expert Model

According to above descriptions, expert model is of advanced knowledge in Sudoku and is capable of solving all practical Sudoku puzzles. Expert model makes use of the given cells as the basic clues and tries to apply each logic rule in every unsolved cell to solve the Sudoku puzzle completely. Rule based expert model is the simplest and the most suitable way in implementing the expert model of this system. In this section, rule based knowledge representation is adopted to construct the knowledge of logical rules in the expert model.

3.6.1 Fundamental Facts on Rule Based Knowledge Representation

As discussed, rule based knowledge representation is adopted in presenting the logic rules in the expert model. To begin with, it is essential to understand the fundamental terms of Sudoku Puzzles. Table 3.1 explains the terms that are used in the knowledge based representation of Sudoku rules.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopes</td>
<td>Scope refers to the column, row and box in the Sudoku Game Board</td>
</tr>
<tr>
<td>Possible Values</td>
<td>Only digit from one to nine is the possible values that can be put in the Sudoku cell</td>
</tr>
<tr>
<td>Definite Cell Value</td>
<td>Every cell should have a defined value in which it is unique within its scopes</td>
</tr>
<tr>
<td>Given Cell Value</td>
<td>Some of the cell values are given initially as the clues for solving the Sudoku Puzzle</td>
</tr>
</tbody>
</table>

Table 3.1 Fundamental Terms of Sudoku Puzzles

In addition, it is significant to understand the rule based knowledge representation method, which is the simplest and the most worldwide approach. Knowledge of logic rule is expressed in terms of condition and consequence where both condition and consequences are expressions. Its syntactical form is shown below:

\[
\text{if (Condition) then (Consequences) ;}
\]
3.6.2 Rules Based Representation of Logical Rules

Logic rules are divided into two categories, including declaration rules and elimination rules. Declaration rules such as Single In Group, Naked Single and Hidden Single aims to declare the definite values of a particular cell. Elimination Rules compute the impossible values for the cells, including Intersection Reduction, Hidden Pairs and Naked Pairs. Table 3.2 illustrates the rules based knowledge representation of each of the logic rules. The table also provides a simple description of the logic rules.
<table>
<thead>
<tr>
<th>Simple Descriptions</th>
<th>Rule Based Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single in Group</strong></td>
<td>If there are already eight definite values in a scope, the remaining cell should contain the last value</td>
</tr>
<tr>
<td><strong>Naked Single</strong></td>
<td>If the cell can only accommodate one possible value, the value should be assigned to that cell</td>
</tr>
<tr>
<td><strong>Hidden Single</strong></td>
<td>If only one cell are capable to contain a value within the scope, the value should be assigned to that cell</td>
</tr>
<tr>
<td><strong>Intersection Reduction</strong></td>
<td>If a value can only be accommodated by one row or column within a box, value from adjacent boxes of that box can be eliminated</td>
</tr>
<tr>
<td><strong>Naked Pairs</strong></td>
<td>If two values are possible by two cell within the same scope, the two values of other cells in the same scope can be eliminated</td>
</tr>
<tr>
<td><strong>Hidden Pairs</strong></td>
<td>If two values are possible exactly in two cells AND at least one of the cells containing other possible values, then (the other values can be eliminated from that cell)</td>
</tr>
</tbody>
</table>

**Table 3.2 Rule Based Knowledge Representation of Logical Rules**
The requirements of the ITS model has been discussed in detail in this chapter. In the next chapter, the design based on the requirements and conceptual model is described.
CHAPTER FOUR

DESING OF THE WEB BASED ITS IN SUDOKU

After the discussion of the conceptual model of web based ITS in Sudoku, the overall design is discussed in this chapter. The design of this chapter is based on the requirements discussed in the previous chapter.

4.1 DESIGN OF APPLICATION INTERFACE

Application interface is an important component of an application. A user-friendly interface facilitates the purpose of the application and raises the interest of the student. In this application, design application interface does not only provide a platform for students to work on Sudoku puzzle, but also implement the Sudoku puzzle solving in an interesting way.

4.1.1 Overview the Design of Application Interface

The application interface of the proposed application facilitates the functions that are provided for logic training such as instruction request and rules priority setting, etc. Figure 4.1 shows the interface of the application. It consists of five main parts. They are the position indicator, game grid, game information and settings panel, drag and drop input and instructions & solving procedure box.
To begin with, position indicator is to determine the position of a cell by the two dimension position indicator. Additionally, the drag and drop input is provided for assign or erase the value of cells. This drag and drop input bar makes the input interesting and straightforward. The other three parts, including the game grid, game information & setting panel and instructions & solving procedure box, are detailed discussed in the later subsections (Section 4.1.2, 4.1.3, 4.1.4 & 4.1.5).
4.1.2 Design of the Game Grid

The game grid provides 9 x 9 cells for students to complete the Sudoku puzzle. Cells are considered as the crucial components in the construction of game grid. There are altogether four types of cells inside the grid (see Figure 4.2).

![Figure 4.2 Enlarged Image of part of the game grid](image)

To begin with, given cells are the cells with predefined value, which are the important clues for solving the Sudoku puzzle. For user convenience, the given cells are red and have a pink background. Students are forbidden in selecting these cells as the value of given cells are assumed to be correct.

In addition, solved cells are divided into two categories based on the objects that solved the cell. If the cells are solved by the system and are assumed to be correct, it is shown in grey in the application interface. Similar to given cells, students are prohibited from selecting these cells. For cell solved by students, it is shown in black in the grid. Since these cell values are determined by the student, it may be wrong. Therefore, “Erase” drag and drop button is provided to cancel any incorrect value of cell. However, for student convenience, a warning is prompt out if the student mistakenly tends to erase the value of a correct cell (see Figure 4.3). In other words, incorrect cells are allowed to be erased. Additionally, students allow assigning
another value of the cell before erase the incorrect value from it. Otherwise, a warning is prompt out (see Figure 4.3).

![Figure 4.3 Warnings for handling apparent errors from students](image)

Furthermore, the last type of cell to be discussed is the unsolved cell. Since the application provides possible values in every cell for students to eliminate the values by elimination rules, possible values determined by students are shown in unsolved cell. The possible values of the cell can be eliminated by simply right click on it. Meanwhile, the impossible values of the cell can be recovered by right click it again on the specific location.

The main purpose of the game grid is to provide a game board for students to work on. However, if there is any characteristic such as the difficulty of Sudoku is required to change, game information and settings panel, which described in the next subsection, provides functions to support such changes.

### 4.1.3 Game Information and Settings Panel

Game information and settings panel support the pre-game settings and in-game settings. Additionally, it provides information of the game states. Figure 4.4 demonstrates the interface of the game information and settings panel.
To begin with, difficulty and rules priority settings are considered as pre-game settings. These two settings are only allowed to set before the game gets started. It is because these two settings are required for the expert model to select the most suitable Sudoku puzzle for students and solve the Sudoku puzzle according to student preferences. Difficulty spinner supports three levels of difficulty of Sudoku puzzle; they are “gentle”, “moderate” and “difficulty”. On the other hands, rule
priority spinner allowed the students to change the priority of the rules. Meanwhile, the uniqueness of the rules in different priority is required to verify.

Following the pre-game settings, instant states of Sudoku puzzle are shown. It does not only display the time used in solving the Sudoku, but also display the remaining cells to be solved. Additionally, it provides the information of the instructions used by students. Moreover, immediate feedback is also provided. If the students make a mistake in solving the game, the game states will change to “something goes wrong” to inform students the incorrectness of strategy applied.

Last but not the least, an in-game setting allows enabling or disabling the automatic instruction provision function. It facilitates the tutoring of application similar to the human coach. The instructions are displayed in instructions and solving procedure box.

4.1.4 Instructions and Solving Procedure Box

Figure 4.5 shows the information displayed in the Instructions and Solving Procedure Box.

![Figure 4.5 Detailed Descriptions of Instructions and Solving Procedure Box](image)

Procedures and steps applied by students are shown in the box. There are two types of steps that are required to record. One of them is the declaration of value while
the other one is the elimination of possible value. This information provided by the applications enables students to trace back their step when they have done something wrong in the middle. In addition, instructions of different levels are also demonstrated in Figure 4.5. It shows two different levels of instructions including the rules hints and answer hints. The answer hint provides step by step premises of specific logical rules and finally drawn the conclusion of it and solved a cell.

The application interface facilitates the work of the expert model. It gets the logic set (rules priority) of student and sends to expert model for solving the Sudoku puzzle. In the next section, design of the expert model is explained.

4.2 Design of Expert Model

Expert model is capable of solving the input Sudoku puzzle from the database by applying similar logic set of students. Figure 4.6 describes the design class components of the expert model which facilitates the operations of the expert model. The overall class diagram is demonstrated in Appendix B.
4.2.1 Control Mechanism of Expert Model

The overall control mechanism has been shown in Figure 4.7. The expert model is capable of solving the puzzle and generating the steps matrix and skills matrix. In the figure, we find that the expert model can determine the difficult level of the Sudoku puzzle stored in the database. It will base on the student preference in the provision of Sudoku puzzle.

To begin with, the setup of the expert model is done once the student press the “Start button” in the application interface. There are two important components should be defined in the expert model. One of them is the logical rule priority and the other one is the Sudoku puzzle.
Expert model first searches the database and finds out the Sudoku puzzle with students defined difficulty. Meanwhile, expert model gets the rules priority set according to the preference of student. Afterwards, expert model set up the SSG based on the Sudoku puzzle selected before. The SSG keeps recording in the expert model together with other parameters such as effective scopes to provide more information for the SSG. A sample of the recording information in the expert model is shown in Table 4.1.

<table>
<thead>
<tr>
<th>X coordinate</th>
<th>Y coordinate</th>
<th>Possible values</th>
<th>Logical rules</th>
<th>Effective Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1,2,5,6,7,9</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>Single In Group</td>
<td>column</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>1,2,3,5,6,8</td>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>

Table 4.1 Sample storage of strategy graph

After the setup of the expert model, the expert model applies the logical rules one by one in the SSG based on the order of rule priority. This process is to ensure the accuracy of the Sudoku puzzle and make sure the selected puzzle is solvable by the proposed application. Then, the first Sudoku Steps Matrix is generated and recorded in the database (see Figure 4.8).

<table>
<thead>
<tr>
<th>UserID</th>
<th>PuzzleID</th>
<th>StepID</th>
<th>DepthSteps</th>
<th>BreadthSteps</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>42-3:Hidden Single</td>
<td>1</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>52-3:Hidden Single</td>
<td>2</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>8</td>
<td>72-3:Hidden Single</td>
<td>3</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>6</td>
<td>18-9:Hidden Single</td>
<td>4</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>7</td>
<td>36-3:Hidden Single</td>
<td>5</td>
<td>null</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>null</td>
<td>201000000006</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>44-7:Hidden Single</td>
<td>6</td>
<td>null</td>
</tr>
</tbody>
</table>

Figure 4.8 A List of Sudoku Steps Matrix

In the figure, one row represents a Sudoku Steps Matrix. As discussed, Sudoku steps matrix is divided into two types, depth matrix and breadth matrix. After the
generation of the Sudoku steps matrix, the breadth matrix sends to tutor model. Tutor model then extracts the steps out based on the order of rules priority and uses as an instruction for students.

Apart from the Sudoku Steps Matrix, Sudoku skill matrix is also updated. Sudoku steps matrix is recorded after every step of students. However, Sudoku skills matrix is only recorded in the database after the Sudoku puzzle is complete. Each row of record in Figure 4.9 determines the skills that could be applied in each grid of the Sudoku puzzle.

<table>
<thead>
<tr>
<th>UserID</th>
<th>PuzzleID</th>
<th>GridID</th>
<th>Singleton</th>
<th>NakedSingle</th>
<th>HiddenSingle</th>
<th>IntersectionSolutions</th>
<th>NakedPairs</th>
<th>HiddenPairs</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>37</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>46</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>66</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>74</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*Figure 4.9 Sudoku Skills Matrix of the application*

After discussing the overall control mechanism of the expert model, the mechanism in providing the decision making ability is discussed in the next subsection.

### 4.2.2 Solving Sudoku Ability of Expert Model

This subsection focuses on the Sudoku solving ability of the expert model. In other words, it describes how we can find out the procedures and skills of the puzzle solving for steps matrix and skills matrix. Figure 4.10 demonstrates the mechanism in achieving the decision making ability.
Chapter Four

Desing of the Web Based ITS in Sudoku

**Figure 4.10 Control Mechanism of Sudoku Puzzle Solving**
In the figure, a copied version of the instant strategy graph is used to generate the solving procedures of the Sudoku from that instant until completion. If the logical rule can be applied in a particular unsolved cell, the model adds the index of logical rule in skills matrix. If the rules can be applied in a particular cell, it indicates that it is a solving procedure. Therefore, the solving procedure will add to the steps matrix.

The algorithms in implementing the application of logical rules are discussed in detail in Chapter 5. After the discussion of design of the expert model, design of the tutor model to meet the requirements of the project is suggested.

### 4.3 DESIGN OF TUTOR MODEL

Tutor model is a significant component in facilitating the logic training of the application. Figure 4.11 demonstrates the object oriented design of the tutor model. In the figure, the main operation of the tutor model is providing instructions. In this section, the design strategy in providing instructions is discussed.

![Figure 4.11 Components of Tutor Model](image-url)
4.3.1 Operations design of tutor model

Tutor model is set up when the Sudoku puzzles get started. Tutor model should also record the priority of rules from input of student since types of instructions are displayed in the order of rule priority.

To begin with, the tutor model gets the priority of rule from the application interface. Afterwards, the tutor model extracts the waiting time for the instruction provision of a particular student from the database. Meanwhile, the tutor model sets up its time counter. The next possible steps of students, which predicts by the expert system, are sent to tutor model. Tutor model orders the possible steps based on the logical rules’ priority and applied them as instructions for students whenever the countdown waiting time of automatic instructions is zero or student press the “Hints” button from the application interface.

Instructions provided are in various forms including the row index, column index, logical rule and deductive arguments. These instructions are displayed in the application interface for students’ reference. In the next subsection, the mechanism for instruction provision is discussed.

4.3.2 Instruction Provision Mechanism

As discussed, instructions are provided under different situations. The logical step that the tutor model used for the next instructions is sent to the expert model for the interpretation of that step. The expert model sends the interpretation to the tutor model in terms of deductive arguments.

Tutor model then prepares the four different levels of instructions for the next step. A sample of different types of instructions is described in Figure 4.12.
Students can get the instructions manually and automatically. Under manual mode, students can choose the most suitable instruction for them. By evaluate the types of instructions request; it is possible to find out the weakness of students. For example, if student request for instructions based on the position of potential solved cell, he/she is difficult to trace out the potentially unsolved cells instead of having a problem in applying logical rules. On the other hand, instructions are provided in level under automatic mode. In this way, the student is guided by the system step by step and finally solves the cell.

However, the time for providing automatic instructions should base on the performance of student. The best time to provide the instructions during the game progress is discussed in the next subsection.
4.3.3 Initialization for Automatic Instruction

In this section, the design of initialization for automatic instructions is discussed. However, the strategy to formulate the time waiting to provide instruction is a complicated task, since there is no similar function was done previously. Therefore, the formula for waiting time is initialized based on the statistics of Sudoku.org.hk. The statistic is illustrated in Figure 4.13.

In this figure, only the curves with gentle, moderate and tough difficulty are considered. Consider the time band 10, only a few numbers of the students solve the Sudoku puzzle over 120 minutes. In the proposed application, 120 minutes are considered as the maximum time allowed for students to solve the puzzle. Since the different level of Sudoku puzzle is of different number range of cells to be solved. The time band and number of cells to be solved is used to formulate the waiting time for automatic instructions. The following is the proposed time for the automatic instruction provision at the initialize state.

For gentle puzzle, the Sudoku puzzle provides at least 30 given cells as clues to solve the Sudoku puzzle. The average time to solve a cell is
120 x 60 / (81 – 30) = ~140 seconds

As seen from the graph, a great proportion of people are capable of solving the Sudoku puzzle within 60 minutes.

Therefore, the first level instructions are given after 140/2 = ~70 seconds.

After that, higher level of instructions are given after every 70/4 = ~17 seconds.

This strategy ensures the student should solve the Sudoku puzzle within 120 minutes.

Similarly, for moderate and difficult Sudoku puzzle, the calculation of the waiting time is the same. Therefore, for moderate Sudoku puzzle,

The first level instructions are given after 64 seconds.

After that, higher level of instructions are given after every 64/4 = ~16 seconds.

For difficult Sudoku puzzle,

The first level instructions are given after 60 seconds.

After that, higher level of instructions are given after every 60/4 = ~15 seconds.

The waiting time calculated is only for the initial value. The waiting time is automatically adjusted after students complete every 10 Sudoku puzzle. If the percentage instructions provided is greater than 10%, the waiting time is delayed for one second. Conversely, if no automatic instructions are provided, the waiting time is scheduled one second earlier.

Although Conati(2009) has proposed that the best time to provide automatic instruction is before the student gets upset during the game, there is not still not enough information about when is the best time to provide automatic instructions. However, the proposed application cannot measure any information about emotion of students, the time calculated here is only based on the statistic which does not enough to formulate the most efficient time for automatic instructions. The initialized waiting time, number of puzzles complete and the time should be verified and adjusted after a cast number of evaluations and testing.
The operational design of the tutor model has been discussed. In the next section, the design of the student model, which is the performance keeper and evaluator, is focused.

## 4.4 Design of Student Model

Figure 4.14 shows the student model package of the overall class diagram. Since the class diagram only shows the design of the applet application of the project, information storage such as student profile is not shown in the figure. In other words, the student model functions in both the proposed web page and applet application.

![Figure 4.14 Components of Tutor Model](image)

### 4.4.1 Operations design of the student model

Operations of the student model are divided into two main categories. One is data storage while the other is the data analysis. Data recorded includes the student profile and student performance in solving the Sudoku puzzle. Student’s profile is recorded in the web page when he/she registers. The registration form is shown in Figure 4.15.
Chapter Four

Desing of the Web Based ITS in Sudoku

![Registration form of the application](image)

In the figure, simple information is captured for student profile. Age, sex, education level and Sudoku experience are the important component for further analysis in the relation of logical training. To cite an example, a great amount of data could be used to analysis the logical training in relation with different age group. However, this kind of analysis would not be conducted in the proposed project.

Apart from the student personal information is recorded through the website, his/her performance is also recorded through the application. Student model inside the application gets the steps performed by students, time required to accomplish the Sudoku puzzle, number and types of instructions used and number of misconception made from the application interface. This information is stored in the database.

As discussed in section 3.5, several graphs would be used to measure the learning progress of student. The web page illustrates the student performance by graphs. The methods in formulating the graphs are described in the next subsections.
4.4.2 Formulation of Graphs

The first one is the performance chart; this diagram is used to determine the percentage of different knowledge learning during the education. In this case, the different knowledge learning refers to different logical rules. The education refers to the Sudoku. Figure 4.16 shows an example of the performance chart. Different curves refer to the leaning percentage of respective logical rules.

![Performance Chart](image)

*Figure 4.16 Sample of Performance Chart*

The knowledge of the expert is always assumed to be one hundred percents. As there may not be a chance that every puzzle can apply every rule, zero percent either indicate students are incapable of applying the logical rules or rules cannot be applied in that game.

Another one is the instruction usage chart. One of the samples is demonstrated in Figure 4.17. Different types of hints used are capable of finding out the weaknesses of student. Additionally, this chart also provides information about the improvement and sensitivity of students towards a particular type of instructions.
The last chart is the combination of three independent curves, which is the rules; misconceptions and hints relation chart (see Figure 4.18).
This chart demonstrates the most transparent information of the overall performance of students. There may be a case that the student is capable of solving the puzzle with a high knowledge gained rate. However, it may be achieved by the guess of students, which could be traced out if the misconception rate is high.

The student model has a close relationship with the database. The data captured during the game progress is record in the database for analysis. In the next section, the design of database is described.

4.5 Design of Database

Database record the important data for the student model. Apart from that, it also stores the Sudoku Steps Matrix and Sudoku Skills Matrix generated by the expert model. Figure 5.19 demonstrates the design of database. In the figure, every student of the application should provide information of his/her profile. Additionally, the system will automatically generate the waiting time for providing automatic instructions during the game. Each student is capable of solving all the suggested puzzles in the database. For each puzzle, records of skills matrix, solving steps, misconception and steps matrix are recorded. Since the table recording solving steps is only capable of recording all the declaration of cells, the elimination of cells is recorded by another table which cell elimination entries. Furthermore, table which supports reporting errors is also proposed to handle some exceptional case in the connection with the database such as fail in selecting a record, fail to add a record, etc.
After reviewing the table design of the database, the connection between database and applet is introduced. A direct connection between database and applet can only facilitate one client since the applet is launched in the client side. Therefore, in the proposed application, a three-tier connection is suggested. Figure 4.20 shows the design of database connection.
In the figure, the two side connection of database should transfer through the web server. In this way, the database can handle the data communications between applet and database with multiple clients efficiently and effectively.

After a detailed discussion on the design of the ITS in Sudoku, the implementation of the project is described in the next chapter.
CHAPTER FIVE
IMPLEMENTATION OF THE WEB BASED ITS IN SUDOKU

After the discussion about the design of the application, this chapter described how the design is put into reality. To begin with, the development tools and Environment of the proposed application are discussed. Afterwards, the detail of implementing different modulus is described. Moreover, some implementation difficulties and how they are addressed are also discussed.

5.1 DEVELOPMENT TOOLS AND ENVIRONMENT

As described in Section1.2, this proposed project is originated from the Intelligent Tutoring System in Sudoku for Logic Training (Yu, 2007). The originated project is developed using Java. Since some of the code developed can be reused in the proposed project, the proposed project develops an applet application and launches it in the web page. Table 5.1 shows the PC environment in implementing the application. Also, table 5.2 demonstrates the development tools applied during implementation.

<table>
<thead>
<tr>
<th>Window Edition</th>
<th>Window Server Enterprise Service Pack 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel(R) Core(TM)2 Duo CPU E8500 @ 3.16GHz 3.17 GHz</td>
</tr>
<tr>
<td>Memory (RAM)</td>
<td>4.00 GB</td>
</tr>
<tr>
<td>System Type</td>
<td>32-bit Operating System</td>
</tr>
</tbody>
</table>

Table 5.1 Application Development Environment
Table 5.2 Development tools of Application

<table>
<thead>
<tr>
<th>Tools</th>
<th>Software &amp; Version</th>
<th>Open Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support Jar file</td>
<td>jcommon-1.0.8.jar</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>jfreechart-1.0.13.jar</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mysql-connector-java-5.0.8-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bin.jar</td>
<td></td>
</tr>
</tbody>
</table>

5.2 IMPLEMENTATION OF LOGICAL RULES IN EXPERT MODEL

As discussed before, implementation of the expert model is the most complicated. There are altogether six logical rules that are required in this project. The algorithm in implementing them is illustrated in this section. The algorithm and implementation of rules are done by Yu(2007). In this project, the coding for the logical rules of expert model is reusable from Yu(2007). Therefore, the algorithm and figure described are the same as those in Yu(2007).
5.2.1 Algorithm in implementing SG

SG is the simplest and most straightforward logical rule. Figure 5.1 shows the algorithm of Single in Group. Since SG is applicable in all effective scopes. The algorithm loops for each scope and check whether there is only one unsolved cell in the same scope. If there is only one unsolved cell in each scope, SG can be applied and the unsolved cell is assigned the remaining value as its definite value.

![Figure 5.1 Algorithm of SG](image)

---

For each row, column and box

Is there only one cell unsolved in the scope?

Yes

Calculate the remaining value of the box

Set the remaining value as the definite value of unsolved cell

No

Is SG applied in any scope?

Yes

SG Applicable

No

SG Not Applicable
5.2.2 Algorithm in implementing NS

As discussed, NS is easy if an elimination tool is provided. Figure 5.2 demonstrates the algorithm of NS. Every unsolved cell is checked if NS is applicable. Based on the given cell values and the declared cell values in the same row, column and box as the unsolved cell, some of the possible values of the unsolved cell can be eliminated. If there is only one possible value left after the elimination and the only possible value of the cell is not declared by other rules, NS can be applicable. The definite value of the unsolved cell is set as the only possible value of the cell.

![Figure 5.2 Algorithms of NS](image-url)
5.2.3 Algorithm in implementing HS

Figure 5.3 shows the algorithm of HS. The application first counts the number of appearance of each possible value with respect to each scope. The possible value appears only once in the scope indicates that only one cell in the same scope can accommodate that possible. In this case, HS is applicable and the cell containing that possible value can be set it as the definite value.

![Figure 5.3 Algorithms of HS](image)

5.2.4 Algorithm in implementing IR

IR is applicable when the system ensures only a certain row or column in the box have the possibility to accommodate a certain possible value. Figure 5.4 illustrates the steps in implementing IR. The cell positions are recorded corresponding to the possible value that it may accommodate. If the cell positions recorded for a certain possible value are of the same row or column within the box, the possible value of other boxes of the same row or column should be eliminated. It is because the system can ensure only that row or column in the box should accommodate this possible value. Based on the constraint that each row, column and box should only
appear digit one to nine exactly once, the possibility of this possible value of other boxes in the same row or column can be eliminated.

**Figure 5.4 Algorithm of IR**

- **For each box**
  - **For each possible value**, record the cell position that is possible for the value.
  - **For each possible value**
    - Is there exist possible value with all the cell positions recorded are in the same row R or column C?
      - Yes
        - Is there any cell of other boxes containing this possible value in the row R or column C?
          - Yes
            - Eliminate this possible value in row R and column C of other boxes
            - IR Applicable
          - No
          - IR Not Applicable
        - No
      - No
      - IR Not Applicable
5.2.5 Algorithm in implementing NP and HP

Both the application of NP and HP are determined by the possible values pair appeared in the same row, column or box. Figure 5.5 shows the algorithm in finding the possible pairs. Possible values pair refers to the pair of possible value that appears in a pair of cells.

Upon the possible values pairs are found, the algorithm in the implementing the NP and HP become much easier. Figure 5.6 demonstrates the algorithm of NP and HP. Once the possible pair values are found to be the unique pair having those possible values, either NP or HP can be applied. It is because there are only two positions for the accommodation of two possible values. Based on the constraint of Sudoku, other positions are impossible to accommodate those two possible values. Meanwhile, those two cell positions also cannot accommodate values other that those two possible values.
5.3 IMPLEMENTATION OF AUTOMATIC INSTRUCTIONS PROVISION IN TUTOR MODEL

The algorithm in the generation of deductive arguments is similar to the implementation of logical rules in section 5.2. The only differences are that significant information is captured and some wordings are added to describe the premises and conclusion within the codes. The function for the generation of deductive argument results in the output message sends to tutor model under its request.

After discussing how the deductive arguments are generated, Figure 5.7 illustrates the algorithm in implementing and controlling the automatic instruction provision. There is always a loop to trace out whether the instant time is the suitable time for providing instructions. In automatic mode, the instructions are given level by level to
guide the students to get the definite value or eliminate some possible values from the semi-completed Sudoku puzzle.

![Figure 5.7 Algorithm of Automatic Instructions Provision](image)

### 5.4 Implementation the Evaluation Graphs in the Student Model

As discussed, there are altogether three graphs results from the evaluation in the student model. They are the performance chart, hints usage chart and hints, rules & misconceptions chart. These charts are implemented by the data recorded in the database using the JFreeChart. A sample code in creating a graph is shown in Figure 5.8.
The figure shows the sample code of the generation of hints usage chart. Within the codes, the parameters and attributes are well defined. Every curve of the graph sets with different colors for representing different data. JFreeChart capture the data and finally generate the image of the graph and send it to the Servlet for display.

### 5.5 Implementation of Web Page

Since the main concern of the proposed application is the Sudoku training through the ITS model, the web page functions such as login and register is modified from the open source code in the internet. In addition, the design of the web page which is the CSS is based on the template download from [http://www.freelayouts.com/](http://www.freelayouts.com/). Figure 5.9 shows the interface of the web page.

```java
JFreeChart hintsChart = ChartFactory.createLineChart("Hints Usage Chart", // chart title "Puzzle ID", // domain axis label "Number of Hints used", // range axis label hintsdataset, // data PlotOrientation.VERTICAL, // orientation true, // include legend true, // tooltips false // urls
);
hintsChart.setBackgroundPaint(Color.white);

CategoryPlot hintsplot = hintsChart.getCategoryPlot();
hintsplot.setBackgroundPaint(Color.white);
hintsplot.setDomainGridlinePaint(Color.white);
hintsplot.setRangeGridlinePaint(Color.white);

final LineAndShapeRenderer hintsrenderer = (LineAndShapeRenderer) hintsplot.getRenderer();
hintsrenderer.setShapesVisible(true);

hintsrenderer.setSeriesPaint(0, Color.red);
hintsrenderer.setSeriesPaint(1, Color.blue);
hintsrenderer.setSeriesPaint(2, Color.green);
hintsrenderer.setSeriesPaint(3, Color.MAGENTA);
hintsrenderer.setSeriesPaint(4, Color.orange);

hintsrenderer.setStroke(
    new BasicStroke(2f, BasicStroke.JOIN_ROUND, BasicStroke.JOIN_BEVEL));
```
5.6 COMMUNICATIONS BETWEEN APPLET AND DATABASE

As mentioned in section 4.6, the communication between applet and database should be under a three-tier model to support multiple clients. However, the actual implementation requires the use of two Servlets. One of them is responsible for handling the case of retrieve data from the stored database. Another one handles the modification of data in the database. Sample code of the Servlet is shown in Figure 5.10. In the figure, only partial code is shown. This is the important code in doPost function, which supports the retrieval of data. The coding of Servlet for modification is more or less the same expert the execute query change into update query (Red Color in the Figure). The applet prepares all SQL string and sends to the Servlet. Once the Servlet received the SQL, it performs the SQL and communicates with the database.
After implementing the proposed application, we should determine how well the implementation of the application. In the next chapter, the characteristics of a good ITS is determined. Additionally, it describes the testing on the application and discusses the result.
CHAPTER SIX

DISCUSSION AND EVALUATION OF ITS IN SUDOKU

The web based Sudoku in ITS aims to trace the development of logic through Sudoku learning. The essential characteristics of an ITS should be described to evaluate advantages and insufficiencies of the proposed project. In this chapter, the significant criteria of a good ITS is discussed. Afterwards, it illustrates how well the ITS model is adapted in this application. Then, results of a testing are demonstrated to shows the effectiveness of the system.

6.1 DISCUSSION OF A PERFECT ITS IN SUDOKU

According to Piramuthu, S.(2004), a good ITS should contain the a number of necessary characteristics. Firstly, the knowledge domain of expert knowledge should keep abreast. Additionally, it should cope with the problem in a different learning environment. Secondly, tutor model should provide tailored instructions. More specifically, the instructions given should base on the ability of student and real time performance. Thirdly, student model should record and evaluate the learning of student. Moreover, it should provide a mechanism to adjust the variables and attributes of learning rate and identifies the learning characteristics of student.

6.2 EFFECTIVENESS OF ITS ADAPTION IN THIS PROJECT

Based on the description of Section 6.1, the characteristics of an ITS model in that can applied in this project is summarized in Figure 6.1. In this figure, more detailed functions that can achieve a good ITS is also discussed.
## 6.3 Discussion on Insufficient Adaption of ITS

ITS is developing continuously and the demands of ITS are increasing. There are many characteristics that can be implemented in the ITS. Although there are already lists of functions is provided in the proposed ITS, there are still many requirements that can be added. In this section, some insufficient implementation of the ITS in Sudoku is proposed.

### Table 6.1 Characteristics of proposed ITS system

<table>
<thead>
<tr>
<th>Characteristics of a good ITS</th>
<th>Suggested functions</th>
<th>Applied in System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comprehensive Knowledge domain</td>
<td>Solve cell by Single In Group</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Hidden Single</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Naked Single</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Intersection Reduction</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Naked Pairs</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Hidden Pairs</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Aligned Pairs</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Y-Wings</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Solve cell by Multi-Coloring</td>
<td>No</td>
</tr>
<tr>
<td>Cope with different environment</td>
<td>Solve cell with different rules priority</td>
<td>Yes</td>
</tr>
<tr>
<td>Provide tailored instructions</td>
<td>Provide understandable instructions based on student language ability</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Provide understandable instructions based on student logical ability</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Provide levels of instruction</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Provide real time instruction</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Provide initiative instruction</td>
<td>Yes</td>
</tr>
<tr>
<td>Record student performance</td>
<td>Record student profile</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record solving steps</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record Sudoku steps matrix</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record Sudoku skills matrix</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record mal rules used</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record time to solve puzzle</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Record level of instructions used</td>
<td>Yes</td>
</tr>
<tr>
<td>Evaluate student learning</td>
<td>Identify learning characteristics of student</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Provide graphical descriptions of learning progress</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In this figure, a great proportion of criteria has been reached by the proposed application. In other words, the application is a well developed ITS application. However, there are still some areas that are insufficient in applying the ITS model. These insufficiencies are discussed in the next section.
6.3.1 Lack of Advance Rules in Expert Model

As discussed in the previous Chapters, only six out of nine rules are implemented in this application. Nevertheless, the declaration rules and elimination rules are insufficient for advanced students. Aligned Pairs, Y-wings and Multi-Coloring are powerful rules to solve the Sudoku puzzles. Since there is no existing open source of the advanced rules for the application, the implementation becomes difficult.

6.3.2 Unreal Tailor-Made Instructions in Tutor Model

Tailor made instructions refer to real time instructions that suitable for the student ability. In the proposed application, the instructions given depend on the priority of the rules and the current progress of the solving procedure.

The instructions provided by the application is said to be real time since it depends on the current progress of solving procedure. Instructions provided at the instant can be varied. In other words, for example, instructions given may base on SG rule or HS rules. In the proposed application, the rules applied at the instant are depending on the rules priority preferences inputted by students. Therefore, the instructions provided can also be said to be partly based on student suitability.

Consequently, instructions of the proposed application are said to be partly tailor-made. This is because students may not truly understand his/her need. Other measurement parameters can be applied to raise the suitability of instructions. For example, wordings of deductive arguments can be varied according to students’ ability. Apart from the wordings in describing the deductive arguments, the automatic adjustment the weighting of logical rules should also be triggered if students are found to have changed the familiarity of logical rules.
6.3.3 Inefficient in Time Management for Automatic Instructions

As discussed previously, the waiting time for the provision of automatic instructions is only depended on the statistics from Sudoku.org.hk. Actually, only concern the statistics is not enough for the calculation and initialization of waiting time.

In Chapter Two, some measurement parameters are considered for tutoring. One of them is emotional control. Conati(2009) suggested that instructions should be provided right before students get upset. Therefore, some emotional control management should be added to ensure the time for providing automatic instructions is suitable.

More than that, some learning abilities should also take into account. The cognitive state mentioned in section 2.8.2 is a sample to measure the learning abilities. The cognitive state can act as the parameters to calculate a more professional calculation on the knowledge gained, knowledge exposed and learning progress. The tutor model should base on the information of student ability, student background and student goods, etc to provide instructions. In this way, a more initiative and sensitive tutoring system can be achieved.

6.3.4 Incomprehensive Evaluation for Student Model

As discussed, the learning evaluation of the student model in the proposed project is determined by the performance chart, hints usage chart and rules, hints & misconception chart. However, these charts cannot provide a comprehensive evaluation on the student performance.

To begin with, one of the main purposes of the student model is information storage. In the proposed project, only student profiles and their solving steps are recorded. However, some other measurements such as the consciousness of students should also be recorded. It is because it may affect the performance of students.
Additionally, student model should be capable of evaluating the student learning procedures. The charts provided in the proposed project are not comprehensive. The charts with the parameters show the learning of rules, the use of instructions and the mal rules applied by students. However, these three parameters cannot fully reflect the learning of students. For example, the experience of students, his/her age group, and their ability in mathematics may also affect their learning. Experienced student may learn in a slower progress since he/she is already with enough knowledge and sense to solve the suggested puzzle. The ability in mathematics provides information on their background knowledge of logic. The learning ability may also depend on the age group.

6.4 TESTING OF THE APPLICATION

The web based Sudoku in ITS aims to trace the development of logic through Sudoku learning. A testing has been done to find out if the system can successfully train the logic of the student. In this section, background of testers and their results are described.

6.4.1 Background of Students

There are altogether 10 students participate in the test. All of them are ranged from 22 to 25. Within this group of people, 30% of them are considered of skillful students in which they have experiences in Sudoku for at least one year. One of them even has four years experience. 70% of them are navies; most of them are new to Sudoku.

6.4.2 Testing procedures
Students are called to finish 10 Sudoku puzzles with medium difficulty. For medium level of Sudoku puzzle, they have to solve from 52 to 56 emptied cell using different logical rules. Automatic instructions are provided when their responses are too slow or out of expectation.

In this testing, students with different experiences are called to solve the same group of Sudoku puzzles. This is because it helps in measuring the learning ability and speed of people from different backgrounds. Additionally, medium level of puzzles in chosen since gentle Sudoku puzzle is too easy for skillful students and difficult Sudoku puzzle is too complex for naïves. Furthermore, the number of cells to be solved in Sudoku puzzles is ranged from 52 to 56 and their appearance order is random.

6.4.3 Testing Results Evaluation

Figure 6.1 demonstrates the rules learning chart generated from the participation of 10 students in a testing. The red curve shows the average result of all participated students. The curve is increasing from puzzle ID between 1 and 5. This shows that learning has been occurred as the increased percentage is from about 45% to 65%. 20% is a significant increase. After that, there is a gradually decreasing from puzzle ID 6 to 10 and finally reach marginally above 55% learning. The other two curves (blue one and green one) illustrate the learning of skillful students and naive students. Obviously, the starting point of skillful students is almost 8% greater than the naive.
First, let’s discuss the decreasing leaning percentage from puzzle ID 6 to 10, no matter for experienced or non-experienced students. After the tracking of their time of conducting the test, the data recorded found that most of them are capable of solving the 10 puzzle within the same day, some of them even conduct the testing in 1am to 3am. Therefore, the reasons for the decreasing may be due to two reasons. The first one is learning and applies logical rules in different metal states. People are mentally poor at night; this may affect the learning ability. Another reason is that may be the puzzle with ID 6 to 10 is much more difficult. Although they are in the same difficult level, the percentage of each rule applied in each puzzle is different. In other words, students may feel more familiar with the rules required in solving the Sudoku puzzle ID from 1 to 5.

Actually, the learning percentage for experienced students is fluctuating from 55% to 65%. This phenomenon may conclude that the system may not be very effective on
skillful students. Skillful students are of based knowledge, the new knowledge that
the system provides to them comparative more complex. Many trail and examples
are required for them to learn or even understand. Therefore, 10 Sudoku puzzles
may not enough for skillful students to practice new knowledge.

However, for naïve students, even they are mentally poor when they participate in
the last few puzzles; their learning range is much greater that their starting point.
This can be explained that naïve students are capable of understanding some basic,
simple or easy knowledge of Sudoku. Even they feel asleep at night, they can even
apply a degree of knowledge they learn at the beginning of the test.

6.5 FURTHER TESTING FOR EVALUATION

Since there are a limited time and manpower during the testing and evaluation
process, the testing result provided does not indicate a full part of success of the
proposed project. Therefore, some test cases are proposed for further testing for the
adjustment of some variables in student learning progress and formulate a suitable
formula for student learning.

6.5.1 Test the learning function of application

This test case required 50 students to complete 20 Sudoku puzzles with the same
difficulty. Sudoku puzzle with medium difficulty is suggested. Students are asked to
finish two Sudoku puzzles every day when they are mentally well. For example, the
time that they have already woken up in the morning. This test case is used to
ensure the functions of the web based ITS in Sudoku do enhance the logical
development of student. A predicted graph is shown in figure 6.2.
Discussion and Evaluation of ITS in Sudoku

Average Learning Curve of Students

![Average Learning Curve of Students](image)

*Figure 6.2 Average predicted learning curve of student*

This testing has not been conducted because there are limited manpower and time. The testing have been conducted does not show the actual reflection of the learning of the project.

### 6.5.2 Test the mental state against logical learning

This test case is used to determine the variables which affect the logical learning. The test case is conducted by 50 students in which they are divided into 2 groups. One of the groups is called to finish 20 Sudoku puzzles with a day while the other group is called to complete the same categories of the puzzle when they are mentally well. The results obtained are used to deduce the variables that affect the performance of learning and applying logical rules in the puzzle.

This test case is proposed since the test described in section 6.4 demonstrates there is some effect on mental state against logical learning. One of the purposes of this test case is to verify the discussion of the test described in section 6.4. Another purpose is to deduce the variables in measuring the logical learning of student when he/she is mentally poor.
CHAPTER SEVEN

CONCLUSION OF SUDOKU IN ITS

After the evaluation of Sudoku in ITS, a conclusion of Sudoku in ITS is described. In this section, achievements of the project are discussed first. After that, the limitations of the project are also provided. Finally, some suggestions on the enhancement for the future are described.

7.1 ACHIEVEMENTS OF WEB BASED SUDOKU IN ITS

Web based Sudoku in ITS is the result of the proposed project. In this section, how the project fulfills the objectives of the project are described. Additionally, how well the adoption of ITS in the project is described.

7.1.1 Project Objectives Fulfillment

In this subsection, the objectives of the project are reviewed and their fulfillment is discussed. Chapter one stated that the main objective of the proposed project is to provide a platform for students to develop the logic mind in solving Sudoku puzzle based on the ITS model. The main objective is said to be achieved if all the additional objectives are reached.

The additional objectives and theirs achievements in this project are stated as follows:

1. Enrich my knowledge on the deduction of the logic, Sudoku rules and intelligent tutoring system
   Firstly, I have reviewed a cast number of researches on logic. Additionally, I have found out the most efficient and effective representation for students in developing their logical mind. The progress of logic deduction is also described.
Secondly, I have an extensive and comprehensive understanding of logical rules in Sudoku. Besides, the various representations of logical rules have been summarized and the deductive arguments and concept of the logical rules are deduced. Thirdly, the concept and recent development of the ITS have been reviewed. Some of the existing Sudoku applications which enhance logic learning are also reviewed. During the review progress, I have enriched my knowledge and all the above studies and reviewed are summarized in Chapter 2 of this report. Therefore, this objective has been achieved.

2. Derive high level requirements for logic training in Sudoku based on the concept of an intelligent tutoring system
   This objective is completed. In Chapter 3, the operations required for the expert model, tutor model and student model are described. These operations are suggested based on the important characteristics of an ITS described in Chapter 2. The conceptual representation also proposes some operations that fixed the common problems and challenges of an ITS, i.e. tutor model acts as initiative as human coach.

3. Design an application with functions that facilitate the logic training in Sudoku and based on ITS model
   This objective has been achieved. In Chapter 4, the design of the interface, expert model, tutor model and student model are discussed. Firstly, the application interface provides numbers of student settings to facilitate the operations of three modulus of ITS. Secondly, a control mechanism for each model and the overall object oriented class diagram is described.

4. Implement ITS model in the selected environment and with well defined algorithms
   In Chapter 5, the environment for the development of the application is proposed. As the project is based on the existing application using java, applet is suggested for the reusability of code. The algorithms for implementing the logical rules of the expert model are also discussed. In addition, the algorithms
for instruction generation and database communication with the applet are also discussed. Therefore, the objective is considered as complete.

5. Summarize the characteristics of a good ITS. Perform testing and evaluate how ITS model is adapted in the proposed application
The characteristics of a good ITS have been described in Chapter 6. Based on these characteristics, the report provides how well the application has been developed based on the adaption of the ITS model. Additionally, a simple testing has been conducted as there is only limited manpower and time. Moreover, some discussion on the insufficient adaption of ITS model is also described. Undoubtedly, this objective is achieved.

Since all the additional objectives have been achieved, the main objective is met. In the next subsection, a brief description of the successfully implemented functions is discussed.

**7.2.2 Successfully Implemented Functions**

Web based Intelligent Tutoring System in Sudoku for logic training provides a platform for students to utilize the application in the internet.

To begin with, the logical rules interpreted with examples are provided in the web page. The web page provides function such as registration to collect background information for students, which allows any future enhanced project to get the data for deciding the additional function of the enhanced project. Additionally, it also provides web page with performance of individual puzzle and trend of learning.

In addition, the applet application reuses most of the coding of the expert model from Yu(2007). The proposed project also modifies the two elimination rules of the expert model. They are NP and HP. Additionally, the steps matrix and skills matrix are also generated and recorded.
Moreover, the tutor model of the application does not only provide deductive arguments as instructions under request, but also provide automatic instructions when student performance is out of expectation. More than that, the waiting time for providing instructions could be changed by the human coach of the student.

Besides, the student model of the proposed project providing the knowledge learning of each logical rule. Apart from that, hints usage and misconception made are also recorded and present the data into curves.

Furthermore, the connection between applet and database is set up in a three tier model. The database is reconstructed from the previous file configuration system.

### 7.2 Limitations of Web Based Sudoku in ITS

In the web based Sudoku in ITS for logic training, the web page of the proposed system does not contain any role other than a student. A tutor role can be provided to control the logical rules priority and waiting time for automatically instructions for students. In the proposed application, human tutor can coach the student by the logical rules priority and waiting time for automatically instructions for students only when students login the page.

In addition, the puzzles to be solved are limited in the expert model. Since not all logical rules available are implemented in the expert model, the model is not capable of solving the advanced puzzles.


7.3 Further Enhancement of Proposed Project

In chapter six, the insufficiencies of applying ITS model in Sudoku for logic training are described. Further enhancement of the project is based on the insufficient utilization of ITS model.

To begin with, the three advanced rules should be implemented. As discussed, there is no existing source for the advanced rules’ implementation. Therefore, there is a high demand on the advanced rules.

Moreover, more measurement parameters should be added to facilitate the logical training by tutor model. Different presentation of instructions should be provided for students with different learning and logical abilities. Additionally, emotional parameters should be added to control the time for providing automatically instructions.

Furthermore, the information captured should be detailed. This helps in evaluation of learning progress of students. In addition, more graphs with a different combination of parameters should be displayed for analyzing the most efficient and effective way to represent the learning progress of student.
REFERENCES


## APPENDIX B MONTHLY LOG

<table>
<thead>
<tr>
<th>Date</th>
<th>Log</th>
</tr>
</thead>
</table>
| Sep, 2009| Reviews on Intelligent Tutoring System  
Reviews on Sudoku  
Reviews on Web Applications  
Reviews on Web based Learning  
Reviews on existing ITS system  
Reviews on existing Sudoku Applications  
Reviews on logic  
Restructure the ITS in Sudoku for logic training(from Carter) into Applet format |
| Nov, 2009| Implementation on ITS model  
- add strategy graph  
- add steps matrix  
- add skills matrix  
- analyze rules used by user  
- provide automatic hints  
- store steps matrix, skills matrix and user solving steps in DB |
| Dec, 2009| Research on Student Model  
Implementation on ITS model  
- presentation on student performance on student knowledge  
- add record for the elimination rules |
| Jan, 2010| Implementation  
-Student Model Data presentation  
-Expert Model (Naked Pairs and Hidden Pairs)  
-Evaluation of student performance |
| Feb, 2010| Research on relationship between student model and tutor model  
Implementation  
-Student Model Data presentation  
-Database connection(applet) improvement  
-Chinese/English interface |
| Mar, 2010| Finalize all programme code  
Write final year project report |