

# AN APPROACH TO EMBEDDING ITSs INTO EXISTING SYSTEMS

---

A thesis submitted in partial fulfilment  
of the requirements for the Degree of  
Doctor of Philosophy  
in the University of Canterbury  
by Sagaya Sabestinal Amalathas

---

Supervisor: Professor Antonija Mitrovic

Associate Supervisor: Professor David Evison

Associate Supervisor: Mr. Ravan Saravanan

Department of Computer Science and Software Engineering

University of Canterbury

2012

## **ACKNOWLEDGMENTS**

I owe my eternal gratitude to Professor Tanja Mitrovic for accepting me as her student and to Mr Ravan Saravanan for encouraging me to pursue this PhD. I would like to also thank Professor Tanja Mitrovic for the supervisory role played in guiding me to complete this research and thesis, for the advice given, for being so very patient with me and answering my questions tirelessly. I am very grateful to Mr Ravan and RANN Consulting for funding this research. I am deeply indebted to my daughter, Angelica, the best daughter in the world a mother could ever have, for being far more understanding than I ever expected her to be. I would like to thank my friends Pushpa, Ray Hidayat, David Thomson and Jaime Galvez Cordero for their support and valuable advice that was very helpful for the creation of this thesis.

# ABSTRACT

Intelligent Tutoring Systems (ITSs) have proven their effectiveness in many domains, but very few attempts have been made to embed them with existing systems. This area of research has a lot of potential in providing life-long learning and work place training.

This PhD project makes several significant contributions. This is the first attempt to embed a Constraint-Based Tutor (CBT) with an existing system, in order to investigate the benefits of providing on-the-job training. We also propose a framework for embedded ITSs, and develop DM-Tutor (Decision-Making Tutor) embedded with the MIS for palm oil. DM-Tutor is the first ITS for the domain of oil palm plantation decision making, and was developed in the ASPIRE authoring system. Our hypothesis was that DM-Tutor embedded with the MIS for palm oil would provide effective instruction and training for oil palm plantation decision making. We also wanted to investigate the role of feedback messages in helping to provide effective training.

We conducted two evaluations to test our hypothesis. A pilot study was conducted with a group of volunteer employees from an oil palm plantation company. From the results and analysis of the pilot study it is proven that the participants' knowledge of oil palm plantation decision making improved significantly after interacting with DM-Tutor. A full evaluation study was conducted with a group of student volunteers enrolled in a Bachelor of Agriculture program. The study shows that students' knowledge improved significantly after interacting with DM-Tutor.

Through this research we have proven that DM-Tutor embedded with the MIS for palm oil provided effective instructions and training. Using DM-Tutor developed with the support of ASPIRE and embedding it with the MIS for palm oil we have developed an architecture for embedded ITSs. Finally, through this research we have proposed our approach to creating a framework for embedded ITSs.

# TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>TABLE OF CONTENTS</b> .....	<b>iii</b>
<b>CHAPTER 1</b> .....	<b>1</b>
<b>Introduction</b> .....	<b>1</b>
1.1 Decision Making in Oil Palm Plantations.....	2
1.2 Computer Based Training .....	3
1.3 Intelligent Tutoring Systems.....	5
1.4 Aims of the project.....	7
1.5 Thesis Outline .....	7
<b>CHAPTER 2</b> .....	<b>9</b>
<b>Background Research</b> .....	<b>9</b>
2.1 Intelligent Tutoring Systems.....	11
2.1.1 Domain Module .....	14
2.1.2 Student Model & Modeller .....	17
2.1.3 Pedagogical Module.....	19
2.1.4 Interface Module .....	20
2.2 Aspire.....	21
2.3 ITSs Integrated with Existing Systems or Training Tools.....	25
2.3.1 Lumiere/ Excel Project .....	26
2.3.2 An Automated Consultant for Macsyma .....	27
2.3.3 Training Wheels in a User Interface .....	29
2.3.4 Plug-In Tutor Agents .....	31
2.3.4.1 Tool-Tutor for Geometric Construction using Geometer Sketchpad .....	32
2.3.4.2. A Tool Tutor for Algebra Problem Solving using Microsoft Excel .....	33
2.3.4.3 Geometer Sketchpad and Excel Tutor as learning environments .....	33
2.3.5 Embedded training for Complex Information System.....	34
2.3.6 Integrating ITSs and Health Information Systems.....	36
2.3.7 Personal Access Tutor (PAT) .....	39
2.3.8 xPST (Extensible Problem-Specific Tutor) .....	41
2.3.9 SBT-AID (Scenario-Based Training: Adaptive, Intelligent, Dynamic) Training System 44	
2.4 Summary of Embedded ITSs .....	45
<b>CHAPTER 3</b> .....	<b>48</b>
<b>Domain of Palm Oil</b> .....	<b>48</b>
3.1 Management Information System (MIS) for Palm Oil .....	49
3.2 MIS Development Environment .....	50
3.3 Reports and Analyses.....	50
3.4 MIS Challenges.....	55
<b>CHAPTER 4</b> .....	<b>57</b>
<b>Design and Development of DM-Tutor</b> .....	<b>57</b>

4.1	Design of DM-Tutor .....	57
4.2	Problem Design.....	58
4.2.1	Yield Gap Analysis .....	59
4.2.2	Fertilizer management analysis.....	60
4.2.3	Yield Forecasting Analysis .....	61
4.3	Components of DM-Tutor .....	62
4.4	Development .....	64
4.4.1	Characteristics of the domain.....	64
4.4.2	Ontology development.....	65
4.4.3	Problem and Solution Structure .....	68
4.4.4	Student Interface .....	70
4.4.5	Adding Problems and Solutions.....	72
4.4.6	Generation of Syntax and Semantic constraints .....	74
4.4.7	Deployment of DM-Tutor.....	77
4.5	Yield gap analysis task.....	79
4.6	Fertilizer management analysis task .....	80
4.7	Forecasting analysis task.....	81
<b>CHAPTER 5 .....</b>		<b>82</b>
<b>An Approach to integrating ITSs into existing systems .....</b>		<b>82</b>
5.1	Requirements .....	82
5.2	Comparisons with other embedded systems .....	85
5.3	Architecture for embedded ITSs .....	87
5.4	Integration of DM-Tutor with the MIS .....	88
<b>CHAPTER 6 .....</b>		<b>93</b>
<b>Evaluation.....</b>		<b>93</b>
6.1	Pilot Study	94
6.1.1	Design and Implementation .....	94
6.1.2	Results and Analysis for Pilot Study .....	96
6.2	Full Evaluation Study .....	98
6.2.1	Evaluation Design.....	100
6.2.2	Pre and Post Tests .....	101
6.2.3	Recruitment of participants.....	101
6.2.4	Evaluation Procedure .....	103
6.2.5	Results and Analysis for Full Evaluation Study .....	103
<b>CHAPTER 7.....</b>		<b>113</b>
<b>Conclusions and Future Work.....</b>		<b>113</b>
7.1	Research Limitations .....	117
7.2	Future Directions .....	119
<b>REFERENCES.....</b>		<b>122</b>
<b>Appendix A: Information Sheet .....</b>		<b>129</b>
<b>Appendix B: Consent Form .....</b>		<b>130</b>
<b>Appendix C: HEC Approval.....</b>		<b>131</b>
<b>Appendix D: Pre-Tests and Post-Tests .....</b>		<b>132</b>
<b>Appendix E: Questionnaires .....</b>		<b>136</b>

<b>Appendix F: Problems attempted by the control Group During Full Evaluation Study.....</b>	<b>142</b>
<b>Appendix G: Published Papers.....</b>	<b>154</b>

## LIST OF FIGURES

Figure 2.1: Typical architecture for ITS .....	14
Figure 2.2: Architecture of ASPIRE author with components included [Mitrovic, 2009] .....	22
Figure 2.3: General architecture for plug-in tutor agents [Ritter & Koedinger, 1996]...	31
Figure 2.4: Architecture of ETS with the integration components included .....	35
Figure 2.5: ITS-CBR architecture and integration component [Gonzalez et al., 2007] .	38
Figure 2.6: Architecture and main components of PAT [Risco & Rey, 2009].....	40
Figure 2.7: Architecture of xPST system [Gilbert et al., 2009].....	42
Figure 2.8: Architecture of SBT-AID [Shatz et al., 2009].....	45
Figure 3.1 Screen-shot of the MIS interface viewed when users first log-in into the system. ....	51
Figure 3.2: Screen shot of yield analysis report.....	52
Figure 3.3: Screen shot of material consumption report.....	55
Figure 4.1: Components of DM-Tutor.....	62
Figure 4.2: Specifying the domain details for the yield gap analysis task.....	65
Figure 4.3: Ontology for fertilizer management analysis .....	67
Figure 4.4: Solution structure for yield gap analysis task.....	69
Figure 4.5: Student interface of DM-Tutor.....	70
Figure 4.6: Creating problems for fertilizer management task .....	73
Figure 4.7: Ideal solution for fertilizer management task.....	74
Figure 4.8: Syntax constraints for fertilizer management.....	75
Figure 4.9: Semantic constraints for fertilizer placement task .....	77
Figure 4.10: Deployment page for forecasting task.....	78
Figure 5.1: Integration architecture of MIS and DM-Tutor.....	87
Figure 5.2: DM-Tutor interface presented in the MIS showing tasks available for the users. ....	89
Figure 5.3: Detailed report shown in separate window in the MIS .....	90
Figure 5.4: Integrated problem solving interface.....	92
Figure 6.1: Probability of violating a constraint as a function of the occasion when that constraint was relevant.....	108

## LIST OF TABLES

Table 6.1: DM- Tutor interaction results (S.D. given in parentheses).....	97
Table 6.2: System interaction statistics (S.D. given in parentheses) .....	104
Table 6 3: Pre-test and post-test scores (S.D given in parentheses) .....	106
Table 6.4: Scores from the questionnaires (Likert scale from 1-5) S.D. given in parentheses. ....	110



# CHAPTER 1

## INTRODUCTION

Over the last few years there has been a growing need for creating learning at a workplace environment. Rapid technological changes and the competition between local and international organizations have led many organizations to require their workforce to acquire the skills needed to fulfil the needs of the ever changing work environment and customers' requirements. Workplace learning has certain characteristics that differentiate it from other types of learning. Workplace learning has to be task focused, interactive, learner driven and flexible enough to cater to each individual employee's needs. Workplace learning is cognitively different from learning in schools or other educational environments. Workplace learning should be done in a just-in-time manner and provide knowledge at the required level to the employees [Matthews, 1999].

Even though organizations generally understand the need for workplace learning, many of them still limit the amount of training and learning provided to employees due to time and cost constraints, so in most cases employees have to resort to learning on-the-job. Placing employees who have been trained on-the-job, could have dire consequences as the ineffective decisions they make could make organizations risk losing money and business opportunities.

### **1.1 DECISION MAKING IN OIL PALM PLANTATIONS**

Plantation decision making is a complex domain. Employees, especially managers, are required to know all the conditions and plantation inputs before making a decision involving plantation resources, cost controls or planning for future development. Many of the oil palm plantations are managed remotely from the central offices located far from the actual plantations and managers are expected to make decisions catering to the highly dynamic and ever changing plantation conditions. Due to the high level of complexity involved, many oil palm plantations rely on plantation management software to help them manage their day-to-day plantation activities.

The Management Information System (MIS) for palm oil industry [Ravan, 2007] provides an integrated software solution for managing oil palm plantation operations. The MIS modules provide for resources management, future plantation growth and other management concerns. As the information is highly domain specific and requires detailed knowledge of the oil palm domain and the MIS, managers who are new to the domain or to the MIS face difficulties in making accurate and well

defined analyses. This in turn affects the management and operational decisions they make.

### **1.2 COMPUTER BASED TRAINING**

There are various types of computer-based training and e-learning applications that are used in educational environments and workplaces. Convenience, standardized delivery and content variety have made these methods of learning very popular in schools, colleges and workplaces. Learners love the flexibility and independence it gives them. Computer-based training or computer-aided instruction systems generally introduce facts and concepts to many people at the same time and usually test knowledge learned by providing a standard set of relatively simple questions. They are not able to provide intelligent and personalized training. Even though they do assess performance, they neither provide personalised feedback nor model the students.

Web-based training (WBT) is delivered through the internet and could include videoconferencing and hyperlinked web pages. This method of training is widely used in situations where bringing all the students together or teaching them individually would be cost prohibitive. Collaborative training through WBT allows students to interact more directly with systems. A trainer could be conducting training to hundreds of employees working in another city. The students will be able to see the trainer and even hear the teacher while interacting with system during training. Certain WBT or collaborative trainings even allow students to click a button and ask

the teacher a question and wait for a response from the teacher. Even though WBT overcomes the logistical difficulties for collaborative training and creates more interactivity compared to traditional computer-based training, it has similar pedagogical limitations. When faced with a problem, the students have to wait for help from an online instructor, as the number of students in collaborative training are usually high the students need to wait for a long time to get a response from the instructor [Ong & Ramachandran, 2003]. These training technologies could be appropriate if the aim of the training was to produce trained operators who require knowledge at the novice level. However, to become truly proficient in decision making, an employee requires extensive practice solving realistically complex problems in a wide range of situations, combined with coaching or feedback from senior or more experienced peers, or experts in the domain.

Computer-based simulation provides training on different actions in a variety of simulated situations. Simulated training was first used by the aviation industry and the military. However, they are also being used by businesses to train employees on engineering, business, analytical and interpersonal skills. Simulations are effective when they are relatively simple and expose students to just a few actions at each step. As these simulations become more complex and present many possible actions to the students and attempt to model a variety of cause-and effect relationships during the training, it becomes increasingly difficult for the student to exactly determine what action they accomplished well or which particular action went wrong [Ong & Ramachandran, 2003].

### 1.3 INTELLIGENT TUTORING SYSTEMS

It is well known that one-on-one human tutoring is much more effective than traditional classroom instruction as it provides an increase of two standard deviations in learning performance [Bloom, 1984]. Imagine if each learner in a classroom or a workplace environment could have their own personal tutor that could pay attention to all their learning needs, diagnose problems and provide guidance when the learner requires it, that would be every learner's dream albeit an impossible one. Providing a personal tutor or trainer for each student or employee is beyond the teaching and training budget of any learning institution or organization.

The solution to this problem could come from Intelligent Tutoring Systems (ITSs). The goal of ITSs is to provide the benefits of one-on-one teaching or training to one or many at the same time. Unlike other computer-based training technologies, ITSs assess the learning actions of the students within a highly interactive learning environment. ITSs model a student's current knowledge and use this model to adapt teaching strategies appropriate for the student. ITSs provide feedback to the student that can guide the learning actions of the students and help to increase their knowledge of the domain.

ITSs have become an important class of educational technology that is capable of playing a crucial role in helping learners of any age acquire the skills needed to succeed. ITSs have been developed and effectively used for teaching and learning for many years and have been proven in the past to be successful in providing tutoring in various domains [Freedman, 2000]. ITSs hope to achieve the same effectiveness and benefits of personal tutoring with learning gain of 2 standard deviations, as achieved

by human tutors in one-to-one interactions with students. Over the years many ITSs including LISP tutor [Anderson & Reiser, 1985], Andes [Van Lehn et al., 2005], PUMP Algebra Tutor [Koedinger et al., 1997] and others have been successfully used. SQL-Tutor [Mitrovic, 2003], NORMIT [Mitrovic, 2002], KERMIT [Suraweera & Mitrovic, 2004] and UML-Tutor [Baghaei et al., 2007] are among the many constraint-based tutors (CBTs) [Mitrovic et al., 2007] that have been developed and successfully implemented by the Intelligent Computer Tutoring Group.

Even though ITSs have been proven as effective teaching tools, there have been very few attempts to embed them within other systems. Some of the attempts made to embed ITSs into existing systems are as follows; Macsyma Advisor was developed to assist users in using Macsyma, the algebraic manipulation system [Genesereth, 1979]. Excel Tutor provided descriptive explanations and interactive guidance for students to solve excel problems and Geometer Sketchpad Tutor taught students how to sketch geometric diagrams [Ritter, & Koedinger, 1996]. ETS (Embedded Training System) was developed and integrated with an existing Complex Information System for training on military tasks and operations [Cheikes et al., 1998]. PAT (Personal Access Tutor) was added into MS Access to help students build reports and forms using MS Access [Risco & Reye, 2009]. We will discuss the above systems in further detail in Chapter 2 and compare them to our research in Chapter 6.

### 1.4 AIMS OF THE PROJECT

We aim to make several contributions through this research project. This will be the first attempt to embed a CBT with an existing system. Secondly, we will investigate the benefits of providing on-the-job training through this integration. Finally, and very importantly we aim to create a framework for embedded ITSs. We aim to develop an architecture for embedded ITSs taking into account of using DM-Tutor (Decision-Making Tutor) developed with the help of ASPIRE [Mitrovic et al., 2009] authoring system embedded with the MIS for palm oil.

DM-Tutor embedded with the MIS would provide training on plantation decision-making using real-life operational data from the MIS. We believe that our research has many potential benefits. Apart from the research contributions discussed above, DM-Tutor also has the potential benefit of providing life-long workplace learning, anywhere and anytime.

### 1.5 THESIS OUTLINE

In Chapters 2 and 3 we describe the background of our research. Chapter 2 covers the discussion on ITSs, Constraint Based Modelling (CBM), ASPIRE authoring system and past research on embedded ITSs. In Chapter 3 we describe the oil palm plantation domain. In Chapter 4 we describe the development of DM-Tutor. In chapter 5 we discuss our approach to integrating ITSs into existing systems. Chapter 6 discusses the evaluation study of DM-Tutor and the analysis of the evaluation results. Finally,

in Chapter 7 we draw the conclusions of our research and our proposed framework for embedding ITSs with existing systems.



# CHAPTER 2

## BACKGROUND RESEARCH

Many educational systems have been developed for teaching, training and also as ‘help’ tools in both the educational world and workplace environments. These systems have successfully taught users how to operate equipments, manage databases, handle software and carry out many daily learning and work activities.

The ICTG group at the University of Canterbury developed a suite of web-based ITSs to help students learn about databases. SQL-Tutor [Mitrovic, 2003] teaches SQL query language, NORMIT is a data normalization tutor and KERMIT [Suraweera & Mitrovic, 2004] taught conceptual database modelling using the Entity-Relationship data modelling. Repeated studies have shown that all three tutors have been successfully used by students in various classroom settings to learn the database language.

PUMP Algebra Tutor (PAT) [Koedinger et al., 1997] was developed to teach high school algebra using “real” world situations. When the tutor was used in three

public high schools, students responded positively towards using PAT to learn algebra. Teachers commented that when using PAT students did not present the learning difficulties that existed when students learned algebra through traditional methods in classrooms.

A Recovery Boiler Tutor (RBT) [Woolf et al., 1986] was developed to simulate the complex processes involved in pulp and paper mills. The tutor was built after a series of industrial incidents had taken place in the pulp and paper mills. Accidents had occurred mainly due to human errors. The objective of the tutor was to teach operators to solve boiler problems. The operators were encouraged to change settings on RBT to observe the resulting actions in the boiler. RBT has been successfully used in actual trainings in pulp and paper mills throughout the United States of America.

Sherlock [Lesgold et al., 1988] was developed to provide training for complex troubleshooting job in the air force. Sherlock provided coached practice and feedback to novices. Trainees were given full control to explore Sherlock during their practice, whenever they digress from the correct solution path they are given warnings and reminders to help them to come back to the correct solution path. In the version of Sherlock described in this research, each student saw 95 hints during 20 hours of practice. Novice technicians that have practiced about 20 to 25 hours in Sherlock are able to solve system failures at the level of colleagues with more than four years of on-the-job experience. The evaluation study conducted showed that Sherlock was effective in providing coached practice and individualized instructions to trainees.

Online help tools are also widely used by novices in business environments and also online learning environments. iHelp [Wang et al., 2010] is an intelligent helpdesk system that provided help for online customers. When a customer made a query, iHelp tried to match problem and solution patterns between customers and customer representatives. It searched and ranked past cases based on relevance to provided recommended solutions to users. User studies conducted with a group of students proved that iHelp was effective for the tasks like opening accounts, software installations, printer problems, making new orders and network connection problems. The participants also reported that they were happy to use iHelp.

Many ITSs and educational systems have been developed; however, very few have been embedded with existing systems. Embedded ITSs have the potential of leveraging in the knowledge contained within existing systems to provide tutoring or training. In this chapter we describe research on two approaches used to develop ITSs; model tracing tutors (MTTs) and constraint based tutors (CBTs) and we present past attempts on embedding ITSs and educational systems with existing systems.

### **2.1 INTELLIGENT TUTORING SYSTEMS**

The general requirements for an ITS is that an ITS must contain knowledge of the domain (expert model), knowledge of the learner (student model) and knowledge of teaching strategies. In an ITS, students' learning occurs mainly through solving problems. Components of the ITS work together with the objective of allowing problem solving to occur in an efficient manner. The student's knowledge of the

domain is reflected in the student model and is updated every time they submit their solution to the ITS. The component that provides knowledge in an ITS is called the domain module, curriculum or domain expertise. ITSs have to consider also how to present a certain concept of learning to the student and the learning strategy that is most appropriate for the domain. ITS provides knowledge through a learning strategy to students in the form of problem solving and collects student's solutions to be diagnosed through the interface [Shute & Psotka, 1994].

Two prominent techniques have been widely used in developing ITSs; the first is the traditional approach of model-tracing tutors (MTTs) or cognitive tutors. MTTs follow the eight principles generated based on the Adaptive Character of Thought-Rational (ACT-R) theory which states that to obtain knowledge, a great number of IF-THEN rules have to be formed for all the goals and conditions of each task; and the actions and consequences resulting from them [Anderson, 1996]. PUMP Algebra Tutor [Koedinger et al., 1997], Andes Physics Tutor [Van Lehn et al., 2005] and LISP Tutor [Anderson & Reiser, 1985] are among the successful MTTs that have been implemented.

The second approach, Constraint-Based Tutors (CBTs) use constraint-based modelling (CBM), a technique based on the theory of learning from performance errors [Ohlsson, 1996]. In this approach, knowledge is represented as constraints that specify what the solution ought to be, instead of generating a multitude of problem-solving paths. Constraints are used to model both the domain and the student. CBM matches the pattern in student's solution with the ideal solution stored within the system and does not need to use a runnable expert model or an extensive bug library,

therefore, making it computationally simpler and more cost effective compared to MTTs. CBM does not force one problem solving path compared to MTT. The students are given more freedom to choose their own problem-solving procedure as CBM only evaluates the submitted solution and does not infer the path the student needs to take to arrive at the solution [Mitrovic, 2001]. SQL-Tutor [Mitrovic, 2003], NORMIT [Mitrovic, 2002], KERMIT [Suraweera & Mitrovic, 2004] and UML-Tutor [Baghaei et al., 2007] are among the CBTs that have been developed and successfully implemented. CBM was used to develop DM-Tutor for this research project.

A four-model component architecture currently used by various types of ITSs consisted of the three major components traditionally used with one additional component, the user interface [Dede, 1986]. Researchers further separated the ITSs architecture to a five component architecture by dividing the domain expertise into domain knowledge and domain expert [Beck et al., 1996]. The general architecture for ITSs especially CBTs can be illustrated as below in Figure 2.1.

The domain module component contains knowledge the ITS aims to teach the students. The student model consists of the current knowledge of the learner and could also include cognitive processes, meta-cognitive strategies and psychological attributes of the learner. In the pedagogical module, by using the learner's current state of knowledge, an efficient path is formed to achieve the desired expert level of knowledge. Through the user interface, a two-way communication is achieved between the ITS and the learner. We discuss next, each component of the ITS while comparing the two main approaches used in developing them.

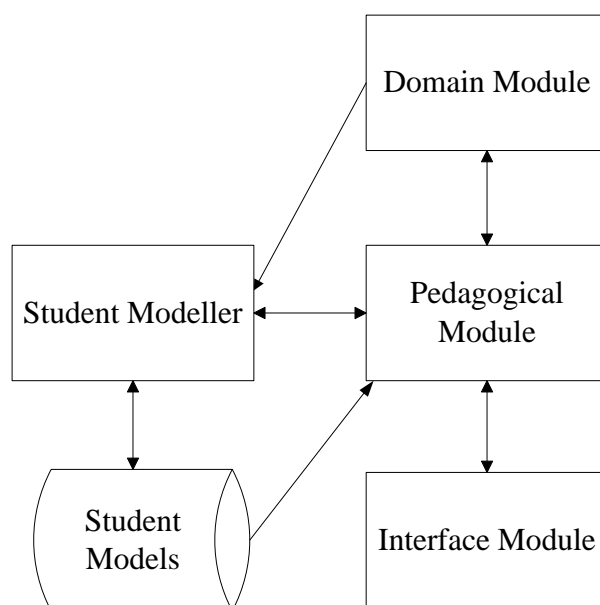


Figure 2.1: Typical architecture for ITS

### 2.1.1 DOMAIN MODULE

This component is sometimes referred to as the knowledge base and contains the declarative and procedural knowledge of the domain to be taught in the ITS. In this section we will describe how domain knowledge is captured in MTTs and compare with the CBM method of developing the domain module.

MTTs follow the ACT-R theory that requires goal-independent declarative knowledge of the domain to be converted into production rules that are goal oriented. These production rules represent procedural knowledge with each rule representing one part of the domain model. MTTs consists of expert (production) rules that model all the steps an expert would take to solve the given problem and a model-tracer that looks for all the possible sequences of rule execution until it identifies a match with the student's solution [Anderson, 1996]. Each production rule is written as an "IF-

THEN” pair. The “IF” part of the rule consists of a goal and a state, and the “THEN” part is made up of the action and/or the new goal or state to be achieved. Below is a possible production rule for a geometry domain:

*IF the goal is to prove two triangles are congruent*

*THEN set as a subgoal to prove that corresponding parts are congruent*

The domain module can generate all the steps required to solve a problem in the domain. Every action occurring during student’s interaction with the ITS is evaluated by the system. When students perform a wrong problem step the MTT informs the student that the action was incorrect and also why it is incorrect [Anderson, 1995]. The system restricts the student from moving off the predefined problem solving path. A bug library is created for all the possible incorrect actions in the domain and is used to provide feedback when students perform a wrong step in the system. Despite the success of model-tracing tutors, they do have their limitations. During the development of MTTs, all the problem-solving paths have to be explicitly programmed [Ohlsson, 1994]. If students use a problem-solving strategy that is different from the one specified, MTTs will reject their step even though their actions could possibly have arrived at the correct solution. Developing a production-rule set that could provide sufficient coverage of the domain consumes a lot of time. Past research has shown that it could take 100 development hours to produce 1 hour of instruction [Woolf & Cunningham, 1987].

In CBM domain knowledge is considered as an abstract manner of prescribing features of the correct solution and is represented as a collection of constraints. Constraints represent only correct knowledge in terms of pedagogically significant

states. Each constraint includes an ordered pair made up of a relevance condition and a satisfaction condition [Ohlsson, 1994]. We can describe constraints using an If-Then statement, i.e. “IF condition X is true, THEN (for this constraint to be satisfied) it must be the case that condition Y is also true”. Below is an example of a constraint for the Lisp programming domain:

*IF the code for a Lisp function has N left parentheses,*

*THEN there has to be N right parentheses as well (or else there is an error).*

In the example above, ‘the code for a Lisp function has N left parentheses’ is the relevance condition. The code should have ‘N right parentheses as well’ is the satisfaction condition. The relevance condition describes in terms of problem/solution structure when this constraint is applicable. If the constraint is relevant (applicable) then the satisfaction condition is applied to the student’s solution to check whether it is correct. If the student’s solution is incorrect this means that the constraint has been violated. When a particular solution does not violate any constraint CBM considers it to be correct. This is contrasting from MTTs that typically assume that when a student performs an action outside the predefined solution paths, their action is wrong. In CBM there are two types of constraints, syntactic and semantic constraints. Syntactic constraints represent the problem-independent structural knowledge of the domain. Semantic constraints are problem-dependent; they use the information available in the problem statement or the coded ideal solution and compare it to the solution submitted by the student. As constraints are better explained in relation to a particular domain, we will discuss the constraints developed for our system DM-Tutor in detail in Chapter 5.



### 2.1.2 STUDENT MODEL & MODELLER

A student model describes student's progress in learning a particular domain. It can be derived from the results of their system usage and problems attempted in the system. Information from the student model is then used to select appropriate tutoring strategies for the student. Traditional methods of developing student models such as machine learning and bug libraries require a vast amount of knowledge to create a comprehensive student model. However, research on previous developed ITSs has shown that even student models that are not complete or have inaccuracies in them can still be useful and valid as we need to model students for the tasks required in a particular teaching system and not more than that [Self, 1990].

There are several approaches used in creating student models. The first one is the Bug library technique. In this method, students' solutions to problems are studied and all recurring errors are listed. Explanations are constructed to explain how each error could have occurred. Identifying errors is a complicated process as it requires extensive research of the domain. It is also computationally expensive. The second approach is machine learning. Machine learning looks at all the possible procedures for all the possible paths required for a correct or incorrect procedure. The computational complexity required for machine learning is even greater than the bug library technique [Ohlsson, 1994].

The next technique described has been widely used to model students; it is the traditional approach of model-tracing technique that also requires the usage of a bug library. In this technique student's problem solving actions are monitored step-by-step

using a library of correct and incorrect rules. By tracking the student's steps and matching their steps against the rules library, a model of the student is derived. In MTTs, a domain model is represented as a production rule set and the student model is developed using a subset of the production rule set of the domain combined by the student's erroneous knowledge of the domain (known as buggy rules). Buggy rules try to reflect all the possible misconceptions students could have of the domain [Anderson, 1996]. MTT still depends on a set of rules to model the student's correct and incorrect knowledge. Developing all these rules require extensive research of the domain and the student population. Another problem with MTTs is that it is very difficult to interpret student's problem solving steps outside the predefined paths allowed by the MTT. MTTs resort to limiting the possible solution paths the students should take and if the student leaves the allowed path it becomes almost impossible for the cognitive tutor to follow the student's problem solving actions [Ohlsson, 1994].

In CBM constraints are used to represent the student's knowledge or student model. When a student submits her/ his solutions, the CBT analyses the student's solutions in several steps using constraints. In the first step, the constraints that are relevant to the student's solutions are identified. If the constraint is relevant, the student's solution is checked to see whether it satisfies the constraint. If the student's solution violates the constraint, an appropriate feedback message that is attached to each constraint is provided to the student. When the constraint is violated the feedback message informs the user that their solution is wrong and the reason why it is wrong [Mitrovic et al., 2007]. If the student's solution satisfies the constraint, no

action is taken by CBM and the student may proceed to the next problem. The list of the relevant, satisfied and violated constraints forms the short-term student model. The short-term model is then used to update the long-term model of the student's understanding of the domain. The history of each relevant constraint for a solution is retained in the CBT. This allows the system to gauge the student's knowledge, in terms of the correct knowledge the student has of the domain (satisfied constraints) and also the student's incorrect knowledge of the domain (violated constraints) [Mitrovic, 1999], [Suraweera & Mitrovic, 2004].

### 2.1.3 PEDAGOGICAL MODULE

A pedagogical module (PM) takes care of all the teaching decisions made in an ITS. PM selects the relevant teaching strategy for the student based on the student model. It also provides feedback appropriate to student's learning actions in the ITS. Approaches used by PMs vary between ITSs and sometimes even between domains. In cognitive tutors, students are given feedback for every step they make while attempting to solve the problem. Errors are detected when a student's action does not match the rule for the correct knowledge or when it matches any one of the buggy rules. Curriculum managers help in selecting problems for the students to attempt based on their student model.

Like MTTs, CBTs can also track the student's actions step by step. However CBTs can be implemented not to follow the student's problem-solving actions step-by-step and only evaluate CBTs student's solution once it is submitted. The student

modeller identifies constraints that have been violated by the student's solution and appropriate feedback is provided to students on their incorrect solution [Mitrovic, 1999, Suraweera et al., 2004]. The feedback students receive varies among ITSs. ITSs could be designed to provide immediate feedback or delayed feedback. Generally ITSs give negative feedback, i.e. they inform students of their problem solving error and how they could correct them. However, in recent studies, the effect of positive feedback is being researched as well. In CBTs, PMs allow students to have the option where they can rely on the system to choose the next suitable problem for them to attempt based on their student model or to select the next problem they wish to attempt on their own.

### 2.1.4 INTERFACE MODULE

The interface module of an ITS allows students to interact with the system. Students retrieve problems, submit their solutions to the problems and also receive feedback from the systems on their solutions or problem solving actions through the interface module. As this is the only component of the ITS the students are able to view physically, the interface module has to fulfil certain user requirements. It has been expected that an ITS interface should promote ease of use, possess an effective screen design, ensure smooth flow of task-oriented activities and be able to embrace a variety of interaction styles. But emphasis has been placed on the ITS interfaces to also reduce the cognitive load of the students so that students can focus their time and efforts to solving problems rather than trying to understand the interface.

### 2.2 ASPIRE

Authoring systems reduce the complexities and cost involved in the development of ITSs. Numerous authoring tools have been developed for various domains and tasks [Murray, 2003]. Demonstr8 [Blessing, 1997] is an authoring system that supports the development of model-tracing tutors in the area of arithmetic. Disciple [Tecuci & Keeling, 1999] is an authoring system for developing intelligent educational agents. Disciple also plays an additional role of observing and rectifying behaviour of educational agents. Other authoring systems include Diligent [Angros et al., 2002], an authoring system used in simulated and agent-based learning systems and Cognitive Tutor Authoring Tools (CTAT) [Aleven et al., 2006], an authoring system for developing model-tracing tutors.

ASPIRE [Mitrovic et al., 2009] is both an authoring system and a deployment platform for constraint-based tutors (CBT) [Mitrovic et al., 2007]. The goals of ASPIRE are to reduce the time and complexity required to develop CBTs. ASPIRE architecture includes ASPIRE Author, the authoring server where ITSs can be developed, and ASPIRE Tutor, the tutoring server from where developed ITSs can be deployed and accessed from. Figure 2.2 illustrates the architecture of ASPIRE Author, including all the components needed for the ITS authoring process.

Even though developing an ITS with the support of ASPIRE is a process that is less complex than developing ITSs from scratch, it still requires human expertise in a wide range of skills. *Domain Structure Modeller* requires the author to have an expert understanding of the domain and the pedagogical strategy that the domain is best taught with. As ASPIRE focuses on problem-based learning, the author is also

expected to have complete knowledge of all the concepts that are to be taught in a domain and the problem-solving actions required for those tasks in the domain. The author must plan and decide if the domain requires problem solving to have procedural steps or non-procedural solution. In the case of procedural domains, the author needs to specify each problem-solving step.

ASPIRE provides an *Ontology Workspace* for the author to describe all the concepts contributing to the solution structure using ontologies [Suraweera et al., 2004]. Concepts are linked using a one-to-one or a one-to-many relationship in a hierarchical structure. Identifying all the important concepts and the relationships between them in the domain ontology is a very important step in the authoring process as ASPIRE generates syntax constraints based on the ontology created.

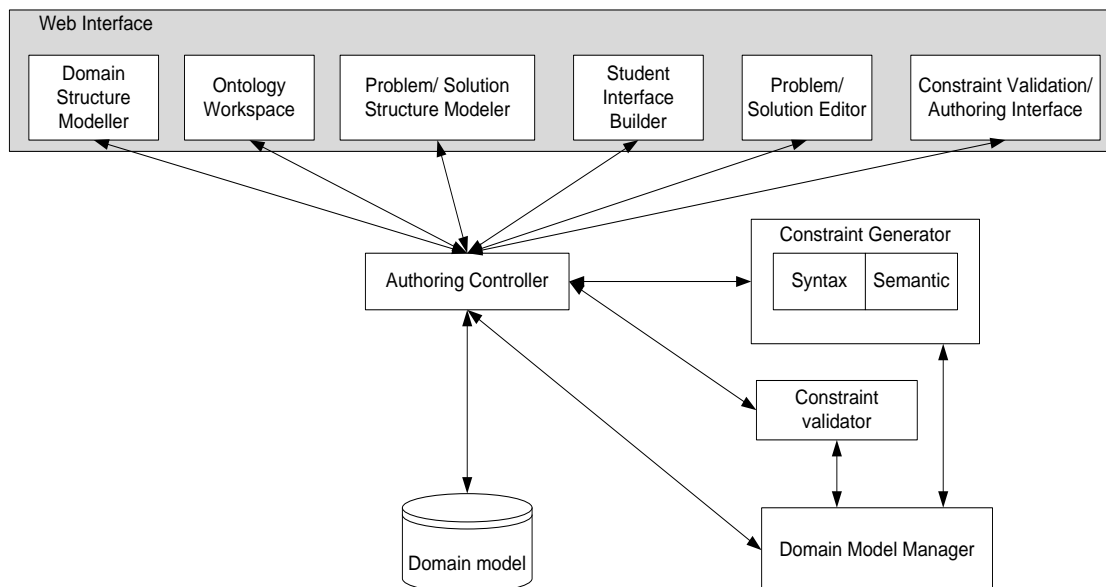


Figure 2.2: Architecture of ASPIRE author with components included [Mitrovic, 2009]

*Problem/Solution Structure Modeller* provides an environment for the author to construct the problem solving structure and ideal solutions for the problems. The author creates additional problems and their solutions in the *Problem Editor* and views the default html interface created in ASPIRE at the *Student Interface Builder*. ASPIRE gives the option for the author to develop a java applet interface to replace the default html interface.

Constraints are generated in the Constraint Generator component. Syntax constraints are generated from the domain ontologies created earlier and semantic constraints are generated based on the ideal solutions specified for the problems by the author [Suraweera et al., 2004]. Each constraint specifies two conditions: relevance and satisfaction. ASPIRE Tutor checks every student's submission of a solution against the constraints. The solution is checked to identify whether the constraint is relevant and if the constraint is found to be relevant, ASPIRE then checks if the student solution satisfies or violates the constraint. When a student's solution violates a constraint, the student receives a feedback message that helps him/her to correct the mistakes and find the correct solution. The author can provide and customise feedback messages that students see. The Authoring Controller component manages all the components and the communication between all the components of ASPIRE-Author. It also handles the communication between ASPIRE-Author and ASPIRE-Tutor and also ASPIRE-Author and the human author.

Different parts of ASPIRE have been evaluated in a number of previous projects. The effectiveness of generating constraints from ontologies was studied in WETAS [Suraweera et al., 2004]. In this study an interface was created for WETAS.

The purpose of the interface was to allow students to visualise all the concepts of ER (Entity Relationship) by creating a domain ontology. All the concepts were linked in a hierarchical manner. Students were also allowed to view the concepts as constraints at the same time they were drawing the ontologies. The knowledge of the students in building ontologies can be described as novice level but out of the 20 constraints, the novice students managed to generate 15 constraints accurately through the creation of domain ontologies and two students managed to generate 20 constraints accurately. Results of the study indicated that ontologies did assist in constraint acquisition.

Past research has shown that constructing constraint bases manually consumed a lot of time and computational effort. SQL-Tutor [Mitrovic, 1998] consists of more than 700 constraints. Each constraint required about 1.1 hour to be produced by Mitrovic, a knowledge engineering and domain expert. A group of novice graduate students were asked to develop an ITS for the domain of fraction addition with WETAS, a precursor to ASPIRE using the constraint authoring system (CAS). The students had to develop all the domain dependent components required by CAS to generate constraints; including a domain ontology, problems and solutions. Results of the study showed that the students took an average time of 1.3 hours to produce each constraint. The time taken by the students is very close to the time taken by Mitrovic. We need to understand that the students are novices in knowledge engineering and constraint authoring and therefore their achievement in producing constraints in a short time through WETAS can be considered as a significant achievement [Suraweera et al., 2005, 2007].



The constraint generation method used in ASPIRE was evaluated through studies carried out on two CBT domains, KERMIT [Suraweera et al., 2002, 2004] and NORMIT [Mitrovic, 2002]. Constraints generated by ASPIRE for KERMIT and NORMIT were compared to the original constraints developed manually for the domains. Results of the study showed that ASPIRE was able to produce complete constraint sets for the domain with the accuracy level of between 90% and 100%. Some of the constraints that ASPIRE failed to produce were covered by the other constraints generated for the domain by ASPIRE [Suraweera et al., 2005 2010].

CIT (Capital Investment Tutor) [Mitrovic et al., 2008] was an ITS developed in ASPIRE to teach capital investment decision making to university students. An evaluation study of CIT was conducted with a group of students enrolled in ACCT102, an Accounting and Finance for Business course at Lincoln University, New Zealand. The pre-test and post-test results of the study were favourable with students showing that their knowledge of capital investment decision making domain had improved after using CIT. For our research we developed DM-Tutor in ASPIRE. We present details of how we developed DM-Tutor in Chapter 4.

### **2.3 ITSS INTEGRATED WITH EXISTING SYSTEMS OR TRAINING TOOLS**

In this section we describe research related to the past attempts of embedding ITSs with existing systems. We describe the embedded systems and the techniques used in

their development. We also describe their purpose and functionalities. Lastly, we look at the benefits and possible limitations of these systems.

### 2.3.1 LUMIERE/ EXCEL PROJECT

Lumiere /Excel prototype [Horvitz et al., 1998] was developed to demonstrate the potential of embedding a Bayesian user model into a Microsoft Office application. A group of researchers conducted a study on the ability of experts in guessing the users' needs and actions. Participants with different expertise levels in Microsoft Excel usage were given a set of tasks to be completed. They were informed that an experimental system will be tracking their actions so as to assist them when they need help. They were told that assistance would appear in the computer next to the one they are using. The participants were unaware that the experts were observing their keyboard and mouse actions through a key hole of the computer's interface. Participants were also asked to verbalise their thoughts while they were interacting with the system. The experts observing the participants tried to guess the participants goals by their reactions and provided them advice whenever they thought that the participants needed it.

A Bayesian user model was developed after identifying users' needs and actions through user studies. A special version of Microsoft Excel was developed to observe user mouse and keyboard actions. The Bayesian user model was then embedded into Excel. Lumiere event language was developed to capture the stream of events caused by users' actions within the system. Users' actions were monitored at both the low level operational actions and also semantic actions. The system

continues to monitor users' actions and even pauses of inactivity to try to infer if user requires help while interacting with the system.

Several observations resulted from the study. The researchers concluded that experts could not be certain of users' goals and therefore were not sure of the type of advice users' needed. When experts gave wrong advice, users had acted on the advice given and continued on a wrong path. User errors and poor advice from experts results in a pattern of work inefficiency. The researchers believe that experts would become better at guessing users' needs with more experience.

The researchers followed a similar approach and developed *Office Assistant* for Microsoft Office '97. Office Assistant uses broader but shallower user models with thousands of user goals in each application. However the user model used is not permanent and does not include considerations of user competency. Office assistant only focuses on a small set of user actions and does not attempt to consider a combination of multiple users' actions.

### 2.3.2 AN AUTOMATED CONSULTANT FOR MACSYMA

MACSYMA Advisor [Genesereth, 1979] was developed as an automated consultant for the algebraic manipulation system, MACSYMA. MACSYMA Advisor was a separate program from MACSYMA with its own separate database and domain knowledge. The objective of building Advisor on top of MACSYMA was to create an online consultant that could provide an interactive environment with intelligent feedback for MACSYMA users. Results and analyses obtained from consultations

with users showed that users required five different classes of information from MACSYMA:

1. Directions: “How do I ...?”, this was called a *HOWDO* need.
2. Factual information: “What are the arguments of ...?”, this was called a *WHAT* need.
3. Facts verification: “Is it the case that...?” this was called an *IS* need.
4. Method description: “How does MACSYMA ..?”, this was called a *HOW* need.
5. Behaviour explanation: “Why did this ..?”, this was a *WHY* need.

Since MACSYMA Advisor only helped the user in her/his problem-solving attempts rather than to teach them how to solve problems, understanding the user's approach well was critical for providing comments or corrections to users. Therefore, much effort was directed towards designing a comprehensive user model and using it to interpret user's actions. In the Advisor, each specific module handled a different type of question. The first and fifth types of questions were complex, as they required a variety of answers to match user's need. When a user requested help in such cases, Advisor needed to give a form of feedback that reflected the system's broader understanding of the user's requirement than merely the problem they were facing at the current step they were in. To answer the first and fifth type of questions MACSYMA Advisor developed a plan (“goal-subgoal tree”) of the user's expectations and worked towards *debugging* the plan. Advisor attempted to identify user's similar error patterns, and then communicate with the user in a question and

answer session to confirm if the assumptions Advisor had made were correct. Once that was confirmed, corrective action was taken.

In contrast to the usual error messages and online help systems, MACSYMA Advisor tried to act as an intelligent consultant that was able to interpret user's actual intentions and to provide appropriate advice whenever possible. At the time of implementation Advisor was considered successful, as it was able to reconstruct plans in the simple cases with which it was presented; however, there were a lot of doubts on how general Advisor could be, because of the difficulty of completely identifying user's needs and requirements even within a single domain like MACSYMA. Even though most of the modules in MACSYMA were implemented, the Advisor was only an experimental system that lacked an interface and never reached the point of being made available to MACSYMA users.

### **2.3.3 TRAINING WHEELS IN A USER INTERFACE**

Novices learning to use any new software make mistakes because they are not familiar with the functionalities of the software. The mistakes they make could prevent them from completing the tasks they had set out to do. They become frustrated, demotivated and sometimes stop trying to learn the new software. A group of researchers observed novices trying to learn basic functions of a stand-alone commercial word processing system [Carrol & Carrithers, 1984]. Next, they developed a list of common errors that new users make. As they aimed to create a training environment for the novices learning how to use the word processing system,

they modified the system's interface to create a *training wheels interface* [Carrol & Carrithers, 1984] that disabled functions that caused users to make errors. *The training wheels interface* only allowed users to carry out a range of functions that were not complex or error-prone. Their hypothesis was that when these error prone activities were blocked, users' frustration or confusion during the learning period can be reduced.

A study was conducted with a group of twelve novices using a commercial word processing system. Six of the novices (control group) were provided with a self-study manual and were informed that they would need to learn how to use the standard word processing system to type a letter as quickly as possible. The other six participants (experimental group) were also provided the same self-study manual and given the same task of typing a letter. The experimental group used the word processing software with the *training wheels interface*.

Results showed that the experimental group participants using the training wheels interface were significantly faster in completing their tasks compared to the control group. Four out of the six participants in the experimental group managed to type and print the letter within the allocated time. In contrast, only two out of the six participants in the control group succeeded in typing and printing out the letter. The experimental group learned better than the control group and were happier to work with the software using the *training wheels interface*. They were also happier with themselves and with their work. From the results, we can conclude that a training wheels interface provides novices with a exploratory environment while at the same time reducing the frustration caused by the complexities of a new software.

### 2.3.4 PLUG-IN TUTOR AGENTS

Plug-In tutor agents [Ritter & Koedinger, 1996] are tutors that are integrated within software applications at the workplace and educational environment and may not have all features that a normal model- tracing ITS would. Their goals are to reduce the cost and time needed to develop ITSs by using the features of the existing software application to provide an effective learning environment to learners. Two cognitive tutors, Geometer Sketchpad Tutor and Excel Tutor were developed and embedded into Geometer Sketchpad and Excel 4.0 for Macintosh respectively. Figure 2.3 presents the general architecture for plug-in tutor agents that were used in the development of both Geometer Sketchpad Tutor and Excel Tutor.

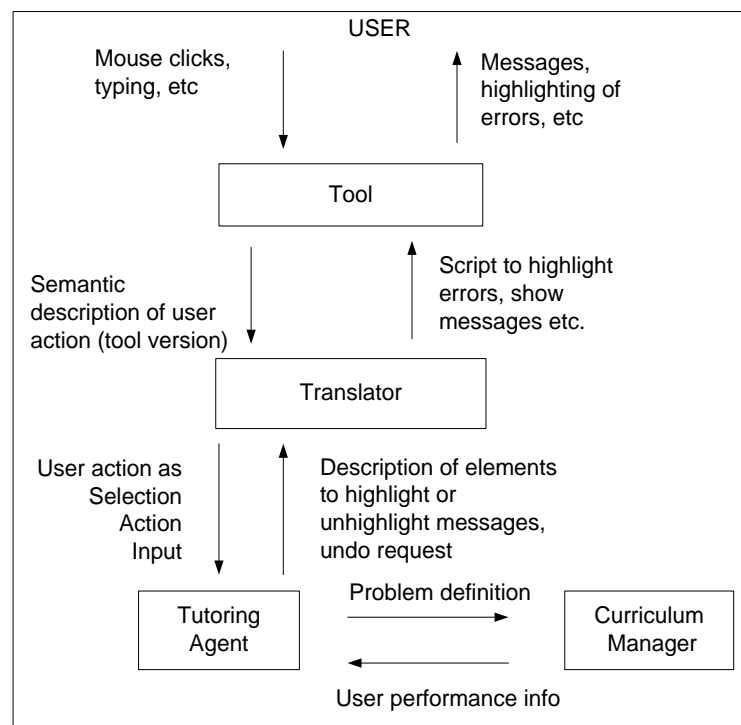


Figure 2.3: General architecture for plug-in tutor agents [Ritter & Koedinger, 1996]

Plug-in tutor agents need to fulfil several important criteria. The first important requirement is that problem solving must be carried out using the tool. Secondly, the application software's core features should not be changed by the tutoring system's presence. Lastly, the tutoring agent must monitor the user's problem solving activities within the system. In order for tutoring to be effective, the plug-in tutor agent needs to know the student's activities on the interface at a more semantically meaningful level compared to the physical, operating system level.

### **2.3.4.1 TOOL-TUTOR FOR GEOMETRIC CONSTRUCTION USING GEOMETER SKETCHPAD**

---

The Geometer Sketchpad is a software tool used for creating and analysing geometric diagrams. A tutor was developed and integrated into Geometer Sketchpad to allow students to create geometric diagrams using Sketchpad as the student's workspace. For the student, the system appeared similar to Geometer Sketchpad application. Several modifications were made to convert Sketchpad into a tutor; a tutor menu to access the tutor was added and an additional window was created to provide hints and feedback to the students. As Geometer Sketchpad was not originally designed to allow for communication with the user, it had to be modified to inform the tutoring agent each time the student requested a hint or feedback. Sketchpad used AppleEvents to monitor student's problem solving activities within the system. AppleEvents carried out a message based interprocess communication mechanism in Mac OS.



#### **2.3.4.2. A TOOL TUTOR FOR ALGEBRA PROBLEM SOLVING USING MICROSOFT EXCEL**

---

Excel Algebra Problem Solving Tutor was created ‘on top’ of Excel 4.0 for Macintosh. A user interface that looked similar to Excel and a tutor menu was created to allow students to access the tutor. When students started the system, they saw an Excel worksheet with a problem statement displayed within it. All the menu selections and students problem-solving activities on the user interface were sent to the tutor through a series of AppleEvents. Communication between Excel and the tutoring agent was carried out through AppleScript. If the user submitted an incorrect answer, a feedback message was passed from the tutoring agent through the translator and was displayed in the Excel interface.

#### **2.3.4.3 GEOMETER SKETCHPAD AND EXCEL TUTOR AS LEARNING ENVIRONMENTS**

---

Plug-in tutor agents have many potential benefits in creating flexible learning both in the education and workplace environments. Cost and time taken to develop ITSs could be lowered by using existing software applications to develop ITSs. This research created the initial architecture and identified the components necessary for plug-in tutor agents. The architecture described was more suitable for model tracing tutors. The studies conducted on Excel Tutor [Mathan & Koedinger, 2002] showed that Excel Tutor provided high learning gains for the students. Geometer Sketchpad

Tutor was not evaluated with students; therefore we could not ascertain its effectiveness as a teaching tool.

### **2.3.5 EMBEDDED TRAINING FOR COMPLEX INFORMATION SYSTEM**

Intelligent embedded training systems (ETS) have been considered a successful approach to providing training using real life working scenarios. Trainees practice solving problems on the main application with guidance and feedback from the training system. Trainees are trained to use a particular system for a period of time using ETS and after the training period is over, ETS still remained within the application system to provide training when the users wanted to enhance their knowledge.

ETS's objective was to provide basic operator training for novices on how to operate the devices they need to use to carry out their work. In this research an ETS was developed and embedded into a complex information-analysis tool used in a military intelligence application [Cheikes et al., 1998]. ETS used computer-based training modules to present learning concepts and ICAI (Intelligent Computer Assisted Instruction) system to provide training exercises either on a simulation of the application or the actual application itself.

The ETS observes user's UI actions on the application's interface such as menu selection, button clicking and other UI actions within the software and; report them to the external application. It also reports changes in the software caused by user's actions. The ETS is also expected to execute commands from an external application

if the user initiates those actions. ETS uses software *hooks* to carry out the above mentioned functions in the external application. ETS builds and maintains a model of the trainee's problem solving activities in the external application. The user model is used to provide guidance and feedback to the user. ETS provides hints to the users that guide them to the next problem solving step. Figure 2.4 illustrates the architecture of ETS and the components of the system.

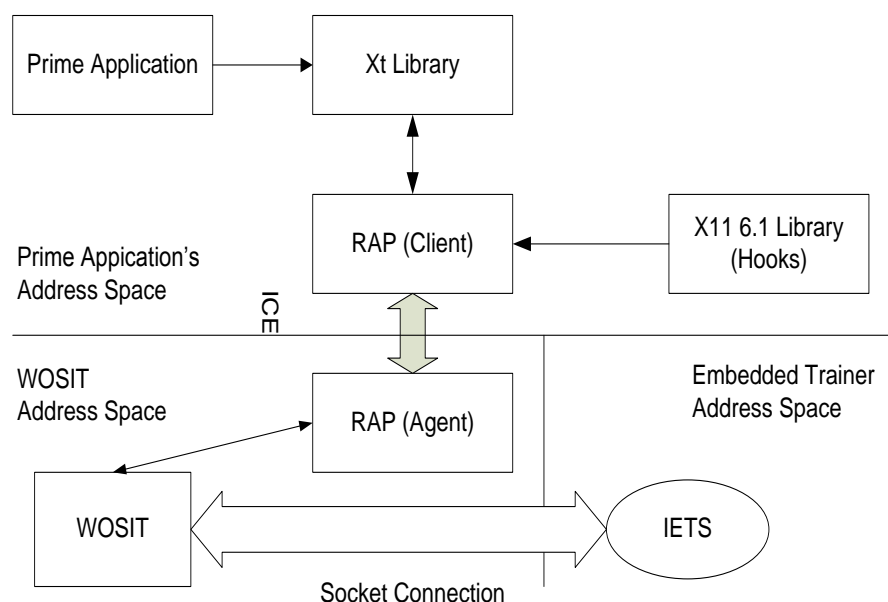


Figure 2.4: Architecture of ETS with the integration components included [Cheikes et al., 1998]

ETS used *WOSIT* (Widget Observation, Scripting and Inspection Tool), a Unix application that acted as an intermediary between the main application and a target application. An ETS needed to be able to monitor the prime application and *WOSIT* provided the required access into the application without having to change the functionalities of the main application. *WOSIT* helped to modify the application's user interface by adding a tutor menu to the main application which made it seem that the

ETS was fully integrated with the main application. Each time *WOSIT* noticed any changes in the user interface, it sent a message through the network socket to ETS. ETS also consisted of an *Inter Client Exchange* (ICE) library that enabled data to be sent and received in a more efficient manner, X-Windows hooks to manage widget-level changes in the application and a *Remote Access Protocol* (RAP) to send information from application to the ETS using network sockets.

ETS was not evaluated with participants as its developers felt that as it could only provide hints as feedback, it was not suitable or mature enough to be evaluated with participants. However, through informal evaluations they were able to identify several limitations in the system. They felt that the ETS's training service was not capable of handling persistent tracking of students during practice exercises. They also felt that the current implementation needed to be improved to handle a variety of student's actions including cancelling work midway when they made a mistake, changing problem solving goals multiple times during the course of the training, wandering off to explore the system and other unexpected student behaviour.

### **2.3.6 INTEGRATING ITSS AND HEALTH INFORMATION SYSTEMS**

Various ITSS have been developed for training in the health education domain. The goals of these systems were to provide effective learning environments for students without increasing health risks to the patients. Health care professionals constantly need to access Health Information Systems (HISs) for data about records, symptoms, diagnosis and other relevant information about patients. Unfortunately, HISs are

heterogeneous and geographically distributed making the clinical information they contain usually scattered.

A cognitive tutoring system, ITS-CBR [Gonzalez et al., 2007] was developed using case-based reasoning (CBR) methodology and Multi-Agent Systems. ITS-CBR included knowledge of the medical domain, student learning behaviour and teaching strategies that were adapted to the student's individual learning requirements. Using CBR methodology, an ideal solution was constructed using solved cases and previous student experiences. Originally, ITS-CBR did not consider sharing data among the different HIS. An additional component, *HL7* was added to the existing system architecture to integrate and share patient's clinical data between ITS-CBR and HIS. *HL7*, an international standard messaging protocol was used to communicate between ITS-CBR and the HIS.

The objective of sharing patients' clinical data between ITS-CBR and HIS was to provide training to students using real live cases and patients' actual medical condition. Through this integration, it was expected that the students will be able to control their learning while exploring real life complex situation. Students collaborate on identifying cases and issues, analyse cases, make hypotheses and also on attempting to find solutions to the problems. Figure 2.5 presents the architecture and components of ITS-CBR and the integration with the HIS.

The integration component consisted of the *HL7 message server* and an *event manager* that allowed messages to be created and parsed, a *transformer* that linked *HL7* messages together and a *message sender* that sent *HL7* messages to a receiving system. ITS-CBR also included a student model component that contained the

knowledge students were required to learn about the domain. The MAS-CBR component handled the case-based reasoning methodology through a *retrieval phase* where the system searched for similar solved cases and a *reuse phase* that altered solutions obtained at the earlier retrieval phase to present an individualised solution to the student. The domain knowledge acquired through the domain expert or from other systems is placed in the knowledge database.

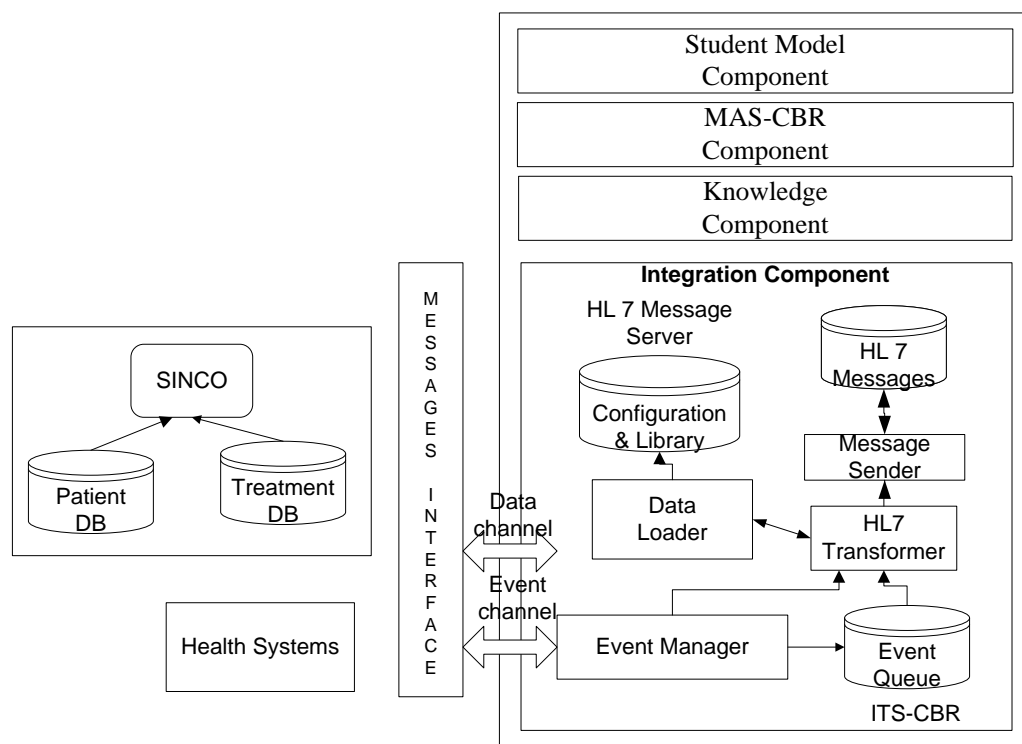


Figure 2.5: ITS-CBR architecture and integration component [Gonzalez et al., 2007]

For this research, STIM-Tutor (ITS-CBR) was integrated with the SINCO-TB database that contained information about patients with tuberculosis; this information then became part of the tutor's knowledge base. Through *HL7*, information was distributed between HISs. When an *HL7* event notification was received by the

system, the message was placed in a suitable queue and sent out immediately using a message ID. When a query was received by ITS-CBR, a message was sent out to request for records in SINCO-TB that matched the query. SINCO-TB then responded with demographic information about the patients. The next message requested patients health conditions and received a response that included the patient's complete health report including symptoms, health problems and diagnosis.

Integrating patients' information between ITSs and Health Information Systems could have a lot of potential benefits especially in the training of new health care students and exposing them to complex and real life situations. However, the integration of ITS-CBR and HL7 was not evaluated with actual users or through any other empirical evaluations.

### **2.3.7 PERSONAL ACCESS TUTOR (PAT)**

PAT [Risco & Reye, 2009] is an ITS that helps students create forms and reports in MS Access. PAT's architecture was designed and implemented to run within MS Access. Figure 2.6 presents the architecture and main components of PAT. PAT consists of a domain model, user model, instructional expert and a user interface. Domain model contains knowledge on all the objects and properties required for forms and reports creation in MS Access. User model represents student's knowledge of the domain. It also includes students' personal characteristics, learning styles, learning preferences, their favourite subjects and favourite topics in subjects. The instructional expert contains pedagogical information. It checks the student's solution

and provides hints to guide the students. The user interface allows two way communication, user requests for exercises from the system and the system provides feedback to the user within the MS Access window.

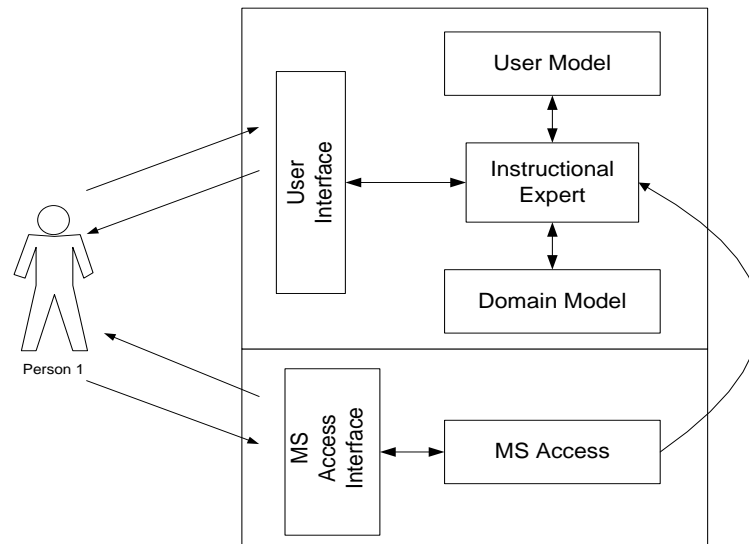


Figure 2.5: Architecture and main components of PAT [Risco & Rey, 2009]

Students accessed PAT through the MS Access interface as PAT was implemented as an add-on to MS Access. Students attempt exercises, and submit solutions to PAT. Their solutions were compared to the ideal solutions stored in PAT's knowledge base. Based on the errors in their solutions, they receive several levels of feedback messages from PAT starting from; "*is my solution correct?*", which informs the student of the error, progressing to; "*how do I fix this?*" and "*what is wrong with my solution?*" which gives the student more specific hints on how to solve the error.

The authors stated that PAT was evaluated by a group of students enrolled in a Database Systems course at QUT (Queensland University of Technology). Questionnaires were used to evaluate the students' responses. Out of the number of



students who participated in the survey, 59% felt that having PAT integrated in MS Access was a useful feature. 70% of the students said they liked PAT and in addition, 64% of the students said that they looked forward to using ITSs for other courses as well. However, questionnaires alone could not be considered sufficient to evaluate the learning gains from using an ITS. The evaluation study results would have been more conclusive if the knowledge students gained in MS Access was tested before using PAT (pre-test ) and after using PAT (post-test). Through that analysis, we would then be able to understand how much knowledge students gained through using PAT.

### **2.3.8 xPST (EXTENSIBLE PROBLEM-SPECIFIC TUTOR)**

The creation of xPST [Gilbert et al., 2009] was influenced from the earlier efforts of Ritter and Kodinger's idea of developing plug-in tutor agents. xPST was overlaid onto two existing live systems; Paint.NET, a standalone .NET application for Windows Paint.NET, a stand-alone .NET application for Windows and CAPE, a Web-based Authoring Tool. xPST was a model tracing system and was developed in a similar manner and architecture as plug-in tutor agents.

The researchers aimed to investigate the effectiveness of their interaction design through the development of both tutors. As the ITS environment and application environment is similar, they feel that the skills learned in the ITS environment can be easily transferred to the actual application. xPST is not deeply integrated into the two external applications, the integration is not permanent as well. The tutor is only present during training and not available during regular use of the applications. The researchers followed certain principles for xPST development, they did not disable or

block any components in the external software, they aimed to interrupt the learner only when necessary and give feedback in small doses. Figure 2.7 presents the overall architecture of xPST.

xPST components included a *cognitive model* that contained the learning objects of the domain, the tasks student needed to complete within the domain and rules that controlled the feedback provided for students as they use the systems. Every element of the third-party software interface was mapped to a learning object found in the cognitive model. The *Event Mapper* “eavesdropped” on every user activity in the system and sent these activities to the xPST tutoring Engine, which checked students solution attempts against the cognitive model. Based on the correctness of the solution, relevant feedback is provided back to the student and displayed in the software system’s interface.

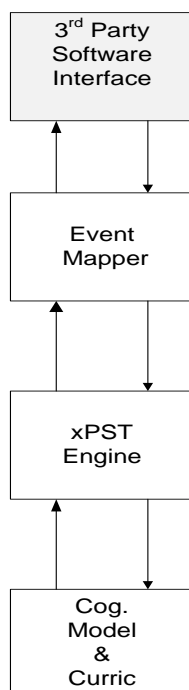


Figure 2.6: Architecture of xPST system [Gilbert et al., 2009]

The third-party software interface represented any stand alone software or application websites. Various methods were used to observe user's activities in the system: through widgets, accessibility hooks and even through low level OS events. xPST Engine existed in its own separate server and communicated with the other components using TCP/IP, allowing the tutored software and the tutor to run on different servers. xPST aimed to provide the same kind of tutoring support provided by human tutors and therefore had a more casual appearance compared to normal tutors. xPST also placed importance in balancing student's exploration of the application software with required interruption for guidance.

xPST faced three design challenges during development: 1) how to embed a problem scenario into the confines of the existing software, 2) how to provide relevant feedback to students while they are attempting problems in the system and 3) how to let students explore the application software's interface while at the same time ensuring that they complete the tasks allocated to them. A major interaction design challenge faced by the researchers was to decide how tightly to control the user while they interact with the system. As the tasks within xPST have multiple successful completion paths, the feedback messages are designed to provide cognitive-based instructions. Users are not only informed what to do next but also the conceptual model of the software. xPST tracks the users actions across tasks to monitor their progress while they interact with the system.

In both Paint.NET and CAPE, a tutor menu was added to the application and students could use the menus to select tasks from the curriculum. Task descriptions and additional information related to the task were displayed in different areas of the

student interface. Just-in-time (JIT) error messages were provided as feedback each time students made mistakes. In both applications, students were allowed to explore the software but were restricted from carrying out any step that could lead them to an irreparable and irreversible condition. When CAPE was tested with users, one user felt that it was too restrictive and thought it was more suitable for novices. To our knowledge the integration of xPST and Paint.NET was not evaluated.

### **2.3.9 SBT-AID (SCENARIO-BASED TRAINING: ADAPTIVE, INTELLIGENT, DYNAMIC) TRAINING SYSTEM**

SBT has been a successful approach for training on dynamic and complex real-world scenarios. Even in areas that ITSs such as situated tutors have been found to be unsuitable, SBT has been considered effective. However, SBT is expensive, its development time-consuming, and it required support from well trained instructors to conduct the training. SBT-AID [Shatz et al., 2009] is the automation of SBT and has tutoring components integrated within it. SBT-AID was developed and integrated into an existing military training simulation software to provide adaptive training using multiple scenarios for many personnel at the same time. Figure 2.8 presents the architecture and high-level design of the SBT-AID prototype.

SBT-AID's training can be broken down to several phases: identify task requirements, create user profiles, select and apply training goals and plans, present various training scenarios, assess trainees during and after training, record trainees performance and post- task activities and finally, record meta data of the training and history of trainees

performance. The objective and focus of SBT-AID was not described through this work. The benefits of the integration were not supported by any evaluation studies as evaluation studies were not conducted for SBT-AID.

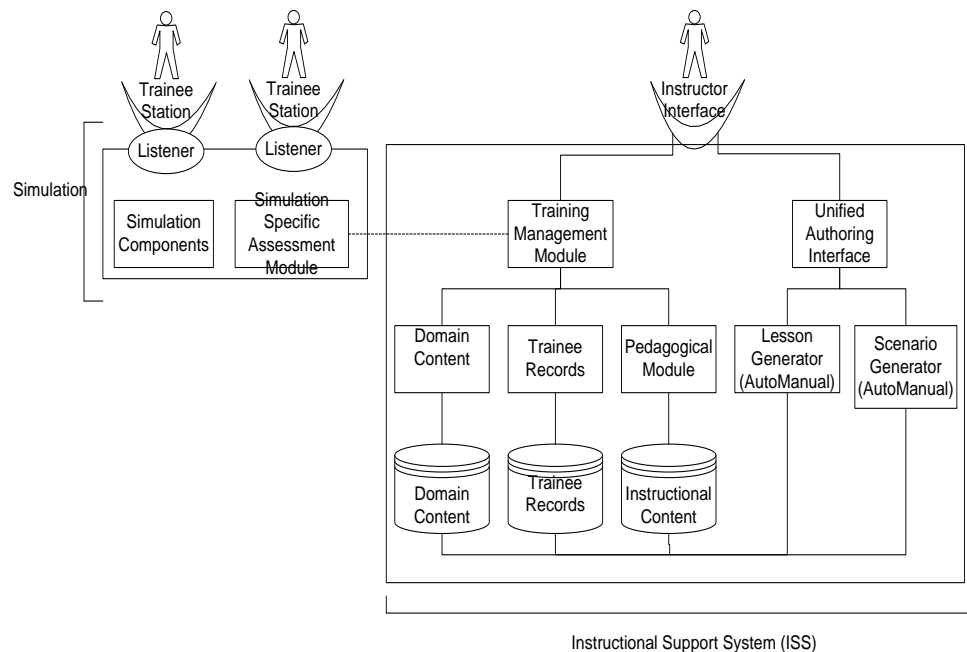


Figure 2.7: Architecture of SBT-AID [Shatz et al., 2009]

## 2.4 SUMMARY OF EMBEDDED ITSS

Geometer Sketchpad Tutor and Excel Algebraic Tutor [Ritter & Koedinger, 1996] monitor all user's actions. ETS [Cheikes et al., 1998] that was embedded into CIS also monitored all of the user's actions. The objective of monitoring the actions was to guide the user's actions in achieving correct solutions. These tutors expected the student to follow only the designated solution path. xPST [Gilbert et al., 2009] monitored all of the actions of users in Paint.NET and CAPE but allowed the user to

follow any one of the multiple solution paths to arrive at the correct solution. Geometer Sketchpad Tutor, Excel Algebraic Tutor, ETS and xPST were all designed and developed following Ritter and Koedinger's plug-in tutor approach. The systems described observe all of the user's actions in the application's interface including mouse actions, menu selections and button clicks. Except for xPST, the tutors described above have a permanent and deep integration with the external applications. xPST is only available during training but not during regular use of Paint.NET and CAPE.

Training wheels interface [Carrol & Carrithers, 1984] blocked and disabled complex functions in a word processing software to provide a training environment to novice users. Users were only provided with functions required to carry out their tasks. PAT [Risco & Reye, 2009] was designed and implemented to run within MS Access. Students learn to create reports and forms in PAT without leaving the MS Access window. Students' actions were diagnosed by the instructional expert component in PAT. They receive feedback and hints from the instructional expert. PAT retains a model of the student's problem solving actions but does not monitor the user's actions in MS Access.

In conclusion, this section discussed the different approaches to integrating ITSs into systems. The systems described here are tightly or deeply integrated into the external applications. In this approach, all of the actions in the system's user interface including keyboard typing and mouse clicks in the external application are monitored by the ITS and reported to the external applications. Some of the systems described in this section block or disallow users from accessing functions and actions that

cannot be monitored by the ITS. In our research, we will use the minimal integration approach as we feel that it is not necessary to monitor all of the user's actions in the external application other than actions related to student's learning. As we will be embedding our ITS into a workplace system that is used on daily basis by employees of multiple levels of expertise, we will not be blocking functions or components of the workplace system.

# CHAPTER 3

## DOMAIN OF PALM OIL

One of the main factors contributing to the success and failure of oil palm plantations management is effective decision making. In the early days, plantation management dealt with relatively smaller-scale operations in less complex environments and estate managers ran plantations using their work experience and skills. As the palm oil industry rapidly emerged as a global industry and many plantations are being managed remotely, making effective decisions in plantation management becomes difficult but more crucial.

The goal of plantation management is to optimise available human, financial and technical resources to maximise yield and profit. Plantation or estate management has to ensure careful planning and proper allocation of all of these resources as unplanned or misdirected resources consumption and poor internal financial control can cause a failure or a major disruption in the plantation's operations. Many oil palm



industries past and present have used a variety of plantation management software to manage and control their plantation operations [Tan, 2003].

In this chapter we will describe MIS [Ravan, 2007] and the plantation decision making environment. MIS is a comprehensive system containing information and reports used by managers to analyse yield, fertilizer and costs operations in the estates and plantation. We will describe Yield Analysis, Upkeep and Cultivation, Harvesting and Collection reports and how they are used in plantation analyses. In our research MIS for palm oil will be embedded with DM-Tutor to provide training on plantation decision making.

### **3.1 MANAGEMENT INFORMATION SYSTEM (MIS) FOR PALM OIL**

Oil palm plantations constantly need to be able to answer two questions: how they are performing, and what they should be doing to improve their performance. The MIS for oil palm plantations [Ravan, 2007] answers those questions by providing reports and analyses that help managers to know their performance level; and planning and budgeting that guide management on the actions they could take to improve their performance. MIS for palm oil includes modules for estate cost functions such as material inventory, vehicle management, human resource management and accounts management. At present, the MIS for palm oil is being used to manage several oil palm plantation operations in Malaysia and Indonesia.

## 3.2 MIS DEVELOPMENT ENVIRONMENT

MIS for palm oil was developed using the integrated software environment Visual Studio version 2005 under the ASP.NET framework and runs primarily on Microsoft Windows. The .NET framework provides MIS a flexible development environment where programs could be executed locally, remotely or executed locally and distributed via the internet. A Microsoft SQL Server provides the relational database containing all the plantation operation information the MIS requires. MIS uses Crystal Reports, a reporting tool integrated with Microsoft Visual Studio to provide a variety of reports, including *yield*, *harvesting*, *material consumption* and other reports required to manage the plantations.

## 3.3 REPORTS AND ANALYSES

The MIS for palm oil is web-based and is accessible for the user to log-in via the internet. When the user logs into the system they see a user interface that is divided into several panes as shown in Figure 3.1. The left panel contains the menu where users can select the kind of report they need to access to do their analysis: the user has the option of analysing information via maps, graphs and reports. On the right panel, the user is given a view of all the plantations available in the MIS. The user is required to select the estate or plantation location and the period of the analysis they wish to perform. When the user selects the type of analysis he/she wishes to carry out a window opens in the right panel showing the available plantation locations.

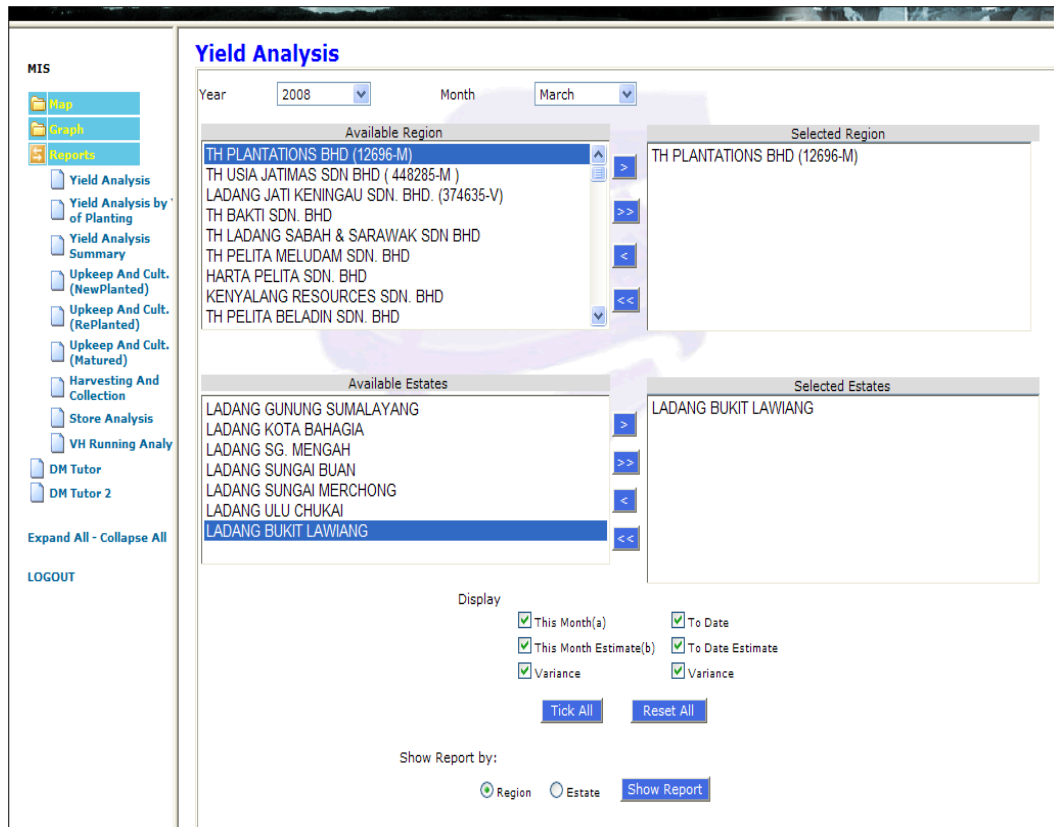


Figure 3.1 Screen-shot of the MIS interface viewed when users first log into the system.

Generally, all of the MIS reports are accessed by selecting the analysis required, the plantation’s name, report year and month. *Yield Analysis* reports provide information about the yield produced by a plantation location, size of the plantation and the amount of rainfall experienced by that location in the particular month. The *Upkeep and Cultivation* reports describe all the activities that take place during the various stages of the oil palm tree cultivation from the immature plant stage to the mature palm tree stage including fertilizer consumption, transportation vehicle usage, labour activities and other plantation cost related activities. *Harvesting and Collection* reports describe the costs involved in the harvesting activities for each plantation location including labour, vehicle and number of FFB (fresh-fruit-bunches) of palm oil that are harvested and transported from oil palm plantations to the refinery mills.

When users select the analysis report they wish to view, another window opens up in the MIS interface showing the detailed report of the plantation relevant to the analysis chosen. Figure 3.2 shows a screen-shot of the detailed yield report for estate Ladang Bukit Lawiang for the month of March 2008.

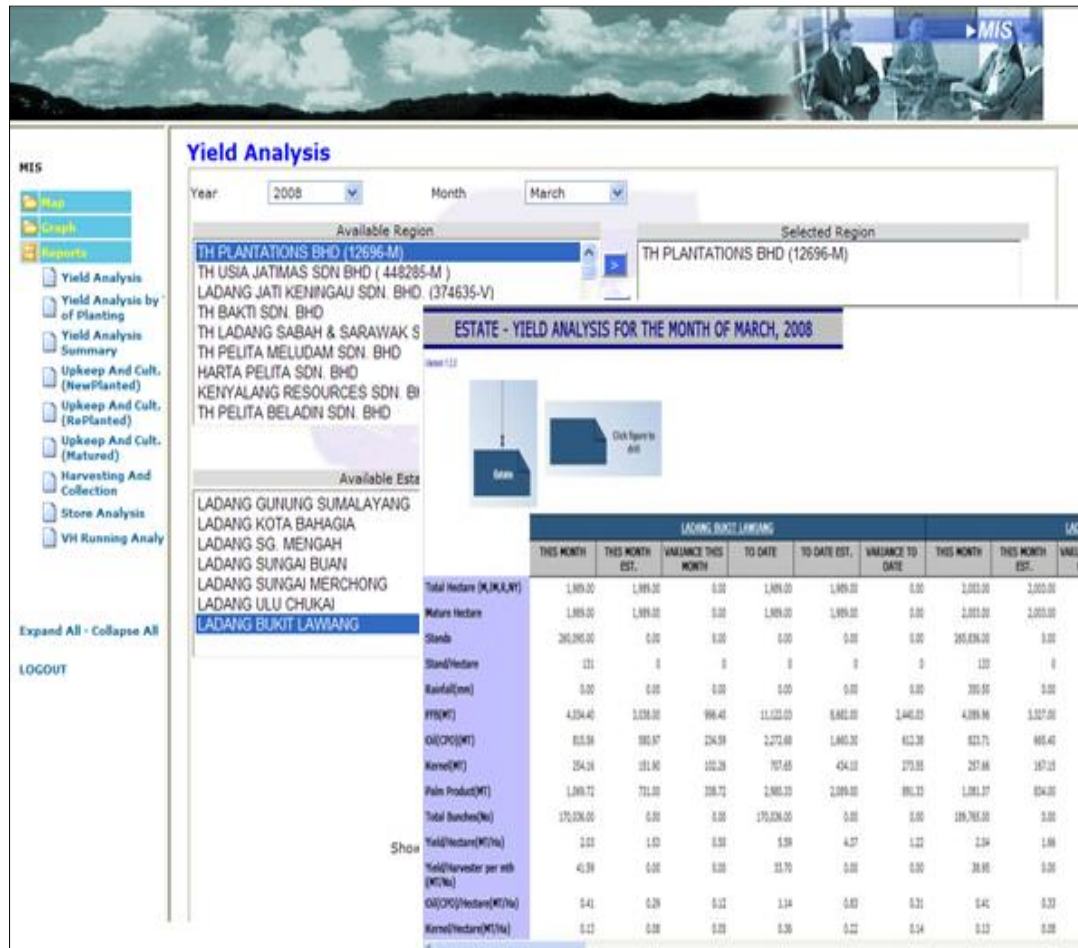


Figure 3.2 Screen-shot of the MIS interface showing detailed yield analysis report in a separate window.

To make a plantation analysis for a given period of time, the manager has to look at all the relevant decision parameters including plantation size, number of trees, number of fruit bunches harvested and even amount of rain received. To decide whether to carry out soil fertilization activities for a particular area, the manager has to look at the amount of rain forecasted for the plantation location and the growth

stage of the palm trees in the particular block of estate. To carry out the yield analysis, the manager has to compare the yields of different plantation locations, to identify potential yield and the plantation location that is not performing well. To forecast yield, a manager has to know the current yield and other plantation conditions.

We describe below the walk-through for decision-making processes and task analyses for MIS problem solving. An oil palm plantation manager doing yield gap analysis in the MIS carries out the following steps:

1. Select Yield Analysis from the list of analyses on the MIS menu.
2. Identify the plantation locations (estate, field or block)
3. Select the time period of the analysis (month and year).
4. Click the 'show report' button to view the yield analysis report. Another window appears on the MIS interface to present detailed yield analysis report as shown in Figure 3.2. In the report, Yield is given as FFB (Fresh Fruit Bunch) in the report.
5. Compare yield values for the plantation locations where yield gap analysis needs to be done.
6. Estimate yield gap analysis for the given plantation location based on past yield records and identify the location that requires immediate management attention.

A manager doing yield forecasting analysis in the MIS carries out the following steps:

1. Select Yield Analysis from the list of analysis on the MIS menu.
2. Identify the plantation locations (estate, field or block)
3. Select the time period of the analysis (month and year).
4. Click the 'show report' button to view the detailed yield analysis report. A window opens up in the MIS interface to present the detailed yield analysis report

as shown in Figure 3.2 Current yield value (month and year) for the plantation is stated in the report.

5. The manager calculates the age of the palm tree in the plantation location and estimates the plantation maturity stage. Based on past yield performance (obtained from yield records), the manager forecasts yield for the required future time period.

A manager doing fertilizer management analysis in the MIS carries out the following steps:

1. Select Upkeep and Cultivation from the list of analysis on the MIS menu. The manager also needs to select if she/he wishes to do fertilizer analysis for a plantation that is matured, replanted or newly planted.
2. Identify the plantation locations (estate, field or block)
3. Select the time period of the analysis (month and year).
4. Click the 'show report' button to view the detailed upkeep and cultivation report.

Another window opens up showing the detailed material consumption for the particular plantation location as presented in Figure 3.3. Figure 3.3 shows the amount of fertilizer used in Estate Ladang Bukit Lawiang. The report also includes the number of trees planted in the estate and the year of planting. This is given as FFB (Fresh Fruit Bunch). Based on the information provided in the report, the manager estimates the fertilizer requirement and the efficiency of fertilizer usage in the plantation location.

MATERIAL CONSUMPTION DETAIL						
BLOCK - UPKEEP & CULTIVATION - MATERIAL CONSUMPTION DETAIL FOR THE MONTH OF APRIL, 2006						
Estate	:	LADANG BUKIT LAWIANG				
Field	:	82A				
Block	:	1				
Planted Year	:	1982				
Mature Planted Area	:	219.00				
Total Stand	:	14652				
Stand/Hect	:	67				
Expenditure Code	:	PM0303 - MANURING - MATERIAL				
Material Cost (RM)	:	24,928.00				
Material Desc	Unit	Qty	Qty/Stand	Cost	Cost/Stand	Cost/Ha
MIX THP2	50KG/Bag	760.00	0.05	24,928.00	1.70	113.83
				<b>24,928.00</b>		

Figure 3.3: Screen shot of material consumption report

### 3.4 MIS CHALLENGES

MIS provides comprehensive information of the entire plantation’s operations. From the walk-through of the MIS tasks analyses, it is clear that an experienced and knowledgeable estate manager would know how to utilize the information contained in the MIS to effectively analyse and make decisions. An experienced manager is able to plan the consumption of materials (fertilizer and insecticide), human, technical and finance resources to ensure the plantation’s continued sustainability and improved yield performance. The manager can also use the MIS to budget, forecast and plan for future plantation growth. However, new or inexperienced plantation managers lack this knowledge and the oil palm expertise required to make effective plantation

decision making through the MIS and could end up making many inaccurate decisions that result in disruptions to the plantation's operations and also financial loss.



# CHAPTER 4

## DESIGN AND DEVELOPMENT OF DM-TUTOR

### 4.1 DESIGN OF DM-TUTOR

The design of DM-Tutor involves many aspects. DM-Tutor's objective is not to provide teaching of fundamental oil palm domain concepts or on how to use the MIS as our expectation is that DM-Tutor users would already have some background knowledge on the oil palm domain and information systems. The goal of DM-Tutor is to provide scenario-based training on oil palm plantation decision making based on the actual operational data from the MIS. The approach used in the development of DM-Tutor had to be suitable to the planned system integration with the MIS. Users will be seeing both the MIS and DM-Tutor at the same time while they do problem solving in DM-Tutor. The design of the problems also needs to be closely related to how the MIS provides data to the users. Since the goal of DM-Tutor is to provide

scenario-based training using operational data and reports from the MIS, what DM-Tutor teaches and the order of decision making has to be carefully planned. The focus on skill set of the audience has to be taken into consideration as well. The nature and format of the problems was designed after detailed discussions with domain experts in oil palm plantation management and decision making. The problems in DM-Tutor are designed to increase in complexity and required knowledge level with each problem step.

Plantation decision making itself is a complex domain and involves the usage of various types of skills. In order to make good plantation management decisions, managers must first access the correct information (reports) from the MIS. Next they will need to perform the analyses by using the correct formulas required for the problems. Lastly they will need to plan on how to improve a particular condition in a given plantation location.

### **4.2 PROBLEM DESIGN**

DM-Tutor focuses on analyses that require solutions specific to a particular plantation location and the problems faced in that location. The student's skills of understanding the problems and providing correct decisions with the appropriate and effective usage of the reports and screens from the MIS would be evaluated.

As the goal of DM-Tutor is to help people make better management decisions, DM-Tutor focuses on the three main areas of plantation decision making: yield gap analysis, yield forecasting analysis and fertilizer management analysis. In a normal plantation management environment, an estate manager would need to know the current yield condition of his plantation, current fertilizer schedules and requirement

and the potential yield she/ he could forecast based on the current plantation yield and condition. DM-Tutor addresses all of these requirements through the three tasks taught in DM-Tutor.

### 4.2.1 YIELD GAP ANALYSIS

In oil palm plantations, yield of a particular year is always affected by climatic and plantation conditions of past production periods. Potential yield [Weng, 1999] ( $Y_p$ ) can be predicted through past yield records from several production months or years. Potential yield is what the management hopes to ideally achieve in an estate or plantation area. The difference between actual yield ( $Y_a$ ) and potential yield ( $Y_p$ ) is an important criterion for making the decisions if management action could be taken to improve yield of a particular field or if replanting of trees needs to be considered in the particular location. An estate achieves greatest return to the inputs of management staff when attention is focused on fields where the yield gap is greatest.

For this task, the student is expected to access various Yield Analysis reports from the MIS and to practice on the three analyses to be applied at various fields/estates:

1. Estimate  $Y_p$  for a production area (Estate, Field)
2. Calculate yield gap using the formula; yield gap=  $Y_p - Y_a$  [Weng, 1999]
3. Identify field/ estate with the highest yield gap
4. Recommend a management action to improve yield in the area with the highest yield gap.

### 4.2.2 FERTILIZER MANAGEMENT ANALYSIS

Fertilizers are usually the largest variable cost item in oil palm plantations. Improvements in palm oil nutrition are also required to improve site yield potential. In order to have a balanced fertilizer program, potential needs of the crop at various stages of growth must be determined. The most suitable nutrient combination and the appropriate placement of the fertilizer relevant to the crop/palm age needs to be identified as well.

For this task, DM-Tutor expects the student to access Yield Analysis reports and Material Consumption reports from the MIS. The student is required to attempt the analyses given below:

1. Calculate Partial Factor Productivity (PFP) [Fairhurst, 2003] to identify how much yield is produced for each kg of fertilizer nutrient using the formula:

$$\text{PFP} = \text{Bunch yield (kg/ha)} / \text{Fertiliser nutrient (FN) (kg)}$$

2. Calculate Agronomic Efficiency (AE) [Fairhurst, 2003] to identify how much additional yield is produced for each kg of fertilizer nutrient applied using the formula:

$$\text{AE} = (\text{Bunch yield +FN} - \text{Bunch yield 0FN}) / \text{Fertiliser Nutrient (FN)}$$

kg.

3. Given various fields/ estates with different palm growth stages the student has to identify the correct type of nutrient combinations and accurate fertilizer placement.

4. Given various plantation conditions that reduce the efficiency of the fertilizer/ nutrient applied, the student has to provide management actions to resolve the problems.

### **4.2.3 YIELD FORECASTING ANALYSIS**

Numerous attempts have been made to devise a reliable yield prediction method. From counting palm fruit bunches to using econometric regression, various methods have been used to forecast yield for both short and long term [Weng, 1999]. To do yield forecasting analysis in DM-Tutor, the student is expected to access yield analysis reports for different periods of times and also access various information from the reports. The student would need to attempt the following analyses:

1. For a given estate/field within the MIS, calculate yield forecast for a future period based on the current yield of the plantation area.

$\text{Yield forecast} = \text{current yield (MT/Ha)} * \text{future plantation size (Ha)}$   
[Weng, 1999].

2. Compare the calculated yield forecast to the actual yield for the new period and analyse the difference between them. For a given low production period of an estate/ field, students will be required to analyse conditions that caused low yield production. Students will be evaluated on the accuracy and efficiency of accessing the relevant reports and information from the MIS to support her/ his answer. For each problem condition, the user has to identify a management action to improve yield in the given plantation location.

### 4.3 COMPONENTS OF DM-TUTOR

DM-Tutor consists of a student model, pedagogical module, interface module, a database of problems and solutions and constraints that represent domain knowledge.

Figure 4.1 presents the overview of the components in DM-Tutor.

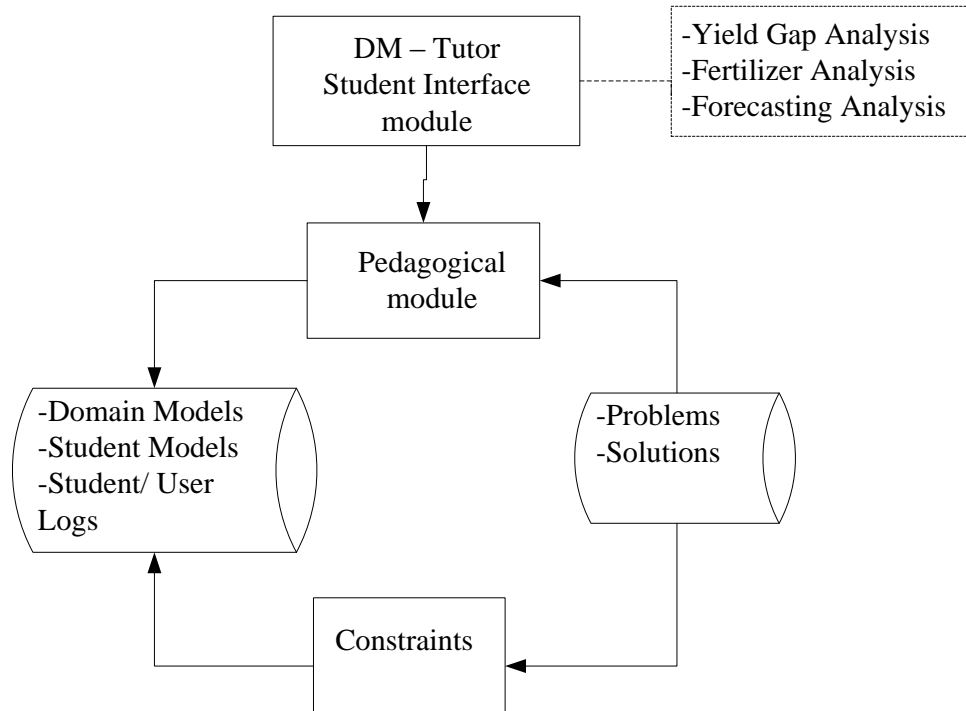


Figure 4.1: Components of DM-Tutor

The student model contains information about student’s knowledge and is updated every time the student uses DM-Tutor. Constraint-based modelling (CBM) (reference) is used to model the domain and the student. The student’s solution is matched to the constraints that are relevant to the task the student is attempting. This is done in order to identify any mistakes in the student’s solution. The relevant constraints and their satisfaction conditions determine whether they have been satisfied or violated. The lists of relevant, satisfied and violated constraints serve as a

short-term student model, which is then used to update the long-term model of the student's understanding of the particular task in DM-Tutor.

The pedagogical module provides all the tasks and problems in DM-Tutor in a procedural manner. It selects instructional actions relevant to the scenario-based problem solving strategy used. It also has the role of providing feedback messages to users when they submit an incorrect solution. For DM-Tutor feedback messages are categorized to levels of increasing help and information to enable students to arrive at the correct solution, starting at: *correct/ incorrect*, *error flag*, *hint* and continuing to *detailed hint*, *all errors* and *full solution*. *Correct/ incorrect* message informs the students if their answer is correct or incorrect. An *error flag* points out to the students which part of the student's solution is wrong. A *hint* provides some information about the error in the student's solution. *Detailed hint* provides more descriptive information of the error to help students understand their error better. If they were still unable to find the correct solution at this stage, the *all errors* message gives the students a list of their solutions that contains errors. Students can request increasing amounts of feedback messages but they are prevented from seeing the full solution until they make three attempts at answering the given question.

The interface module presents the student interface of DM-Tutor. The interface module is responsible for displaying the problems to students. It also provides a solution workspace area for the students to work on their solutions to the problems. When students submit solutions, their solutions will be compared to the ideal solutions in the problems and solutions database. The student interface also displays feedback appropriate to the accuracy of the student's solution. The Problems and Solutions component focuses on the three tasks of DM-Tutor: *yield gap analysis*,

*fertilizer management* and *yield forecasting*, the three main analyses for palm oil plantation management. When students submit a solution, their solution is compared to the ideal solution in ASPIRE [Mitrovic et al., 2009] and appropriate feedback is given to the students.

### 4.4 DEVELOPMENT

DM-Tutor is a constraint-based tutor (CBT). It was developed using ASPIRE, an authoring system and deployment environment for CBTs. ASPIRE supports the development of CBTs by automating a number of the functions involved. The stages involved in developing DM-Tutor are described below.

#### 4.4.1 CHARACTERISTICS OF THE DOMAIN

The first step to developing DM-Tutor using ASPIRE is to identify the characteristics of the oil palm plantation decision-making domain and the most suitable pedagogical approach for teaching this domain. We modularized DM-Tutor into three tasks: *yield gap analysis*, *fertilizer analysis* and *yield forecasting*. Figure 4.2 shows the screenshot of Domain Details for the *yield gap analysis task*. From Figure 4.2 we can observe that, for yield gap analysis task, DM-Tutor first requests students to obtain yield values for all fields of the estate. Students would have to access relevant yield analysis reports from the MIS and input yield value of all the fields into DM-Tutor's interface. Next, DM-Tutor asks students to identify potential yield of the estate and select the correct formula to calculate yield gap for each field of the estate. Lastly



DM-Tutor requires students to identify the field with the highest yield gap and give a suggestion on how to improve yield in that location.

We specified all problem-solving steps and provided additional information to provide clear task descriptions and problem specific instructions for each task. Through ASPIRE we specified the procedural layout of the problems and identified *input yield value* and *calculate yield gap* as repeatable tasks as students need to repeat these steps multiple times for the multiple plantation areas in an estate or plantation location.

The screenshot shows the 'Domain Details' configuration page for 'DM-Tutor'. It includes a navigation menu at the top with options like Home, Domain, Ontology, Problem Structure, Student Interface, Problem Editor, Syntax Constraints, Semantic Constraints, Domain functions, and Test Cor. The main content area is divided into sections: 'Domain Details' with fields for Name (DM-Tutor), Description (Decision Making Tutor), and Type (Procedural selected); 'Procedural Steps' with a table of 5 steps; and 'Problem Sets' with a table of 1 set. A 'Save structure' button is at the bottom.

Domain Details							
Name	DM-Tutor		Backup and Restore				
Description	Decision Making Tutor						
Type	<input type="radio"/> Non-Procedural <input checked="" type="radio"/> Procedural						
Procedural Steps							
	Name	Task Description	New Page	Page no	Problem Specific Instruction	Repeatable	
<input type="checkbox"/>	1 Report Selected	Report Type, Report Name, Report Year	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	2 Location, Yield value	Input Field number or Estate name, Inpu	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	3 Yield Gap formula, Potential y	Select formula for yield gap, Input Potent	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	4 Location, Calculate yield gap	Input Field number or Estate name, Calc	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	5 Highest yield gap, Select recor	Highest yield gap, Select recommendatic	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Problem Sets							
	Name	Description	Scaffolding				
<input type="checkbox"/>	1 Yield Gap Analysis	Yield gap analysis					
Save structure							

Figure 4.2: Specifying the domain details for the yield gap analysis task

#### 4.4.2 ONTOLOGY DEVELOPMENT

In the second stage of DM-Tutor development, we created domain ontologies for each of the tasks in DM-Tutor using the ontology workspace in ASPIRE. Developing domain ontologies is crucial for the authoring process of CBTs because it provides the

necessary information for the generation of syntax constraints. In the domain ontologies we described important concepts for each task in DM-Tutor and the relationship between these concepts using a hierarchical structure. Each concept can have super-concepts, sub-concepts, properties and relationships with other concepts. A concept may be related to another concept through an 'is-a' relationship and this is depicted by the arrows between concepts in the ontology workspace.

Through the creation of ontologies we were able to study the domain in greater detail and this enabled ASPIRE to organize constraints in a meaningful way to develop more accurate constraint bases [Mitrovic et al., 2009]. A preliminary study conducted to evaluate the role of ontologies in manually composing a constraint base showed that constructing domain ontologies assisted the composition of constraints [Suraweera et al., 2004]. The study also showed that ontologies helped the authors to reflect accurately on the domain, organise constraints into meaningful categories and produce more complete constraint bases.

Figure 4.3 presents the ontology for the yield gap analysis task developed in ASPIRE. On top of the ontology workspace, there is a bar that contains a set of drawing tools. The rectangle and arrow tools are used to draw the concepts and the *is-a* relationship between them. The empty page tool allows the user to create a new ontology and the trash can tool allows the user to delete the selected element in the ontology, undoing/redoining tool, the diskette tool that allows us to save the ontology and the finish tool (represented as a flag), that allows us to save and exit the ontology. In the ontology workspace below the tool bar, we develop the domain ontologies represented by super concepts, concepts and relationships between them. In a clockwise order, it can be observed that the super concept Formula,

Recommendations, *Report Type* and *Yield* have various sub-concepts each as well. For DM-Tutor all the super concepts and sub-concepts form part of the solution structure.

For example, as a part of the yield gap analysis task, DM-Tutor requires students to select the correct formula to calculate yield gap value. From the ontology in Figure 4.3 we can observe that the sub-concepts *CPO*, *YieldHa*, *Forecast*, *AE*, *PPF*, *YieldGap* all relate to the super-concept of *Formula*. These concepts relate directly to the solution part of formula selection problem in DM-Tutor. The concepts *Land clearing*, *Replanting*, *Pest and Disease management*, *Improve soil drainage*, *Improve harvesting technique*, *Improve fertilizer management*, *Selective thinning*, *Legume cover plants* relate to the super concept *Recommendations*.

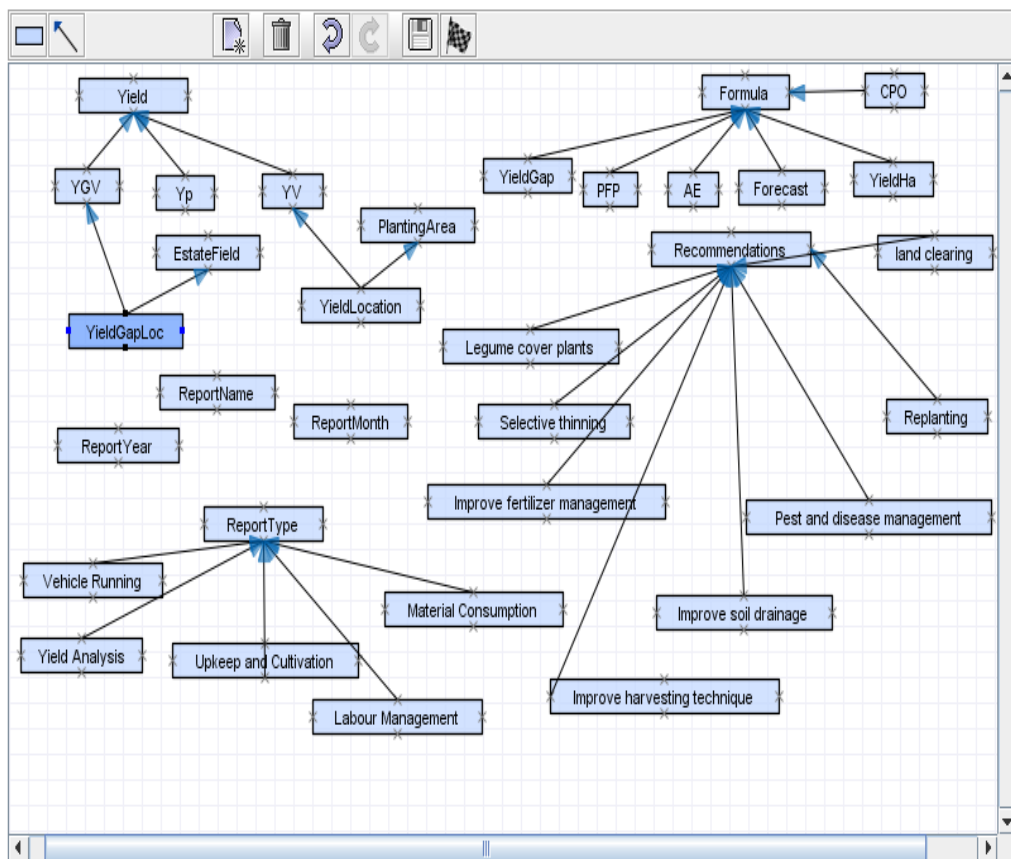


Figure 4.3: Ontology for the yield gap analysis

The concepts *Material Consumption*, *Labour Management*, *Upkeep and Cultivation*, *Yield Analysis* and *Vehicle Running* relate to the super concept *ReportType* representing solution components for report type selected. The super concepts *ReportName*, *ReportYear* and *ReportMonth* relate to the report information part of the solutions component. The concepts *YieldLocation* and *YieldGapLoc* relate to the plantation locations in the solution structure where students list the estate names or field numbers in the *yield gap analysis* task. *YieldLocation* is related to the super concept *YV* (yield value) and *PlantingArea*. The concept *YieldGapLoc* is related to the super concepts *YGV* (yield gap value) and *EstateField*.

### 4.4.3 PROBLEM AND SOLUTION STRUCTURE

DM-Tutor consists of procedural tasks, and each problem-solving step might require several solution components. We have specified the problem steps for each task when we developed the domain details in stage 1 (Figure 4.2). As mentioned earlier in the previous section, each solution component comes from the ontology for each task. Figure 4.4 presents the solution structure for the *Yield Gap Analysis* task in DM-Tutor.

From Figure 4.4, it can be observed that the *Yield Gap Analysis* task consist of several solution components; *Report Selected*, *Location and Yield value*, *Yield Gap formula*, *Location and Calculate yield gap* and lastly, *Highest yield gap and Select recommendations*. The first column gives the ‘label’ for the particular task. For the *Report Selected* solution component, the students have to identify the report type, name, year and month of the report as part of their solution structure. The middle

column labelled as *Concept* includes all the concepts from the domain ontology (Figure 4.3) that are relevant for the solution components.

The screenshot displays the 'Solution structure' interface, which is organized into five distinct sections, each with a gray header bar. Each section contains a table with four columns: 'Label', 'Concept', 'Element Count', and 'Free Text'. The 'Label' column includes a text input field and a selection checkbox. The 'Concept' column features a dropdown menu with a 'Choose item...' prompt and '+' and '-' buttons. The 'Element Count' column has a dropdown menu, and the 'Free Text' column has a checkbox.

Report Selected: Solution Components			
Label	Concept	Element Count	Free Text
<input type="checkbox"/> Report_Type	Yield Analysis,Upkeep and Cultivation,Material Consumption,Vehicle Running,Labour Management	Exactly 1	<input type="checkbox"/>
<input type="checkbox"/> Report_Name	PlantingArea	Exactly 1	<input type="checkbox"/>
<input type="checkbox"/> Report_Month	ReportMonth	Exactly 1	<input type="checkbox"/>
<input type="checkbox"/> Report_Year	ReportYear	Exactly 1	<input type="checkbox"/>
Location, Yield value: Solution Components			
Label	Concept	Element Count	Free Text
<input type="checkbox"/> PlantingArea <input type="checkbox"/> selection	YieldLocation	At least 1	<input type="checkbox"/>
Yield Gap formula, Potential yield : Solution Components			
Label	Concept	Element Count	Free Text
<input type="checkbox"/> Formula <input type="checkbox"/> selection	YieldGap,PFP,AE,Forecast,CPO	Exactly 1	<input type="checkbox"/>
<input type="checkbox"/> PotentialYield <input type="checkbox"/> selection	Yp	Exactly 1	<input type="checkbox"/>
Location, Calculate yield gap: Solution Components			
Label	Concept	Element Count	Free Text
<input type="checkbox"/> EstateField <input type="checkbox"/> selection	YieldGapLoc	At least 1	<input type="checkbox"/>
Highest yield gap, Select recommendation: Solution Components			
Label	Concept	Element Count	Free Text
<input type="checkbox"/> FieldOrEstate <input type="checkbox"/> selection	PlantingArea	Exactly 1	<input type="checkbox"/>
<input type="checkbox"/> Recommendation <input type="checkbox"/> selection	SelectiveThinning,Improve Fertilizer Management,Improve soil drainage,Legume Cover Plants,Pest and Disease Management,Pruning,Replanting,Improve harvesting techniques	Exactly 1	<input type="checkbox"/>

Figure 4.4: Solution structure for yield gap analysis task

Every row in Figure 4.4 corresponds to related concepts in the ontology in Figure 4.3, for example, the first row includes Yield Analysis, Upkeep and Cultivation, Material Consumption, Vehicle Running and Labour Management. These concepts can be found in the domain ontology in Figure 4.3. The solution structure in Figure 4.4 describes the steps the students would need to follow to submit their solutions. As DM Tutor is a procedural tutor, students' solutions will be checked after each solution component, indicated by the gray bar in Figure 4.4. In the yield gap analysis task, students have to first identify the particular report from the MIS that is

relevant to the problem in DM-Tutor. Next they will need to list the yield locations, identify potential yield and calculate yield gap. Lastly, they will need to identify field with highest yield gap and give recommendation to improve yield in that location.

### 4.4.4 STUDENT INTERFACE

DM-Tutor's student interface is divided into four parts. Figure 4.5 shows DM-Tutor's interface generated by ASPIRE based on the problem/solution structure and domain ontology that we created. The bar located at the top part of the interface contains several buttons that provide controls for selecting problems, obtaining help, and changing or leaving the ITS.

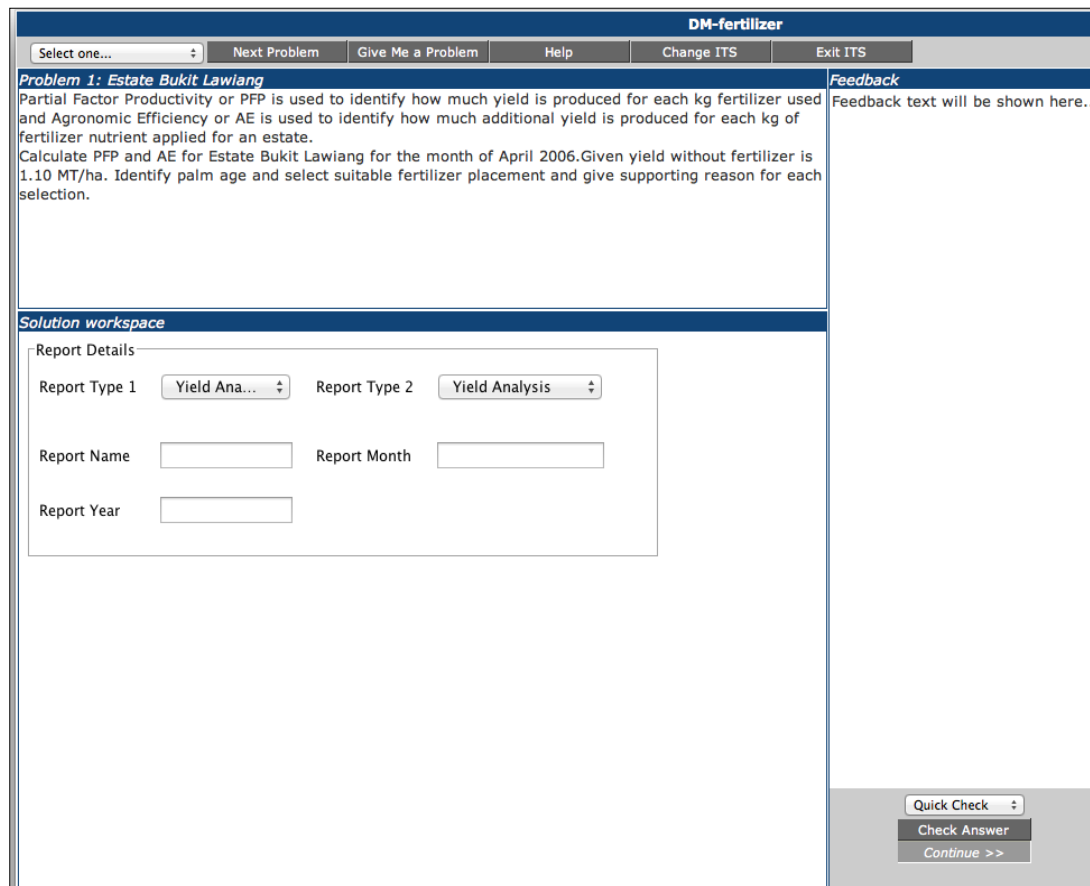


Figure 4.5: Student interface of DM-Tutor

Students can select problems by selecting the problem number they wish to attempt. When they click the 'next problem' button, DM-Tutor gives them the next problem within the same task. The 'Help' button provides help on how to use the ITS. The 'Change ITS' button brings users back to the menu where users can select to attempt other tasks in DM-Tutor including Fertilizer Analysis and Forecasting Analysis. The 'Exit ITS' button allows the users to exit the current problem or task they are attempting.

Below the control bar is the pane where the problem statement is placed so that students always know the problem they are attempting. The bottom pane presents the solution workspace where students need to work on their solutions to the problem and the side pane is used to provide feedback to the students on the problems they are attempting. At the bottom of the Feedback pane, there are several buttons; users use the 'Quick Check' button is used for users to request increasing feedback. DM-Tutor users need to click the 'Check Answer' button to check the accuracy of the solution they submitted.

DM-Tutor will check student's answer in each step and give appropriate feedback. If the student's solution for any step is incorrect, DM-Tutor will give increasing levels of feedback that would help guide the student to the correct solution before allowing her/him to proceed to the next step of problem solving. The 'Continue' button is only active if the student submits a correct solution to a current problem step. The interface design for DM-Tutor is aimed at reducing memory load of the students. Currently DM-Tutor's interface is developed using java applets for every page of each task. DM-Tutor's tasks require students to access various reports and analyses of the MIS. This is necessary as the information required for solving

problems in DM-Tutor comes from the MIS. Each problem-solving step in DM-Tutor appears on a separate page. The students will be seeing the MIS and DM-Tutor at the same time while they carry on their problem solving activities. We describe the interface of DM-Tutor embedded with the MIS in Chapter 5.

### 4.4.5 ADDING PROBLEMS AND SOLUTIONS

Next, we created problems and ideal solutions for *Yield Gap analysis*, *Fertilizer Management analysis* and *Yield Forecasting analysis tasks*. The interface provided by ASPIRE for creating problems and solutions is similar to the default student interface. We will discuss the problem and solution creations in two stages. Figure 4.6 shows the first part of the problem editor page where we create a problem for fertilizer management task. The problem editor allows us to add, edit or delete a problem. The problem number is generated by ASPIRE and is automatically updated every time we create an additional problem. We specified the problem name, difficulty level and problem-specific instructions for each problem that we created. Figure 4.6 describes the *Fertilizer management* problem for Estate Ladang Ulu Chukai. The problem statement describes the situation of the problem and the actions the students are required to do to solve the problem. Next we added an ideal solution for the problem. ASPIRE allows us to specify one or more solutions to a problem.



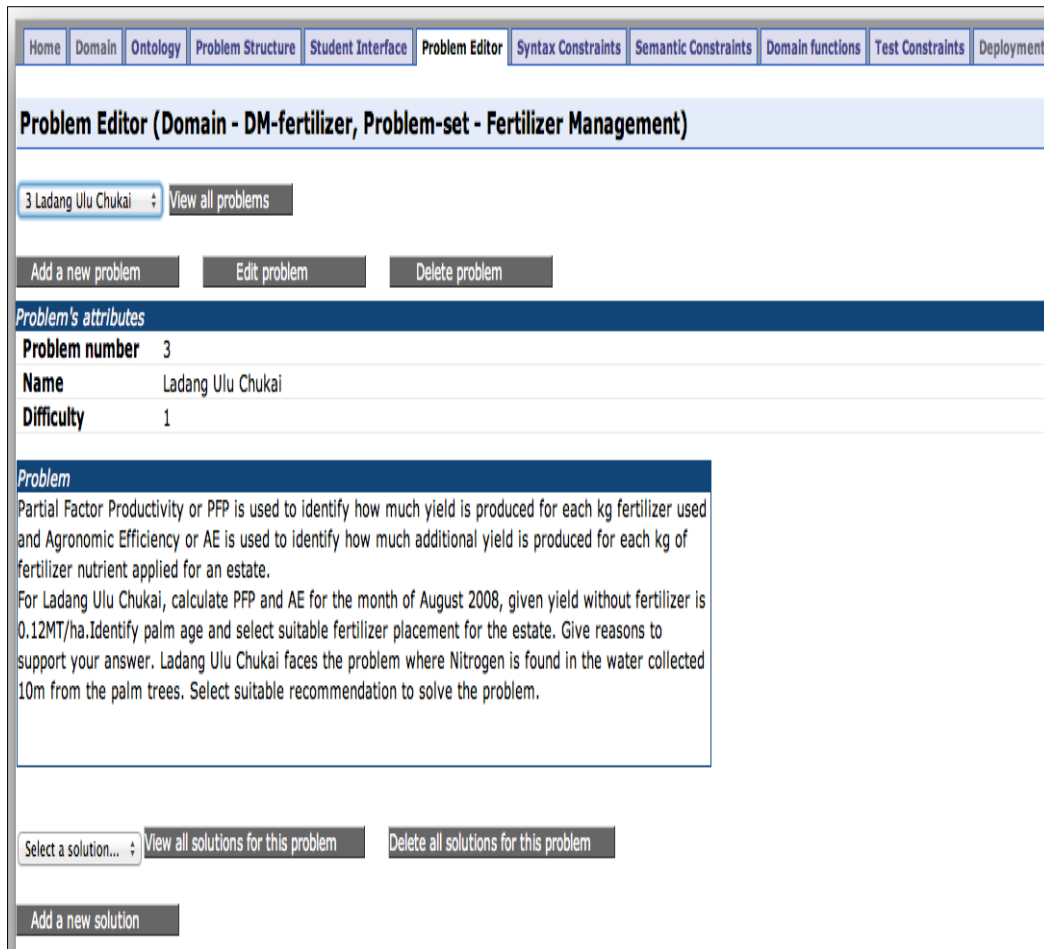


Figure 4.6: Creating problems for fertilizer management task

As DM-Tutor is procedural-based, we would need to specify a solution for each problem step and the student's solution will be checked at every step as well. Figure 4.7 presents the second part of the problem editor page that describes how a solution is created. The solution component for each problem step is based on the problem and solution structure created earlier in step three of DM-Tutor's development. The solution components added here becomes the ideal solution for the created problem. Semantic constraints are generated later using the ideal solutions we specified for each problem.

1	<a href="#">View all solutions for this problem</a>	<a href="#">Delete all solutions for this problem</a>
<a href="#">Add a new solution</a>		<a href="#">Delete solution</a>
<b>Solution's attributes</b>		
<b>Solution number</b>	1	
<b>Name/notes</b>	<input type="text"/>	
<b>Solution</b>		
to help you answer the questions.		
<b>Report1</b>	Type Material Consumpt	Select correct formula for PFP and calculate PFP value Select correct formula for AE and calculate AE value
<b>Report2</b>	Type Yield Analysis	
<b>Name</b>	ReportName Ladang Ulu Chukai	
<b>Report_Year</b>	ReportYear 2008	
<b>Report_Month</b>	ReportMonth August	
<b>PFP formula, PFP value</b> <i>Select formula for PFP, Calculate PFP</i>		
<b>PFP_formula</b>	Type PFP	
<b>PFP</b>	PFP 0.31	
<b>AE formula, AE value</b> <i>Select formula for AE, Calculate AE</i>		
<b>AE_formula</b>	Type AE	
<b>AE</b>	AE value 0.00	
<a href="#">Save solution</a>		<a href="#">Next page &gt;&gt;</a> << <a href="#">Previous page</a>

Figure 4.7: Ideal solution for fertilizer management task

#### 4.4.6 GENERATION OF SYNTAX AND SEMANTIC CONSTRAINTS

Syntax constraints are generated by ASPIRE based on the domain ontologies created earlier for each task. The syntax constraint generation algorithm extracts all the useful syntactic information from the ontology, such as the restrictions on concepts, properties and relationships with other concepts [Suraweera et al., 2004]. This is then translated into constraints. Each constraint specifies two conditions, a relevance condition and a satisfactory condition. The relevance condition checks to see if the

constraint is relevant to the problem and the satisfactory condition checks to see if student 'solution satisfies the constraint.

The constraint evaluator in ASPIRE checks whether the constraint is satisfied or violated. As DM-Tutor consists of procedural tasks, ASPIRE created an additional set of constraints called path constraints. This was necessary in order to ensure that problem solving occurs in the correct order. Figure 4.8 shows a screen shot of the syntax constraint generator showing constraints that were generated for the Fertilizer Management analysis task.

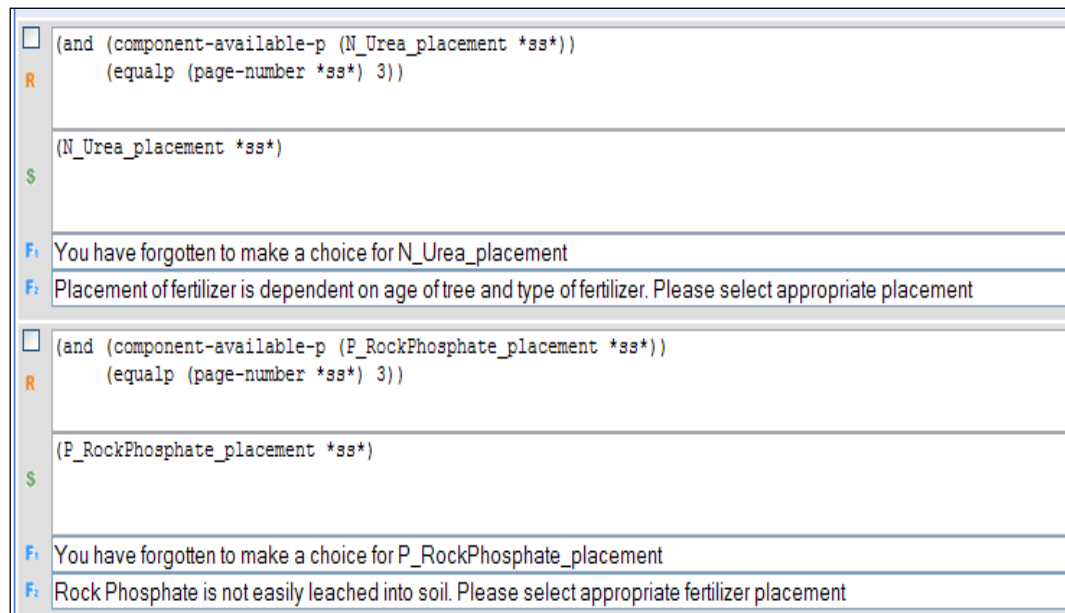


Figure 4.8: Syntax constraints for fertilizer management task

For example, Figure 4.8 shows that in the Fertilizer Management analysis task, one of the constraints generated is:

$$\begin{aligned} &(\text{and} (\text{component-available-p} (\text{N-Urea\_placement } *ss*)) \\ &(\text{equalp} (\text{page-number } *ss*) 3)) \end{aligned}$$

This constraint checks to ensure that the student solution for *fertilizer placement in the weeded circle* is not empty and that the student has made a fertilizer placement

selection before moving on to the next part of the problem. If the student fails to submit an answer, DM-Tutor provides the following feedback messages as shown in Figure 4.8; feedback message 1: *You have forgotten to make a choice for N\_Urea placement.* If the student submits an incorrect solution again, a second feedback message that provides additional help and information is given to the student: *Placement of fertilizer is dependent on age of tree and type of fertilizer. Please select appropriate placement.*

Figure 4.9 shows two semantic constraints generated to check the accuracy of student's solution against the ideal solution for fertilizer placement problem. For example, the first constraint states:

```
(and (equalp (page-number *ss* 4)(component-available-p (Rockphosphate *ss*))(match '(? *d1<i> ?SS-id1 "weeded circle" </i>?*d2) (RockPhosphate *ss*) *bindings*))
```

This constraint checks to see if the student submits a correct solution for the placement of *Rock Phosphate* fertilizer, which is *weeded circle*. If the student submits an incorrect solution, she/he receives the first feedback message that says: *You have made an incorrect choice for Rock Phosphate.* If the student submits an incorrect solution again, she/he is given a more detailed feedback message: *Rock Phosphate is not an easily soluble fertilizer.*

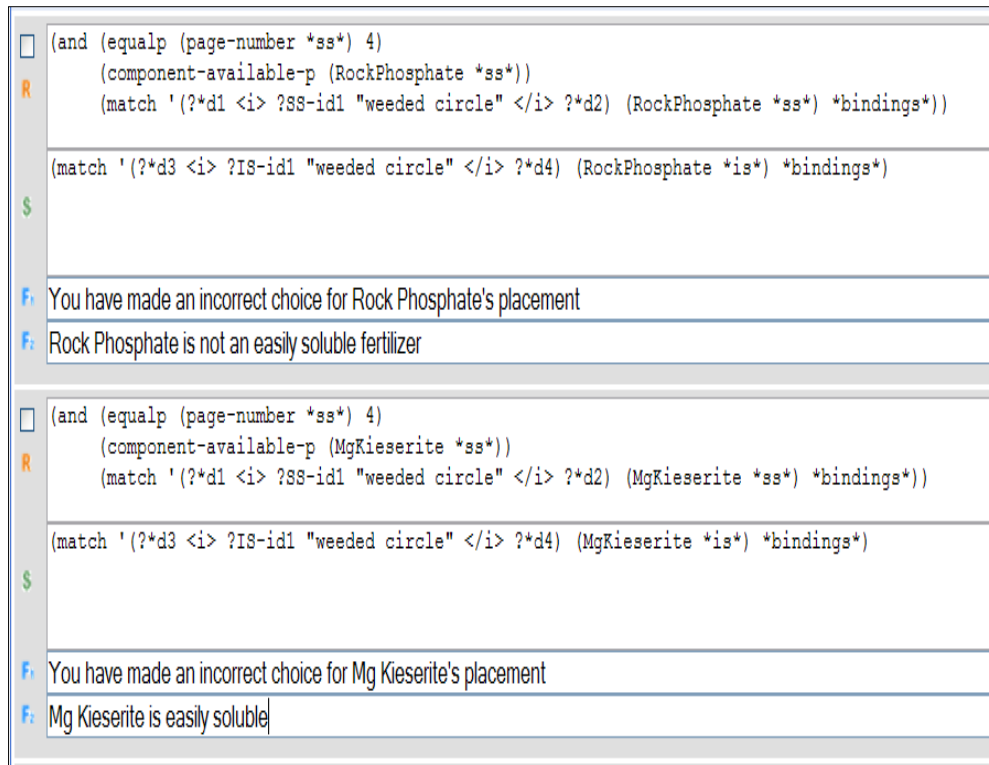


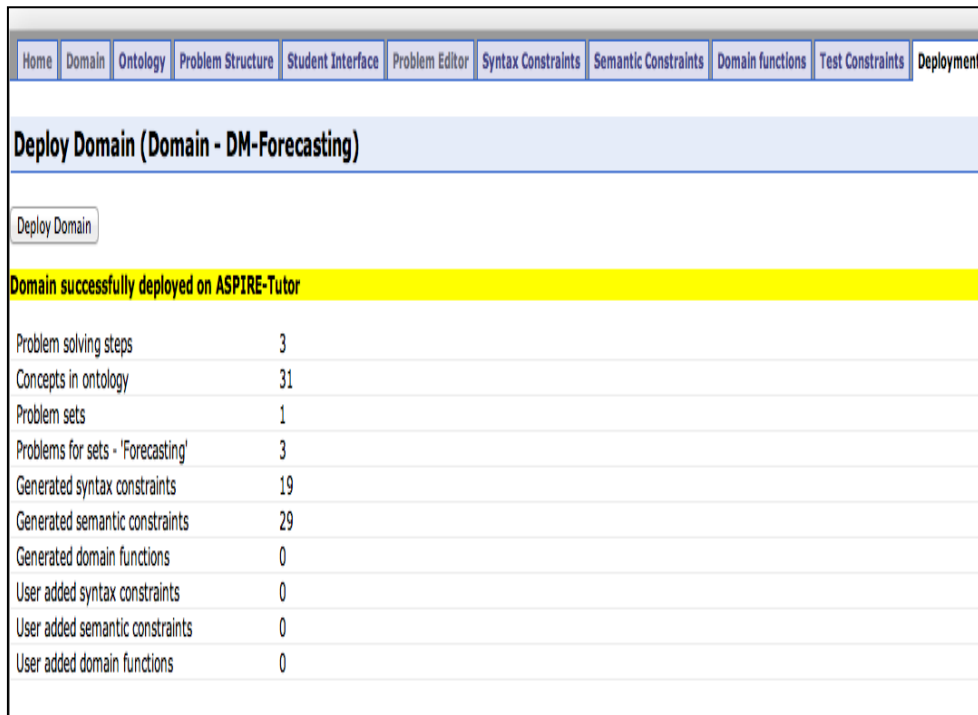
Figure 4.9: Semantic Constraints for Fertilizer Management task

Semantic constraints are generated based on the ideal solution we provided for each problem in DM-Tutor. If there are alternate solutions to problems they would also be compared. Semantic constraints could be specialized to a particular solution or generalized to be consistent across all given solutions. Semantic constraints look for semantics or meaning based on errors in students' solutions enabling DM-Tutor to model alternative correct solution approaches.

#### 4.4.7 DEPLOYMENT OF DM-TUTOR

After ensuring that all the domain information is accurate, DM-Tutor is deployed as an ITS. As we wanted DM-Tutor users to be able to access each of the three tasks (*yield gap analysis, fertilizer management analysis and forecasting analysis*) in DM-

Tutor separately, we deployed each task as a separate domain. Figure 4.10 illustrates the deployment details of forecasting analysis task.



Home	Domain	Ontology	Problem Structure	Student Interface	Problem Editor	Syntax Constraints	Semantic Constraints	Domain functions	Test Constrains	Deployment
<b>Deploy Domain (Domain - DM-Forecasting)</b>										
<input type="button" value="Deploy Domain"/>										
<b>Domain successfully deployed on ASPIRE-Tutor</b>										
Problem solving steps	3									
Concepts in ontology	31									
Problem sets	1									
Problems for sets - 'Forecasting'	3									
Generated syntax constraints	19									
Generated semantic constraints	29									
Generated domain functions	0									
User added syntax constraints	0									
User added semantic constraints	0									
User added domain functions	0									

Figure 4.10: Screen shot showing deployment page for forecasting analysis task

From Figure 4.10 we can observe that there are three problems in the forecasting analysis task, and each problem has three problem solving steps. The deployment page also shows that 19 syntactic and 29 semantic constraints were generated for forecasting analysis task. DM-Tutor has nine problems in total, three for each task. ASPIRE generated 60 syntactic constraints and 139 semantic constraints for DM-Tutor. Once DM-Tutor was deployed, students could access it and carry out problem solving through the student interface similar to Figure 4.5.

## 4.5 YIELD GAP ANALYSIS TASK

When students select to do yield gap analysis in DM-Tutor, they are first presented with a problem that is related to a particular plantation within the MIS as shown by the example below:

*For Estate Ladang Sg Mengah, identify potential yield for August 2008 and calculate the yield gap for each field. Estate Ladang Sg Mengah has been experiencing yield losses for several years due to aging palm trees. Give a suggestion to improve yield for this condition.*

To solve this problem, students will have to first access the accurate report relevant for the yield location and time period. In this case it is Ladang Sg. Mengah estate for the August, 2008. Once they have correctly identified the relevant report, they would need to identify the potential yield for the estate. Secondly, the students will be requested to find the yield for each of the field in Ladang Sg. Mengah estate and to calculate yield gap using an appropriate formula. Next, the students will be asked to identify the field with the highest yield gap. The field with the highest yield gap also has the lowest yield compared to other yield locations within the estate. For the last problem solving step, students would need to provide their suggestion on how to improve yield in the particular location. As mentioned in section 4.4.4 each problem solving step in DM-Tutor appears on a new page.

## 4.6 FERTILIZER MANAGEMENT ANALYSIS TASK

When students wish to learn how to manage fertilizer requirements within a plantation environment, they will be presented with a question similar to the example given below:

*Partial Factor Productivity or PFP is used to identify how much yield is produced for each kg fertilizer used and Agronomic Efficiency or AE is used to identify how much additional yield is produced for each kg of fertilizer nutrient applied for an estate. For the month of August 2008, calculate PFP and AE for Estate Belian, given yield without fertilizer is 0.5 MT/ha. Identify the age of palm trees in the estate and select suitable fertilizer placement. Give reasons to support your selections. Belian Estate has a condition where nutrient applied converted into a non absorbable form due to soil reaction. Give your recommendation to improve this condition.*

The first step to solving this problem is to access the reports relevant to the plantation location and time period from the MIS. Once this step is successfully completed, the students will need to calculate PFP and AE using the appropriate formulas. Next, they will again access the reports to identify the age of the oil palm trees in the estate. After that, they will need to select suitable placement for fertilizers based on the age of the palm tree in the particular plantation location. They will be required to give a reason to support each of their selections. Lastly, they will need to find a solution for a given fertilizer problem faced by the estate.



## 4.7 FORECASTING ANALYSIS TASK

When students select the yield forecasting task, they will be presented with a problem similar to the example given below:

*Ladang Bangka Ulu has 2 estates. For Estate Ladang Bukit Rokan, based on yield for February 2006 forecast yield for February 2007. Estate Ladang Bukit Rokan has been experiencing flooding of 5cm rain water in low lying sections of the field. Select relevant information to identify this problem. Give appropriate recommendation for bigger yield gains.*

For this problem, students need to access and compare two reports from different time periods in the MIS. They then need to select the accurate formula and calculate the forecasted yield for February 2007 based on the yield for February 2006. Next, they need to solve a yield problem by identifying the information they would need to use to solve the problem and also provide a recommendation to increase yield in the given location.

# CHAPTER 5

## AN APPROACH TO INTEGRATING ITSs INTO EXISTING SYSTEMS

### 5.1 REQUIREMENTS

Several factors need to be considered in order to create a framework for embedding ITSs into existing application systems or tools. Our first consideration is about the tool we aim to embed the ITS with. A tool can be defined as any software system that can be used to perform work in a particular domain. One important requirement is that the tool that we are considering allows at least a certain level of modification to be done with it. Another consideration is that the tutoring environment is able to monitor student's learning actions and progressive increase of knowledge. It is also necessary for the tutoring environment to provide appropriate feedback that can help users

achieve their learning goal. We recommend that the learning actions of the student while using the ITS embedded with an application should not just be on how to use the tool but also on applying knowledge gained from using the tool.

It is a challenge to embed an ITS to an existing system, adding to this challenge is our goal of embedding an ITS into an existing workplace system to be used as a training tool in an everyday work place environment. A workplace embedded training system has to be available for the user all the time, as the learner's time is limited, as well as concentration that she/he could allocate for the training at any one time. Therefore, the embedded ITS will be expected to provide just-in-time training for the learner depending on the learner's availability. The embedded ITS must also be able to provide task-oriented training at the learner's required knowledge level because, unlike school or university learning where learners in the same course or grade usually have similar levels of knowledge, employees who work in the same workplace have a variety of knowledge requirements and work experience levels.

Our aim was to embed DM-Tutor into the MIS for palm oil. The MIS contains various plantation analyses and an extensive database of plantation management details. By embedding the ITS into the MIS we hope to leverage on the operational knowledge contained in the MIS to provide on-the-job training to employees through DM-Tutor. We described various embedded systems in Chapter 2. Most of the systems described in Chapter 2 have a deep level of integration with the application systems. There are certain advantages to this type of integration. All user actions at the keystroke and mouse level can be monitored by the tutor. For our research, we wanted to observe only user's problem-solving actions in DM-Tutor and not all of the

user's actions in the MIS. The MIS is a workplace system that is used by employees at all levels, starting from junior executives to senior managers. They use the MIS for many of their business and management functions.

The objective of DM-Tutor is to help novices who are new to MIS and the oil palm plantation domain. We want to observe user's problem-solving actions in the MIS and provide guidance and help them to make effective decisions using the MIS. Our approach to embedding DM-Tutor to the MIS can be described as minimal coupling or minimal integration where the tutor is aware of only certain actions in the embedded application system.

An important factor that contributed to embedding DM-Tutor to the MIS using minimal coupling is because we were only allowed to make minimal changes to the MIS. As the MIS is used extensively by employees every day, it is necessary to ensure that the functionality of the system remained the same after the integration with the ITS. We understood the constraints that employees faced due to their commitments to their daily work load and that they could not afford the extra time required to get used to a new interface, so we made only minimum changes to the interface of the MIS to present the integration of the ITS. As we wanted the ITS to provide learner-paced, task-oriented training to the user, we made the integration of the ITS to the workplace system available to the employees always. The learner or employee could start the ITS, learn any of the analyses they require or want to know, exit the ITS at any point and continue with their daily work using the MIS. The ITS would update the student model with the current knowledge learned and would wait

for the next time the user wishes to practice the same analysis at a different level or practice a different analysis.

## **5.2 COMPARISONS WITH OTHER EMBEDDED SYSTEMS**

When Geometer Sketchpad and Excel Mac 4.0 [Ritter & Koedinger, 1996] were plugged-in with tutoring agents, the systems were modified to allow all the actions of the user while using the systems to be monitored by the tutoring agents so that the system could offer help to students when students did not know what to do next. For ETS [Cheikes et al., 1998] that was embedded into the Complex Information System (CIS), all the actions the students performed in the interface of the CIS were tracked and sent over network sockets to the ETS. The effectiveness of the ETS totally depended on how closely it was able to track the user's actions. In the case of xPST [Gilbert et al., 2009] that was integrated into Paint.NET and CAPE, the actions of the users in the third-party software were also completely monitored and sent to the tutoring system so that the system could intervene when the user is not on the expected problem solving path.

Tracking all the actions of the user is complicated and at times redundant because not all user actions in an external application are related to their learning. This is evident in all the systems we have reviewed above. Students often cancelled work halfway or wander off to explore the system on their own and the tutoring systems were left not able to monitor these actions. To enable the systems to track user's actions effectively, the above tutoring systems had restricted several possible

solution paths for the users to follow and did not allow the users to attempt anything outside the planned solution path. Certain components and functions of the application systems discussed above had to be disallowed from being active as the tutoring systems could not track students' actions when they access these components. As our aim was to create integration with an everyday workplace system with extensive functionalities, we needed to ensure that the integration did not block any component of the application system from working in the same manner even after the integration. Users are free to explore the MIS and access any analyses or reports that they wanted to look at. DM-Tutor did not monitor all the actions of the users in the MIS but only the actions related to learning using DM-Tutor.

PAT [Risco & Reye, 2009] is built inside MS Access. Students access PAT through MS Access and learn how to create forms and reports using MS Access. Students submit their solutions and receive feedback from PAT. In almost the same manner, students access DM-Tutor through the MIS and learn to solve plantation analysis in DM-Tutor. However, to solve the problems presented in DM-Tutor, students need to access relevant information from the MIS and submit their solution in DM-Tutor. The information about the report the student accessed in the MIS is sent directly from the MIS to DM-Tutor as part of the student's solution and is evaluated by ASPIRE. Students receive feedback from ASPIRE through DM-Tutor based on the accuracy of their solution.

### 5.3 ARCHITECTURE FOR EMBEDDED ITSs

Figure 5.1 illustrates our proposed architecture for embedded ITSs taking into account of using DM-Tutor developed through ASPIRE and the MIS. As we have discussed in earlier chapters about the MIS (chapter 3), ASPIRE (chapter 2.2) and DM-Tutor (chapter 4), we now focus our discussion in this chapter on the integration. The MIS and the DM-Tutor are two independent systems connected to two different servers.

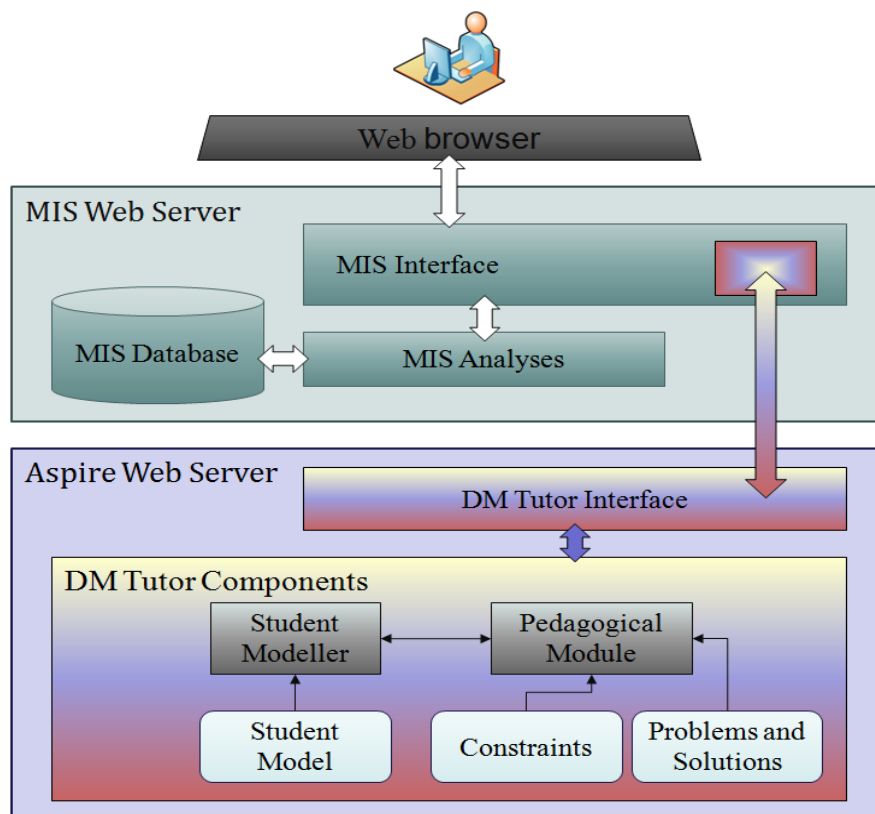


Figure 5.1: Integration architecture of MIS and DM-Tutor

The MIS is a web-based system which is located in a dedicated web server, while the DM-Tutor is located within the ASPIRE server. We designed the integration

to take place in the MIS interface as that is where managers access and view all the various analysis the MIS contains.

From the user's point of view, they access DM-Tutor through the MIS and to them DM-Tutor appears to be a part of the MIS. When the users want to practice or learn an analysis, they select DM-Tutor from the MIS interface and ASPIRE starts to monitor the user's problem solving actions. Normally when employees want to do any operational analysis, they would need to first access relevant reports from the MIS. DM-Tutor trains the user in similar manner. The first problem-solving step in DM-Tutor requires the students to access the relevant reports from the MIS. Through the integration, the information on reports looked at is sent directly to DM-Tutor to be evaluated by ASPIRE.

## **5.4 INTEGRATION OF DM-TUTOR WITH THE MIS**

The integration of DM-Tutor was developed in several stages. In the first stage, DM-Tutor was added as part of the MIS menu as shown in Figure 5.2. When DM-Tutor is selected from the menu, a background process is launched in the MIS to start a session in the ASPIRE system, logging in the user into DM-Tutor.

A frame is made to appear on the MIS interface to present the student interface for DM-Tutor. As shown in Figure 5.2, the frame on the right presents the three tasks available in DM-Tutor. The user is allowed to resize the frame, thus being able to decide the size of the DM-Tutor interface or even completely close it by logging out of DM-Tutor. While the session is active, DM-Tutor monitors user's learning actions.



When students select a task from the three available analyses, the next window opens up in the DM-Tutor interface showing the problem and solution workspace. To solve a problem in DM-Tutor, users need to select and access reports from the MIS that are relevant to the given problem.

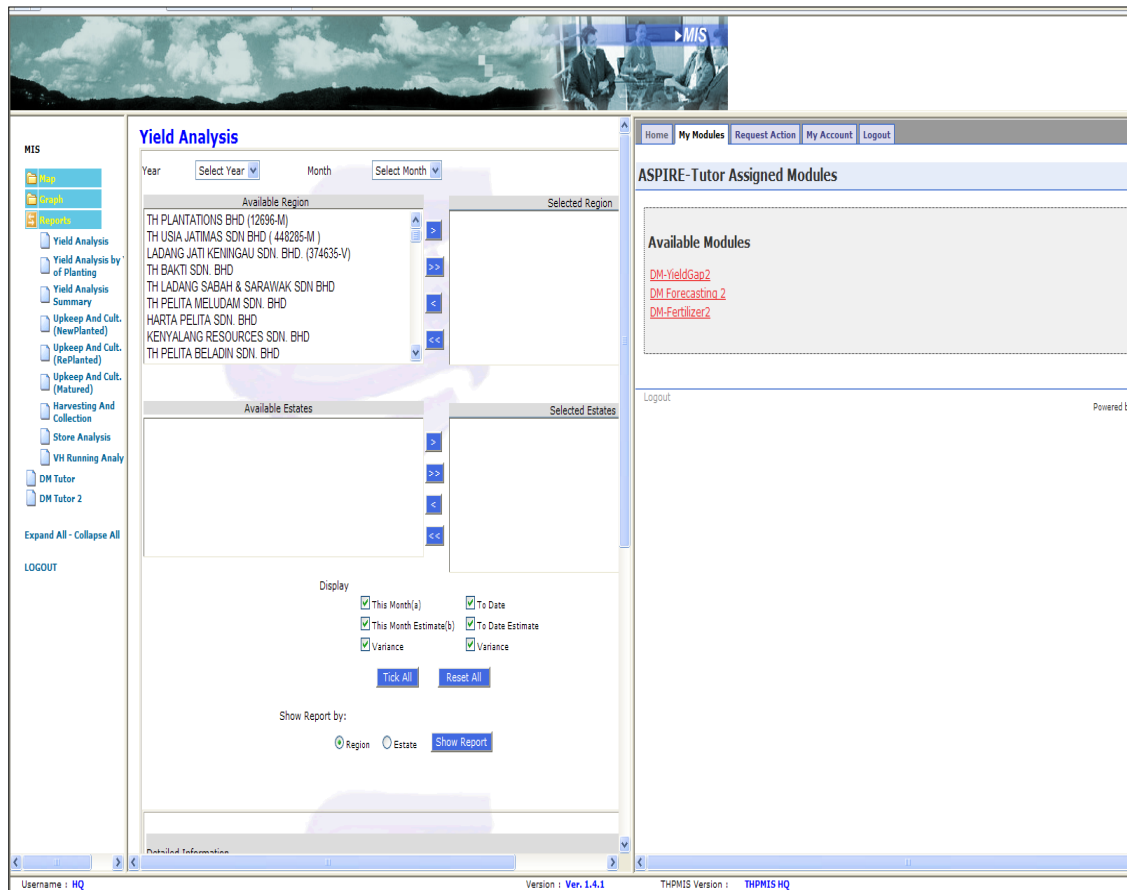


Figure 5.2: DM-Tutor interface presented in the MIS showing tasks available for the users.

When users select a report in the MIS, another separate window opens up in the MIS showing the details of the selected report as shown in Figure 5.3. Figure 5.3 illustrates DM-Tutor's problem-solving page on the extreme right panel, MIS on the left panel and MIS detailed report on the centre window. In the first version of the

integration, DM-Tutor asks the students to type in the information of the reports into DM-Tutor's student interface as part of their answer to the given problem. When students submit their answer, ASPIRE checks their answer against the ideal solution stored in ASPIRE server and gives appropriate feedback to the students. If the answer to the report part of the problem is accurate the student is allowed to move on to the next step. After the first version of the integration was completed, we did a pilot study with a group of employees in a plantation. The results of the evaluation study (discussed in Chapter 6) showed that DM-Tutor embedded into the MIS provided effective training. The participants also provided positive comments on the integrated system interface as well.

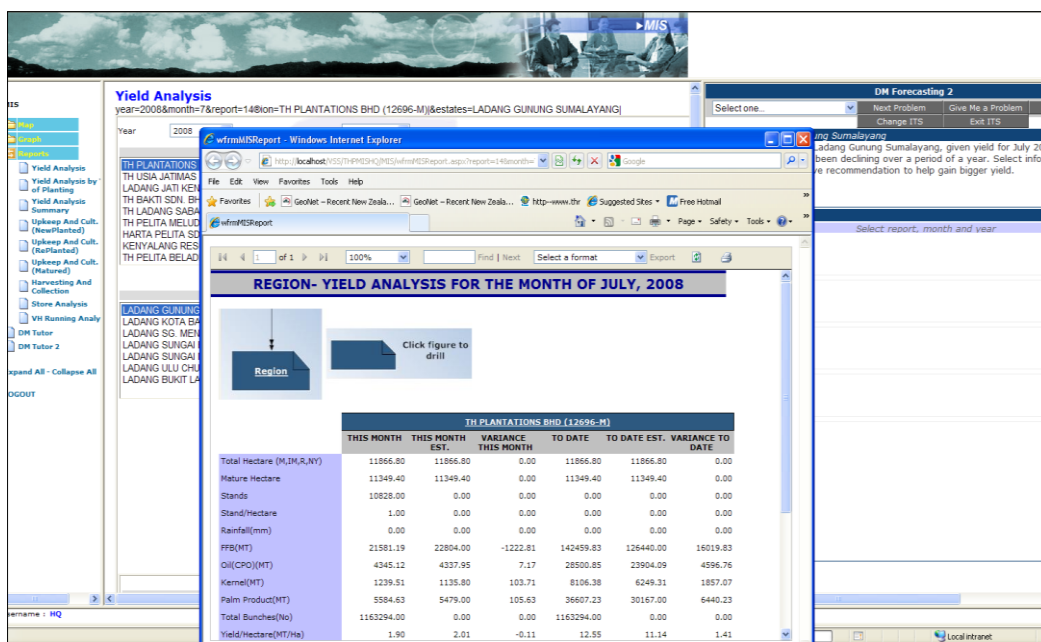


Figure 5.3: Detailed report shown in separate window in the MIS

In the second version of the integration, when users selected a report in the MIS during problem solving in DM-Tutor, several actions take place. As in the first version as shown in Figure 5.3, a window opens up showing the details of the selected

report. Next, the MIS generates a string containing all the information about the report the student looked at and sends this information through the user's browser. Using Java Script that information is composed and sent to DM-Tutor in ASPIRE in form of a GET request, where every piece of information is a parameter of the request. The resulting URL of the request is similar to the example below:

```
http://aspire.cosc.canterbury.ac.nz:8001/problem-  
workspace?populate=true&year=2007&month=March&reportType=Yield+Analys  
is&reportName=LADANG+GUNUNG+SUMALAYANG
```

The problem-workspace of DM-Tutor in ASPIRE checks for the argument "populate" in the URL of the page. If it sees it (and it's set to 'true'), it then grabs the arguments related to the report information including year, month, report type, and the report name from the URL, and then insert their values into the appropriate text fields on the solution structure of DM-Tutor.

The solution is checked by ASPIRE when the student clicks check answer. If the information of the report looked at in the MIS is correct, ASPIRE responds with a feedback message as shown in Figure 5.4 and the user is allowed to move on to the next problem solving step. As mentioned earlier in this section, the user is allowed to resize the frame containing DM-Tutor. Figure 5.4 illustrates this where we have enlarged DM-Tutor's frame to make the problem-solving steps and the feedback panel more visible.

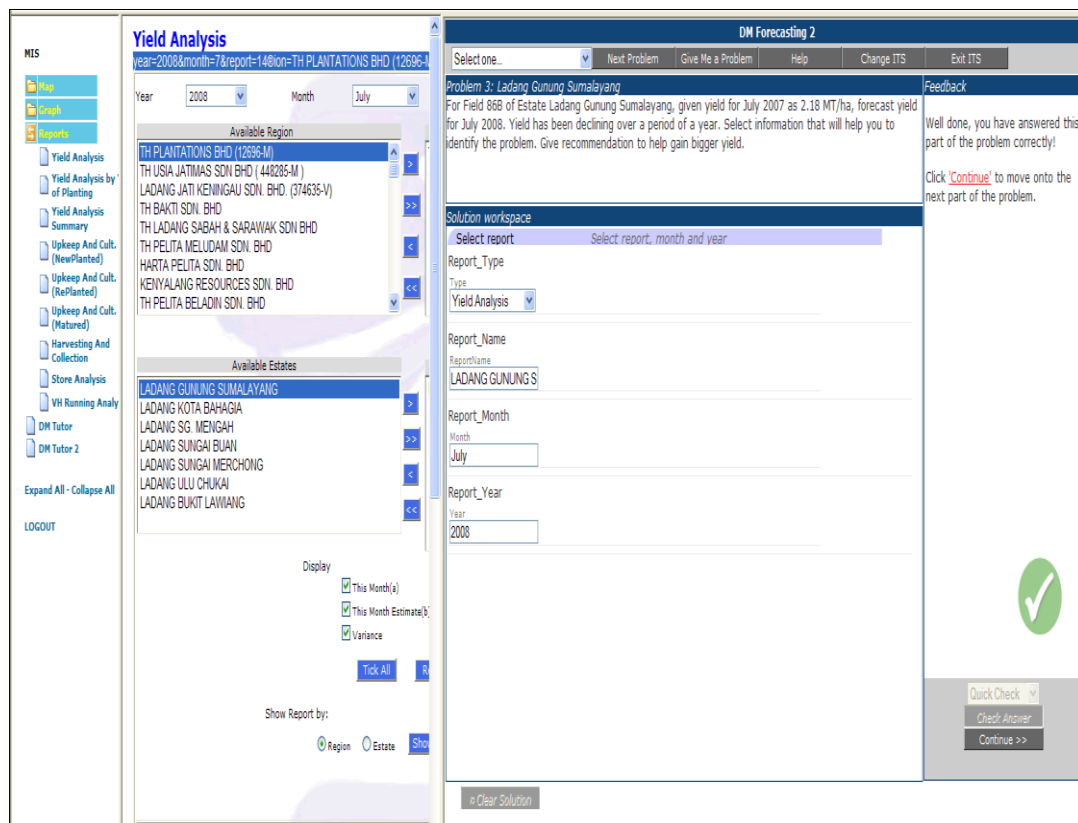


Figure 5.4: Integrated problem solving interface

In our integration approach we have used the MIS for palm oil that contains operational knowledge of palm oil plantation management. Through embedding DM-Tutor with the MIS, we hoped to provide plantation decision making by leveraging on the operational knowledge the MIS contains. DM-Tutor developed with the help ASPIRE provided tutoring, monitoring of user’s learning actions and provided feedback as an embedded ITS.

# CHAPTER 6

## EVALUATION

An evaluation answers the questions for which a system was designed for. The goal of an evaluation is to answer the research question and to test the hypotheses of the research. Evaluation for ITSs should focus on the educational impact of an ITS on students. Since the major goal of an instructional system is to teach, the major test should be whether students learned effectively from it [Littman & Soloway, 1988]. The measurements used to judge the educational effects of ITSs include acquisition of knowledge, understanding, performance, retention, and transfer of a learner's knowledge and skills [Haertel & Calfee, 1983]. In ITSs the observable product is problem solutions. As in any software environment, the efficiency with which the student can solve problems is an important consideration [Helander et al., 1997].

For DM-Tutor, we have followed the same approach by performing two evaluations: a pilot study with employees of a plantation company in Malaysia, and a full evaluation

with students enrolled in the Bachelor of Agriculture Science program at the University Putra Malaysia. This chapter describes the evaluation studies that were performed, and their results. The first section presents the pilot study and the second section details the full evaluation study that was conducted.

### **6.1 PILOT STUDY**

A pilot study is conducted to identify users' problems and concerns of the system being evaluated. The aim is to determine whether systems are used as anticipated [Hofmeister, et al., 1986]. Pilot studies are intended to reveal any serious misjudgements concerning the problem solving environment and to ensure that the tutor supports an acceptable degree of learning. In a typical study, students complete a pre-test, work through problems with the tutor and complete a post-test [Helander et al., 1997].

#### **6.1.1 DESIGN AND IMPLEMENTATION**

We conducted a pilot study of DM-Tutor in February 2011 with a group of employees working for a palm oil plantation company in Kuala Lumpur. The purpose of this study was to gain feedback on DM-Tutor, so that any major technical issues, interface problems and system usability issues could be identified and solved before the full evaluation. As the oil palm plantation employees were familiar with domain and had experience working with the MIS system, we felt that they would be suitable to evaluate DM-Tutor. We informed the employees two weeks before the pilot study was planned to take place. We requested their participation to test the usability of DM-Tutor as a training system for the

oil palm decision-making domain. We informed them that their participation in the study was on a voluntary basis and that they may withdraw from the study anytime they wish to, without any adverse effect on their employment. We wanted them to understand that the purpose of the study was to evaluate the system and not their work performance. As the participants were employees at the executive, management or management trainee level, they were familiar with the MIS for palm oil because they use a similar version of the system for their work.

Each participant was given an instruction sheet explaining the analyses in DM-Tutor. We provided a brief demonstration of DM-Tutor describing the different tasks contained inside DM-Tutor, problem selections and the various levels of feedback messages available. They were informed during this demonstration that their participation in the study was on a voluntary basis.

The participants were given a pre-test (pre-test and post-test are included in Appendix D) before interacting with the system. The pre-test was designed to have one question for each task the users needed to work on with DM-Tutor. Next, the participants were asked to log into the MIS. When they were inside the MIS, they were asked to select DM-Tutor by clicking on the DM-Tutor button. This action automatically logged them into DM-Tutor and provided them with a user ID. The participants were required to interact with the system for one hour. After that, they worked on the post-test and also completed the questionnaire (please refer to Appendix E for questionnaire).

## 6.1.2 RESULTS AND ANALYSIS FOR PILOT STUDY

We did not notice any technical or system operational issues throughout the duration of the pilot study; the participants too did not feel that there were any interaction or interface problems in DM-Tutor. The participants ranged from trainee executives, executives, trainee managers to estate managers. Their work experience in the oil palm plantations ranged from 3 months to 5 years. All of the participants stated that they have used the MIS for their daily work.

Although the study included 19 participants, only four of them interacted with DM-Tutor for one hour or more. On average, the participants interacted with the system for about 27 minutes. The possible reason for this could be that the study was held during their office work hours and they had to complete their pre-assigned daily work load as well. As the participation was on a voluntary basis, the participants also had to make the effort to work on DM-Tutor in their own time.

DM-Tutor recorded the participant's actions within the system in the student logs. From these logs we were able to compute system interaction time, number of problems attempted by participants, number of problems they managed to solve, number of submissions they made into the system and the number of feedback messages they requested and saw in DM-Tutor. The results derived from these records are detailed in Table 6.1.

We found there a significant difference between the participants' pre-test and post-test results ( $t= 1.89$ ,  $p= 0.005$ ,  $df=7$ ). We also found there was a gain between pre-test and post-test results. From those results we could conclude that the participants did improve their knowledge on palm oil decision making after using DM-Tutor. From the



questionnaire we found out that 73.68% of the participants thought that DM-Tutor was easy to use. 52.63% of the participants found the feedback messages in DM-Tutor to be helpful in answering the questions. When they were asked if they liked DM-Tutor's interface, 63.16% of the participants said that they did.

Table 6.1: DM- Tutor interaction results (S.D. given in parentheses)

Interaction Results	Mean (s.d.)
Interaction time (min)	27.20 (24.43)
Number of problems attempted	2.85 (1.23)
Number of problems solved	2.05 (1.05)
Number of submissions made	38.05 (8.96)
Number of hints seen	39.95(19.24)
Pre-test result (%)	50 (16)
Post-test result (%)	78 (21)
Gain (%)	28 (0.15)

One of the participants commented that they liked to see the MIS and this system (DM-Tutor) at the same time. One participant said that he liked this new version of the MIS. When asked if DM-Tutor was able to teach them any new palm oil plantation decision making analyses, all the participants answered 'yes'. One participant said that all the analysis in DM-Tutor helped to teach her how to use the information in the MIS to make better decisions. Another participant said that she knew now what yield gap analysis and fertilizer efficiency were, after using DM-Tutor. When the participants were

asked if they felt that by integrating the Management Information System (MIS) with DM-Tutor they were able to learn the plantation analyses better, all of them answered 'yes'. One participant said that he liked to see the MIS and the teaching system together. Another participant stated that she liked the idea that could stop learning and continue to work with MIS when she needed to and another participant was happy that she could choose the analyses that she needed to learn and that she could check the analysis again if she was not sure.

Overall, the only negative comment they had about DM-Tutor was that some of the feedback messages were not always helpful. Seven participants commented that they could not understand the hints given in DM-Tutor. They also felt that the hints were not descriptive enough. After the pilot study, we modified the feedback messages to make them more explicit and easier for the users to understand. After the pilot study we made further enhancements to the system. We modified the MIS to send the information/reports looked at by the user in the MIS directly to DM-Tutor and we modified DM-Tutor to receive information from the MIS. The enhanced version of DM-Tutor was used for the full evaluation study.

## **6.2 FULL EVALUATION STUDY**

In October 2011 we conducted a full evaluation study with student volunteers from the third and fourth year Bachelor of Agriculture program at the Putra University Malaysia (UPM). The students were enrolled in the Industrial Crops I course, AGR3608. The curriculum of this course included topics such as oil palm plantation industry, plantation

management techniques, commercial field plantings, oil palm harvesting, quality control and processing. As AGR3608 curriculum covered many aspects of oil palm plantation management, we believe that the students have a strong theoretical foundation knowledge related to the oil palm domain.

Even though the objective of DM-Tutor is to provide decision-making training on plantation operations to people working in oil palm plantations, we feel that asking final year students to evaluate DM-Tutor is a reasonable decision. These students are in their final year of their studies and as such are preparing to work in the oil palm domain. Upon completion of their studies many of them would start their employment as management trainees at oil palm plantations. Apart from that the MIS is also currently used by employees who manage the accounting or the human resources area of oil palm operations. These employees have little or no experience in the oil palm domain. Therefore by requesting final year students we would be able to evaluate if DM-Tutor is capable of providing decision-making training to novices in the oil palm domain. This evaluation study was approved by the Human Ethics Committee (HEC) of the University of Canterbury (please refer to Appendix C for HEC Approval).

The students were informed that their participation would be entirely on a voluntary basis and that they could decide to withdraw from the study at any time without any adverse effect on them or their studies. There was a downside to making evaluation studies voluntary, as participants had to make an effort to attend the evaluation in their own time. In addition to this, the students' time was also restricted as a number of them had lectures and exams that they needed to attend on the day of our system evaluation as well. For this study, we requested the students to be available for a total time of two hours

(120 minutes). As we had completed the initial system administration and set up just before the study commenced, students' time could be well utilized for the demo, a pre-test, an estimated 75 minutes interaction time with the system, followed by a post-test and a questionnaire. At the end of the evaluation session each participant was given a compensation of MYR 20 (approximately 7.90 NZD). Data collected during the study was anonymous and each participant was given a random, non-identifying user code to log into the system. The same user code was used to link DM-Tutor or MIS usage, pre-test, post-test and questionnaire data together. We requested the students to retain the user code, so that they could ask us to remove their data at any time from the experiment if they so wished.

### 6.2.1 EVALUATION DESIGN

The objective of this study was to evaluate the effectiveness of DM-Tutor in providing instructions and training in oil palm plantation decision-making to novices. The study was conducted to test the hypothesis that embedded constraint based tutors provide effective training. For this study, we divided participants into two groups: a control group using only the MIS to carry out plantation analyses as managers normally would in oil palm plantations, and an experimental group that attempted to solve the same analyses using the MIS embedded with DM-Tutor. Through the study, we hoped to understand the role of feedback messages in providing effective instructions. Feedback messages are provided to users by DM-Tutor each time they submit an incorrect solution to a problem step. DM-Tutor provides different levels of feedback messages for the user starting from

error flag, hint, detailed hint, all errors and show solution. When needed, users could also request more detailed feedback information from DM-Tutor.

Through the pre-test and post-test (please refer to Appendix D for pre-test and post-test), we would try again to learn user's understanding of the oil palm plantation decision making domain before using DM-Tutor and after using the system. We compare the pre-test and post-test results of the participants to identify if there any learning gains from using DM-Tutor. We record the participants' actions to understand their interaction and learning while using the system. A questionnaire (please refer to Appendix E for questionnaire) is used to gain subjective analysis for the study and users comments on DM-Tutor.

### **6.2.2 PRE AND POST TESTS**

Two tests were created to test the participants' knowledge before and after using the system. The tests were designed to be of similar difficulty, but in order to eliminate any variability between the tests; some participants received test 1 as their pre-test and test 2 as their post-test, whereas others received the reverse. Each test had three questions. Each question covered one task within DM-Tutor and was designed to test student's knowledge in each specific area of plantation decision making.

### **6.2.3 RECRUITMENT OF PARTICIPANTS**

Students from AGR 3608 were informed of the study one week before the evaluation was planned to take place. We informed them that we required their participation to evaluate

software that could help to train oil palm plantation employees on plantation decision making. We explained to the students that their participation would help us evaluate the effectiveness of DM-Tutor as a teaching and training tool. We wanted to ensure that the students clearly understood that we were not trying to test their knowledge in oil palm plantation decision-making domain but rather on how helpful the system was in training them on plantation analyses and decision making.

The students were given a demonstration of the MIS. They were shown how to access various reports from the MIS menu. This was necessary as all the participants will be required to access the MIS reports during the study. Students in the experimental group were given a brief overview of DM-Tutor. They were shown how to access DM-Tutor from the MIS menu, how to select problems, submit solution and how to request feedback from the system. All the students were given information sheets (please refer to Appendix A for information sheet) that explained the study and also explained the student's role during the evaluation study. The information sheet contained a random and unique user id that the students would need to use during the entire evaluation study. Students were also given consent forms (please refer to Appendix B for consent form) that they had to sign and return to us before the end of the study. Both the information sheets and consent forms reminded the students that their participation was on a voluntary basis and that the students could withdraw from the study anytime they wish and that all data related to them will be completely removed if they wished to withdraw from the study.

#### **6.2.4 EVALUATION PROCEDURE**

Participants were evaluated in one session. They were divided into two groups, the experimental and control group. Participants in the control group logged in into the MIS and were assigned a user ID and password. They did the pre-test and then proceeded to do the analyses on paper while accessing the information from the MIS (please refer to Appendix F for problems the control group attempted). Participants were requested to attempt three problems each for Yield gap analysis, Fertilizer analysis and Forecasting analysis, a total of nine questions. The questions attempted by the control group were identical to the questions attempted by the experimental group using the MIS embedded with DM-Tutor. After attempting the analyses, they proceeded to do the post-test and the questionnaire. The participants in the experimental group did the pre-test, logged into the MIS, and then selected DM-Tutor from the MIS menu. Participants selected from Yield gap analysis, Fertilizer analysis and Forecasting analysis tasks in DM-Tutor and were presented with the problems within DM-Tutor. After attempting the analyses, they proceeded to do the post-test and then the questionnaire.

#### **6.2.5 RESULTS AND ANALYSIS FOR FULL EVALUATION STUDY**

68 out of a total number of 74 students from AGR3608 participated in the evaluation study. We had hoped for an even distribution of the student numbers between the experimental and control group, but because the rooms allocated by UPM for the evaluation were in different locations and also of unequal room sizes, we ended up having 28 students in the experimental group and 40 students in the control group.

The students were requested to be available for 120 minutes for the evaluation study including an estimated 75 minutes interaction with the systems. However we encouraged them to continue working with the systems if their schedules permitted it. On average the students in the experimental group interacted with DM-Tutor embedded into the MIS for about 96 minutes and students in the control group did problem solving using MIS alone for about 76 minutes. For both the control and experimental group, we were able to compute the students' interaction time with the system, number of problems attempted by participants and the number of problems they managed to solve. For the experimental group using DM-Tutor we were able to also compute number of submissions students made into the system and the number of feedback messages they requested and saw in DM-Tutor. The statistical results obtained from the student interaction are presented in Table 6.2.

Table 6.2: System interaction statistics (S.D. given in parentheses)

	Experimental	Control	Significant
Number of students	28	40	
Time spent on problem solving (min)	96.10 (24.22)	75.55 (5.20)	p<0.01
Number of problems attempted	7.43 (1.60)	7.22 (1.40)	p=0.26
Number of problems solved	6.04 (2.10)	4.05 (1.06)	p<0.01
Number of submissions made	89.38 (33.36)	N/A	
Number of hints seen	91.64 (42.87)	N/A	



There is a significant difference ( $t=4.4$ ,  $p<0.01$ ,  $df= 29$ ) in system interaction time between the experimental and control groups. The control group results show that only 40% of the students completed all parts of each given problem during the study. It could be that when students in the control group did not know how to answer a question, they stopped attempting that question and moved on to the next. Students in the experimental group received feedback from the system each time they submitted a wrong solution and moved on to the next step. Students in the experimental group were required by the system to complete each problem step before moving on to the next. This behaviour increased their interaction time with the system. We would also observe through the questionnaire analysis that the students were interested to interact with DM-Tutor. This could be another reason why they interacted longer with the system.

Both groups attempted almost the same number of problems as there is no significant difference between them, but there is a significant difference ( $t=4.6$ ,  $p<0.01$ ,  $df=37$ ) in the number of problems they solved. As plantation decision making is a complex domain for novices it could be that students in the control group found a number of the problems or problem steps to be too difficult and could not solve them. For example, students in the control group answered report information inaccurately 35% of the time and also repeatedly made the same mistakes. 39% of students in the experimental group submitted wrong answers related to reports looked at in the MIS the first time they encountered the questions, but in subsequent attempts they were able to identify and use correct reports. Experimental group students were able to request increasing levels of feedback to guide them until they managed to solve a problem step and each problem.

We could observe this behaviour by seeing the number of submissions (please note that submissions are partial answers, covering only the current problem step) the students in the experimental group made into the system and the number of feedback messages they requested and saw during the entire study. Students in both the control and experimental groups were administered the pre-test and post-test and the results are presented in Table 6.3.

We found no significant difference between the pre-test results of the two groups, indicating that the two groups are comparable. The control group improved significantly between pre-test and post-test ( $t= 14.3$ ,  $p<0.01$ ,  $df=39$ ). This shows that the students in the control group did improve on their knowledge of palm oil plantation decision making after interacting with the MIS.

Table 6 3: Pre-test and post-test scores (S.D given in parentheses)

	Experimental Group	Control group	Significant
Pre-test (%)	47.50(10.38)	50.40(9.71)	n.s
Post-test (%)	85.07(10.54)	63.70(9.93)	$p<0.01$
Gain	37.57(14.38)	13.37(5.84)	$p<0.01$
Normalised Gain	70.61(20.96)	27.20(10.53)	$p<0.01$

The experimental group had significant improvement from the pre-test to the post-test results ( $t= 13.43$ ,  $p<0.01$ ,  $df= 54$ ). There is a significant difference ( $t=8.4$ ,  $p<0.01$ ,  $df= 56$ ) between the post-test results of both groups. This shows that the experimental group learned significantly more than the control group, proving that the treatment was

effective for the experimental group. This behaviour is further emphasised when we analysed gain ( $t=8.45$ ,  $p<0.01$ ,  $df=33$ ) and normalised gain ( $t=8.46$ ,  $p<0.01$ ,  $df=33$ ) scores, they show significant differences between experimental and control group.

We calculated the effect size to assess learning effectiveness of using our system. The effect size enables us to compare results of one pedagogical experiment with another. For ITSs, the common way to calculate effect size is to subtract the control group's mean gains score from the experimental group's mean gain score and divide by the standard deviation of the gain scores of both the groups. We calculated the effect size of this experiment to be 1.62, as the effect size is very large; it indicates that the experiment would produce results of this level of statistical significance for the same design and similar population of participants.

The domain knowledge of DM-Tutor is represented as constraints. If the constraints represent an appropriate unit of knowledge of the domain, then learning should follow a smooth curve with a decreasing trend in terms of constraint violations [Mitrovic et al., 2002]. We evaluated this possibility by analysing the student logs and identifying each problem state when the constraint was relevant. From the analysis, we calculated, for each participant in the experimental group, the probability of violating an individual constraint on the first occasion of application, the second occasion and so on. The individual probabilities were averaged across all the constraints in order to obtain an estimation of the probability of violating a given constraint  $C$  on a given occasion. The probabilities were then averaged across all the participants in the experimental group and plotted as a function of the number of occasions when  $C$  was relevant as shown in Figure 6.1.

The graph is for all constraints seen by the students. The curve shows a regular decrease in the probability that can be approximated by a power curve overlaid on the figure and has a very close fit to the data set with an  $R^2$  power-law fit of 0.97. The initial probability of violating a constraint is approximately 19%. This probability can be considered high and due to the fact that even though students did have theoretical knowledge of plantation management aspects, they did not know initially how to apply that knowledge to the actual or real- life plantation management analyses and problems.

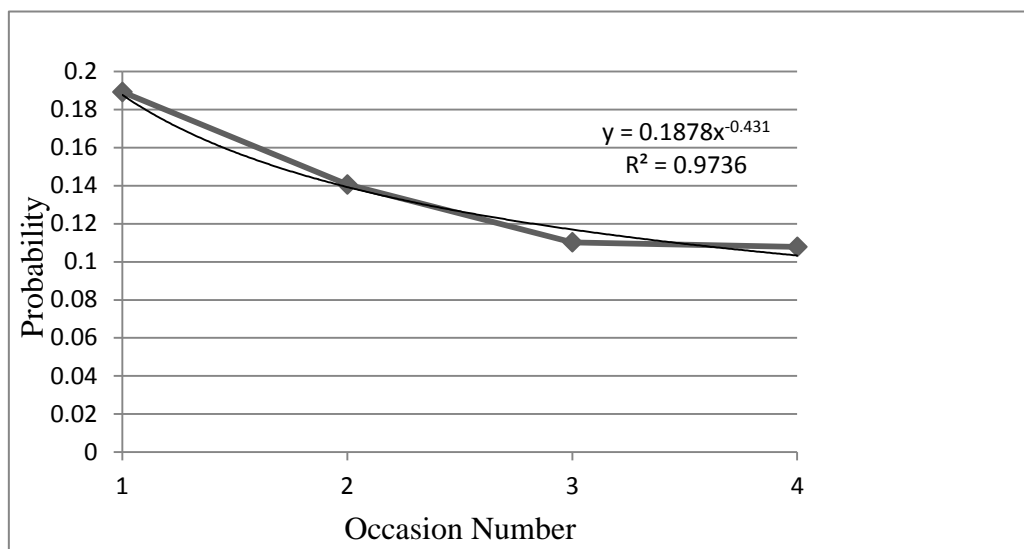


Figure 6.1: Probability of violating a constraint as a function of the occasion when that constraint was relevant

The total number of constraints violated on the first attempt was 3810. The probability for the students to violate constraints on the first occasion was approximately 19%. After four occasions, the probability of violating a constraint dropped down to 11% at 2040 constraints, close to 50% from the first attempt. The steepness of this curve is indication of the speed with which the student, on average, learned new constraints. Since

each constraint represented a specific concept in the domain, we can conclude that this is an indication of how quickly the student was learning the domain.

This graph supports our earlier observation that students have improved their knowledge in oil palm plantation management analyses significantly and effectively through interacting with DM-Tutor embedded into the MIS. The graph also confirms our earlier understanding that students in the experimental group did make mistakes when submitting solutions for report information initially but on subsequent attempts managed to submit correct solutions for that part of the problem and moved on to the next. We do need to accept that as the curve progresses, learning behaviour could contain erroneous behaviour such as slips or random mistakes, so the graph stops trending along the power curve and levels out at the level of random mistakes.

Students from both groups completed a questionnaire at the end of the study. A questionnaire is an effective way to obtain subjective analysis from the participants. It gives the participants an opportunity to provide their opinions and comments of the system, the things they liked and also the things they did not like. The questionnaire also gives us an opportunity to obtain additional information about the participants and their thoughts. From the questionnaires, we found out that only 5 students from both groups have had some experience using plantation management software during one of their practical training experience. Table 6.4 presents the participants' responses in the questionnaires. The answers were on a five point scale, ranging from 1 (*Not at all*) to 5 (*Very much*). We used the Mann-Whitney U test to analyse responses. There is a significant difference ( $U = 319.5$ ,  $N_C=40$ ,  $N_E=28$ ,  $p<0.01$ ) between the groups when we asked if the problem solving questions were easy to attempt. The students in the

experimental group found the questions easier to attempt possibly due to the help they received from the feedback messages. There was also a significant difference ( $U=133.5$ ,  $N_C=40$ ,  $N_E=28$   $p=0$ ) between the two groups when we asked if they found the system easy to use. As expected, the students in the experimental group found DM-Tutor easier to use compared to students who did problem solving using MIS alone. As we can observe from table 6.4, students in the experimental group said that they found the feedback messages to be helpful to them.

Table 6.4: Scores from the questionnaires (Likert scale from 1-5) S.D. given in parentheses.

Survey Questions	Experimental	Control	Significant
Questions easy to attempt?	3.6 (0.87)	2.98 (0.6)	$U=319.5$ , $p<0.01$
Ease of use	3.8 (0.65)	2.70 (0.70)	$U=133.5$ , $p=0$
Helpfulness of feedback messages	4.0 (0.60)	N/A	N/A

As we wanted to know what students thought about the systems we designed the following questions to be open-ended. When we asked them if they liked the interface, interestingly, 85% students from both groups said they liked the interface but there were additional comments from the students. A number of students from the control group said they found the MIS to be difficult to use and that it took them a long time to understand what to do, some students commented that they felt like they needed more time to explore the MIS's interface. This is understandable as the MIS and the oil palm plantation management domain is a complex environment. 93% of the students in the experimental

group said that they found the system interesting to use and helpful in teaching them plantation analyses.

All the students in the experimental group and 62% of the students in the control group answered 'yes' when asked if they felt that they were able to learn any new analyses in oil palm plantation decision making from the systems. For most of the students in both groups this is the first time they have interacted with a plantation management software like the MIS or DM-Tutor (embedded with the MIS) and when we compared the pre-test and post-test results there has been significant improvement within the groups in both the experimental and control group students. The students in the experimental group gave many positive comments about DM-Tutor. Many students felt that they could learn plantation analyses better by using DM-Tutor. One student said that he felt like a plantation manager solving plantation problems such as fertilizer consumption and potential yield through using DM-Tutor. Another student said that he felt that DM-Tutor was interesting and easy to learn even though he did not have any plantation management experience. Yet another student said that she felt that she learned how to increase yield and identify suitable fertilizer recommendation according to crop age from using DM-Tutor. These positive comments were very encouraging and support our belief in DM-Tutor's potential as a teaching and training tool for the oil palm plantation decision making domain.

When we asked the students if they thought that by using the MIS embedded with DM-Tutor they were able to learn the plantation analyses better, all the students in the experimental group said 'yes'. We received many positive comments from the students. One student said that MIS integrated with DM-Tutor is very good idea as he could see

how to extract data from MIS and use it for plantation decision making. Another student said that she has learned from DM-Tutor in a very short time how to understand oil palm plantation's fertilizer requirements and that the feedback in the system has been very helpful. Yet another student said that integration of MIS and DM-Tutor could teach him understand effective plantation decision making.

When we asked if the students thought that DM-Tutor was interesting to use, all but two students said that it was interesting. The two students thought DM-Tutor was not interesting to use and one of them felt that the Fertilizer analysis task was too difficult to him. Another student said that he did find DM-Tutor interesting to use but did not like the idea that he still had to do the calculations using the computer's calculator and hoped that the system could be improved that it could do the calculations automatically. At present DM-Tutor does not provide automatic calculations. Apart from those three comments, all the rest of the students had positive comments about DM-Tutor. Many students said they thought DM-Tutor was interesting and that it gave good guidance and help for novice users. A number of students said that they felt that it taught them how to understand complex analyses in an easy way with the help of feedback messages and that the problem solving actions were effective in helping them understand plantation decision making better.



# CHAPTER 7

## CONCLUSIONS AND FUTURE WORK

This research has presented an approach to creating a framework for embedded ITSs. Embedding an ITS into an existing application allows us to leverage on the knowledge contained within an application system for the development of the ITSs domain model. We demonstrated an architecture for integrating ITSs into existing systems using DM-Tutor developed through ASPIRE and the MIS for palm oil. For the development and implementation of our framework we needed to take into consideration the ITS to be developed, the domain and the application system where the ITS was to be embedded. We needed to consider the potential users and the environment that the embedded ITS had to be made available.

DM-Tutor is the first ITS developed for the oil palm domain. It is the first CBT to be embedded within an existing system. DM-Tutor was developed in the ASPIRE authoring system. We studied the characteristics of the oil palm domain to create the domain model for DM-Tutor. All the problems for DM-Tutor were designed based on the current requirements of the oil palm plantation domain. The ontologies created were crucial for constraints generation. We also identified a suitable problem and solution structure for the tasks in DM-Tutor. ASPIRE generated constraints and developed an accurate constraint base. Finally, DM-Tutor was deployed as an ITS. Each task created in DM-Tutor is designed to teach students how to solve a specific analysis in oil palm plantation management. DM-Tutor monitors students' learning actions and provides feedback. A student model contains the record of student's learning progress and their correct and incorrect knowledge of the domain.

The MIS for palm oil is a complex and comprehensive system that contains vital plantation information for fertilizer, yield and cost management. A manager who is managing the oil palm estate or plantation needs to access a number of relevant reports and carry out a number of analyses before being able to make an effective decision. A manager needs good knowledge of the MIS and palm oil domain to make an accurate analysis or decision. This puts new or inexperienced managers at a severe disadvantage and affects their decision making capability. The MIS is a workplace system used by oil palm plantation employees on a daily basis. For our research we have integrated DM-Tutor with the MIS for oil palm to provide training on plantation decision making using

real life operational data from the MIS. MIS was modified to allow the integration with DM-Tutor.

We conducted two evaluations to assess DM-Tutor's effectiveness in providing training. The first was a pilot study and the second was a full evaluation. Both evaluation studies included pre-tests and post-tests, system interaction and questionnaire analysis. We conducted the pilot study with a volunteer group of employees from an oil palm plantation company. The pilot study was meant to gauge users' opinions of DM-Tutor's usefulness as a training system for the oil palm domain. From the results of the study we could conclude that the participants significantly improved their knowledge in oil palm plantation management after using DM-Tutor. The participants also said that they found DM-Tutor easy to use and that they were interested to use it. Most of the comments given by the participants about DM-Tutor were positive. The only negative comment the participants gave was that the feedback messages given by DM-Tutor was not always helpful and that they could not always understand them. After the pilot study, we enhanced DM-Tutor and improved the feedback messages. We also made DM-Tutor more interactive so that the report information seen in the MIS was sent directly from the MIS to DM-Tutor in ASPIRE.

We did a full evaluation with student volunteers from the Bachelor of Agriculture Science program. Even though the objective of DM-Tutor was to provide just-in-time workplace training, we felt that this particular group of students were a reasonable choice of participants to evaluate the effectiveness of DM-Tutor as training tool for oil palm

plantation domain. The students were in their third and final year of their agriculture studies degree program and have strong background knowledge on oil palm plantation management concepts. They are in preparation to work as management trainees in the oil palm industry upon completion of their studies. By requesting this group of students to evaluate DM-Tutor, we were able to learn the effectiveness of DM-Tutor in providing decision-making training to novices.

The results of the evaluation study prove that DM-Tutor is an effective teaching and training tool for the oil palm plantation decision making domain. Students have shown significant improvement in their knowledge of oil palm plantation management after interacting with the system. This has been proven through the pre-test and post-test results. We have further emphasised this deduction through the plotted graph that displayed a power curve that has a close fit to the data set. From the questionnaire analysis we observed that students found DM-Tutor to be easy to understand and interesting to use. The students also stated that they were happy to learn various plantation analyses from using DM-Tutor and were confident that from using DM-Tutor they will be able to understand real life plantation decision making.

We can conclude that a general framework for embedded ITSs should include certain criteria. Firstly, it should include an application system containing knowledge of a particular domain. The application system should be modifiable to some extent to allow the integration to take place. If the tutor is being integrated with an application system used heavily for functions other than tutoring, the integration with the tutoring

environment should not restrict the functionality of the application by disallowing or blocking any components within the external application. The tutoring environment embedded into the system should monitor students learning actions and provide appropriate feedback. It should retain student's learning progress in a student model that is updated with every progress the student makes in the tutoring environment.

We have created the architecture for embedded ITSs and proven the effectiveness of our approach to framework for embedded ITSs by using DM-Tutor developed through ASPIRE and the MIS for palm oil. The results of the evaluation study showed high learning gains and a big effect size.

Through this research we have made several significant contributions. DM-Tutor is the first ITS for the oil palm domain. It is the first CBT to be embedded within an existing application system. Finally and very importantly, through this research we have created a framework for embedded ITSs and proven its effectiveness in providing tutoring and training for a workplace environment.

### **7.1 RESEARCH LIMITATIONS**

We believe there are avenues for improving DM-Tutor and our research. In the evaluation study, the control group used the standard MIS that is normally used by oil palm plantation employees. We plan to carry out another study where we incorporate worked examples into the MIS as tuition for the control group. The experimental group would use DM-Tutor. We would like to research how this change would affect the results

of the study. For this research we only conducted a pilot study and one full evaluation study. In our full evaluation study we could not obtain similar participant sizes for our control and experimental group. Our control group was larger than our experimental group. We plan to conduct more evaluation studies to analyse further the effectiveness of DM-Tutor as a training system.

We plan to evaluate DM-Tutor with oil palm plantation employees at different levels of working experience, we would like to study the learning differences and analyse their experiences of using DM-Tutor. In the pilot study conducted with oil palm plantation employees, the results showed that some participants felt that the feedback messages were not very helpful because the hints were not descriptive enough. We improved the feedback messages and conducted a full evaluation study. The results showed that students participating in the study found the feedback messages helpful for their problem solving activities in DM-Tutor. We would like to research the effects of the improved feedback messages in a study conducted with oil palm plantation employees.

One method of measuring students learning in a CBT is by identifying the number of constraints students violate or satisfy while using the CBT. When constraints are satisfied, they represent concepts learned and when constraints are violated, they represent concepts students failed to learn. In our research we identified the number of constraints that were relevant for the students. We analysed the number of constraints the students violated on their first attempt of solving a problem step in DM-Tutor and on subsequent attempts. From the results of our evaluation studies we observed that the

number of constraints violated by the participants reduced the longer they interacted with DM-Tutor. However, in this research we did not analyse the number of constraints students violated on all uses, or fifty percent of the time they interacted with DM-Tutor. We also did not analyse individual constraints students violated or concepts they failed to learn. In future evaluation studies we plan to include all of these analyses as well as they would help us to identify concepts students do not understand or failed to learn.

DM-Tutor was developed using ASPIRE authoring system. The approach we have used to integrate DM-Tutor into the MIS for palm oil is suitable and proven to be effective for a CBT developed in ASPIRE and external applications that work similar to the MIS for palm oil. This approach may not be suitable for an ITS or learning system that would require to observe every action the user makes in an external application and track those actions to provide guidance to the learners.

## 7.2 FUTURE DIRECTIONS

Currently DM-Tutor trains users on making effective decisions for yield gap analysis, fertilizer analysis and yield forecasting analysis. We plan to extend DM-Tutor in the future so that DM-Tutor could also teach users how to make effective decisions related to plantation sustainability and environmental concerns.

We would like to research the possibilities of including tutoring dialogues as additional feedback where DM-Tutor could communicate with the students on the

decisions they make during the analyses. By discussing their decisions with DM-Tutor the student could evaluate the decision they made and could make a better decision. From the evaluation studies we have observed that feedback was very important in helping the participants understand how to solve the given problems. Therefore, we feel that having tutorial dialogues could further increase the effectiveness of the training and enable students to make better decisions and judgements.

At present the MIS for palm oil is an application that is more of an interface for an oil palm database. Users access reports and carry out plantation analyses on their own based on their skills and abilities. It is possible that improvements made to the MIS could enable these analyses to be automated, thus converting the MIS into a Decision Support System (DSS). We believe that even when future improvements are done to the MIS, DM-Tutor would still be beneficial as a training tool. As stated in the earlier part of this thesis decision-making is a process that requires the combination of multiple abilities and novices to the oil palm plantation domain would not be able to easily acquire them by just looking at reports or automated analyses.

In this research DM-Tutor was embedded into the MIS for palm oil. The MIS for palm oil was the domain chosen to prove the effectiveness of the integration approach used. The approach used for the development of DM-tutor and the creation of a framework for embedded tutors can be extended for other CBTs developed in ASPIRE integrated with external applications in other domains similar to the MIS for palm oil. We would like to develop more embedded ITSs and evaluate their usefulness in various



## Chapter 7: Conclusions and Future Work

---

domains so that we can further prove the effectiveness of our minimal integration approach and the generality of our framework.

This research shows that DM-Tutor integrated with the MIS for palm oil has the potential of providing life-long workplace learning, anywhere and anytime for the domain of oil palm plantation decision making. We believe that our research has many potential benefits of providing workplace learning and domains that are similar to the MIS for palm oil. Apart from all the benefits discussed in this thesis, we also believe that our minimal coupling approach in embedding ITSs to existing systems could be applied to other ITSs, expert systems, decision support systems, online help systems and other learning systems.

# REFERENCES

- Anderson, J. and Reiser, B. (1985). The LISP Tutor. *Byte*, 10(4), pp.159-175.
- Anderson, J.R., Corbett, A.T., Koedinger, K & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, 4(2), pp. 167-207.
- Anderson, J. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51(4), pp. 355-365.
- Angros, R., Johnson, W. L., Rickel, J., & Scholer, A. (2002). Learning Domain Knowledge for Teaching Procedural Skills. In M. Gini, T. Ishida, C. Castelfranchi, W.L. Johnson (Eds.) *Autonomous Agents and Multiagent Systems AAMAS 2002*, pp. 1372-1378.
- Aleven, V., McLaren, B., Sewall, J. & Koedinger, K. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. In M. Ikeda, K. Ashley, & T-W Chan (Eds.). *In Proceedings of Intelligent Tutoring Systems ITS 2006*, pp. 61-70.
- Baghaei, N., Mitrovic, A., and Irwin, W. (2007). Supporting Collaborative Learning and Problem-Solving in a Constraint-Based CSCL Environment for UML Class Diagrams. *International Journal of Computer-Supported Collaborative Learning Computer-Supported Collaborative Learning* 2, pp. 159–190.
- Beck, J., Stern, M., and Haugsjaa, E. (1996). Applications of AI in education. *Crossroads*, 3(1), pp. 11-15.
- Blessing, S. B. (1997). A Programming by Demonstration Authoring Tool for Model-Tracing Tutors. *International Journal of Artificial Intelligence in Education*, 8, pp. 233-261.

## References

---

- Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), pp. 4-16.
- Carroll, J.M. and Carrithers, C. "Training Wheels in a User Interface," (1984). *Communications of the ACM*, 27(8), pp. 800-806.
- Cheikes, B.A., Geier, M., Hyland, R., Linton, F., Rodi, L. & Schaefer, H.P. (1998). Embedded training for complex information systems. *Artificial Intelligence in Education*, 10, pp. 314-334.
- Dede, C. (1986). A review and synthesis of recent research in intelligent computer-assisted instruction. *International Man-Machine studies*, 24, pp. 329-353.
- Fairhurst, Thomas (2003). Efficient management of natural resources in oil palm plantations. *Proceedings of the International Planters Conference*, June 2003, pp 135-174.
- Freedman, R. (2000). What is an intelligent tutoring system, *Intelligence*, 11(3), pp. 15-16.
- Genesereth, M.R. (1979). The role of plans in automated consultation. *In Proceedings of the 6<sup>th</sup> Int. Joint Conference on Artificial Intelligence*, pp. 311-319.
- Gilbert, B. Stephen, Blessing, B. Stephen, Blankenship, Liz (2009). The accidental tutor: overlaying an intelligent tutor on an existing user interface. *Spotlights on WIP, In Proceedings of the 27th International Conference extended abstracts on Human factors in Computing systems. (CHI 2009)*, pp. 4603-4608.
- Gonzales, C. Burguillo, C., Juan, L and Martin (2007). Intelligent Tutoring Systems and Health Information Systems. *In Proceedings of 18<sup>th</sup> International Workshop on Databases and Expert System Applications*, pp. 633-637.
- Haertel, E. & Calfee, R. (1983) School achievement: Thinking about what to test. *Journal of Educational Achievement*, 20(2), pp.119-132.
- Helander, M., Landauer, T., Prabhu, P. (1997) *Handbook of Human Computer Interaction*. Amsterdam: Elsevier Science.

## References

---

- Hofmeister, A.M. (1986). Formative evaluation in the development and validation of expert systems in education. *Computational Intelligence*, 2, pp. 65-67.
- Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K.: 1998, The Lumiere Project: Bayesian user modeling for inferring the goals and needs of software users. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (Madison, WI), Morgan Kaufmann, San Francisco, pp. 256-265.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- Littman, D. & Soloway, E. (1988) Evaluating ITSs: The cognitive science perspective. In M.C. Polson & J.J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. Lawrence Erlbaum Associates.
- Lesgold, A., Lajoie, S., Bunzo, M. & Eggan, G. (1988). SHERLOCK: A coached practice environment for an electronics troubleshooting Job. Pittsburgh Univ PA Learning Research and Development Center.
- Mathan, S.A. & Koedinger, K.R. (2002). An empirical assessment of comprehension fostering features in an intelligent tutoring system. *In Proceedings of the 6th International Conference: ITS 2002.LNCS 2363*, pp. 330-343.
- Matthews, P. (1999). Workplace learning: developing a holistic model. *The Learning Organization*, vol. 6 Iss: 1, pp. 18-29.
- Mayo, M., Mitrovic, A., McKenzie, J. (2000) CAPIT: an Intelligent Tutoring System for Capitalization and Punctuation. *International Workshop for Advanced Learning Technologies IWALT2000*, pp. 151-154.
- Mitrovic, A. (1998). Experiences in Implementing Constraint-Based Modeling in SQL-Tutor. *In Proceedings of ITS'98*, San Antonio, 17-19 August 1998, pp. 414-423.
- Mitrovic, A. & Ohlsson, S (1999). Evaluation of a Constraint-Based Tutor for a Database Language Mitrovic, A. *International Journal on Artificial Intelligence in Education*, 10(3-4), 1999, pp. 238-256.

## References

---

- Mitrovic, A., Mayo, M., Suraweera, P and Martin, B. In: L. Monostori, J. Vancza and M. Ali (eds).(2001) Constraint-based tutors: a success story , *In Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, LNAI 2070, pp. 931-940.
- Mitrovic, A. (2002). NORMIT, a Web-enabled tutor for database normalization. In: Kinshuk, R. Lewis, K. Akahori, R. Kemp, T. Okamoto, L. Henderson, C-H Lee (eds). *In Proceedings of ICCE 2002*, Auckland, pp. 1276-1280.
- Mitrovic, A., Martin, B., Mayo, M (2002).Using evaluation to shape ITS design: Results and experiences with SQL-Tutor. *User Modelling and User Adapted Interaction* 12(2-3), pp. 243-279.
- Mitrovic, A. (2003). An intelligent SQL-Tutor on the Web. *Artificial Intelligence in Education*, 13(2-4), pp. 173-197.
- Mitrovic, A., Martin, B., Suraweera, P. (2007). Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems*, special issue on *Intelligent Educational Systems*, 22(4), pp. 38-45.
- Mitrovic, A., Martin, B. Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N. (2009). ASPIRE: An Authoring System and Deployment Environment for CBTs. *International Journal Artificial Intelligence in Education*, 19(2), pp. 155-188.
- Murray, T. (2003) An Overview of Intelligent Tutoring System Authoring Tools: Updated Analysis of the State of the Art. In T. Murray, S. Blessing, & S. Ainsworth (Eds.) *Authoring Tools for Advanced Technology Learning Environments*, pp. 491-545.
- Ohlsson, S. (1994). Constraint-based Student Modelling. In J. Greer, G. McCalla (Eds.) *Student Modelling: the Key to Individualized Knowledge-based Instruction*, pp. 167-189.
- Ohlsson, S. (1996). Learning from performance errors. *Psychological Review*, 103(2), pp. 241-262.

## References

---

- Ong, J. & Ramachandran, S. (2003). Intelligent tutoring systems: Using AI to improve training performance and ROI. Stottler Henke Associates. [http://www.stottlerhenke.com/papers/ITS\\_Using\\_AI\\_to\\_improve\\_training\\_performance\\_and\\_ROI.pdf](http://www.stottlerhenke.com/papers/ITS_Using_AI_to_improve_training_performance_and_ROI.pdf). Last accessed on 14 March 2012.
- Ravan, S. (2007). MIS for Palm Oil. RANN Consulting. <http://www.rann-consulting.com>. Last accessed on 14 March 2012.
- Risco, S. & Reye, J. (2009). Personal Access Tutor-Helping students to learn MS- Access. *In Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pp. 541-548.
- Ritter, S. & Koedinger, K.R. (1996). An architecture for plug-in tutor agents. *Artificial Intelligence in Education*, 7(3-4), pp. 315-347.
- Self, J. (1990). Bypassing the intractable problem of student modelling. *In Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education*, pp. 107-123.
- Shatz, S., Bowers, C. & Nicholson, D. (2009). Conceptual and Architectural Design Process of an SBT-AID Training System. *In Proceedings of the 18<sup>th</sup> Conference on Behavior Representation in Modeling and Simulation*. pp. 145-146.
- Shute, V.J. & Psotka, J. (1994). Intelligent Tutoring Systems: Past, Present and Future. Armstrong Lab Brooks AFB TX Human Resources Directorate. <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA280011>. Last accessed 14 March 2012.
- Suraweera, P. and Mitrovic, A. (2004). An Intelligent Tutoring System for Entity Relationship Modeling. *International Journal on Artificial Intelligence in Education*, 14(3-4), pp. 375-417.
- Suraweera, P., Mitrovic, A, Martin, B.(2004) The role of domain ontology in knowledge acquisition for ITSs. In: J. Lester, R. M. Vicari and Fabio Paraguacu (eds) *In Proceedings of the 7th International Conference of. Intelligent Tutoring Systems*, LNCS 3220, pp. 207-216.

## References

---

- Suraweera, P., Mitrovic, A., Martin, B. (2005). A Knowledge Acquisition System for Constraint-Based Intelligent Tutoring Systems. In: C-K Looi, G. McCalla, B. Bredeweg, J. Breuker (eds). *In Proceedings of Artificial Intelligence in Education AIED 2005*, pp. 638-645.
- Suraweera, P., Mitrovic, A. & Martin, B. (2007). Constraint Authoring System: an empirical evaluation. R. Luckin, K. Koedinger, J. Greer (eds). *In Proceedings of the 13th International Conference on Artificial Intelligence in Education AIED 2007*, Los Angeles, pp. 451-458.
- Suraweera, P, Mitrovic, A. & Martin, B. (2010) Widening the Knowledge Acquisition Bottleneck for Constraint-based Tutors. *International Journal of Artificial Intelligence in Education*.,vol 20 Issue 2010, pp. 137-173.
- Tan, K.T., (2003). Challenges Facing Plantation Management. *In Proceedings of the International Planters Conference*, pp. 45-54.
- Tecuci, G. & Keeling, H. (1999). Developing an Intelligent Educational Agent with Disciple, Tecuci, Gheorghe & Keeling Harry *.International Journal of Artificial Intelligence in Education* (1999), 10, pp. 221-237.
- Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R. H., Taylor, L., Treacy, D. J., Weinstein, A., and Wintersgill, M. C. (2005). The Andes physics tutoring system: Lessons Learned. *Artificial Intelligence in Education*, 15, pp. 147-204.
- Wang, D., Li, T., Zhu, S., and Gong, Y. (2010), "ihelp: An intelligent online helpdesk system," *Systems, Man, and Cybernetics, PartB: Cybernetics*, IEEE Transactions on, no. 99, pp. 1–10.
- Weng, C.K., (1999). Towards Higher Yield Potential, Production and its Prediction in Oil Palm. PORIM Bulletin Number 39. MPOB Publications.
- Woolf, B., Blegen, D., Jansen, J., and Verloop, A., (1986). Teaching a Complex Industrial Process, *Proceedings of National Association of Artificial Intelligence*, Philadelphia, PA.

## References

---

Woolf, B. & Cunningham, P. (1987). Building a community memory for intelligent tutoring systems. *In Proceedings of AAAI-87*, pp. 82-87.



# APPENDIX A: INFORMATION SHEET



Telephone: +6017-4603645  
Email: [sagaya.amalathas@pg.canterbury.ac.nz](mailto:sagaya.amalathas@pg.canterbury.ac.nz)  
September, 2011

## Framework for Embedded Intelligent Tutoring Systems Information Sheet for Participants

I am a senior PhD student at the College of Computer Science and Software Engineering at the University of Canterbury, New Zealand. As part of my PhD research project, I have developed an Intelligent Tutoring System called DM-Tutor. DM-Tutor has been embedded within a Management Information System (MIS) currently being used to manage oil palm plantations in Malaysia. The objective of embedding DM-Tutor with MIS is to provide scenario based training using real live operational data in the MIS. I would like to invite you to participate in my study. If you agree to take part you will be asked to do the following:

- Complete a pre-test and post-test on palm oil plantation analyses
- Interact with the system for a duration of 120 minutes
- Complete a questionnaire on your experience of using DM-Tutor

Please note that participation in this study is voluntary. If you do participate, you have the right to withdraw from the study at any time. If you do withdraw from the study, I will remove any information relating to you. If you withdraw from the study it will not have any adverse effect on you or your studies. I will take particular care to ensure the confidentiality of all data gathered for this study. Every participant will be assigned a user code and this anonymity will be maintained during and after the study. I will also take care to ensure your anonymity in publications of the findings. All the data will be securely stored in password protected facilities and locked storage at the University of Canterbury for 10 years following the study. It will then be destroyed.

The results of this research may be used to revise and improve DM-Tutor effectiveness in providing training. The results will also be reported internationally at conferences and in journals. All participants will receive a report on the study, if they so wish. If you have any questions about the study, please contact me (details above). If you have a complaint about the study, you may contact either me, Sagaya Amalathas ([sagaya.amalathas@pg.canterbury.ac.nz](mailto:sagaya.amalathas@pg.canterbury.ac.nz)) or Professor Tanja Mitrovic ([tanja.mitrovic@canterbury.ac.nz](mailto:tanja.mitrovic@canterbury.ac.nz)), who is my supervisor. The study has been approved by the University of Canterbury Human Ethics Committee. If you agree to participate in this study, please complete and sign the attached consent form and return it to me.

I do not have any relationship with Putra University Malaysia (UPM). I am requesting UPM's participation as they conduct programs related to the oil palm plantation domain.

I am looking forward to working with you and thank you in advance for your contributions.

Sagaya Amalathas

# APPENDIX B: CONSENT FORM



Telephone: +6017-4603645  
Email: [sagaya.amalathas@pg.canterbury.ac.nz](mailto:sagaya.amalathas@pg.canterbury.ac.nz)  
September, 2011

## Framework for Embedded Intelligent Tutoring Systems

### Consent Form for Participants

I have been given a full explanation of this project and have been given an opportunity to ask questions.

I understand what will be required of me if I agree to take part in this project.

I understand that my participation is voluntary and that I may withdraw at any stage. I understand that if I withdraw from the study any information pertaining to me will be removed completely. I understand that withdrawal from the study will not have any adverse effect on me or my studies in UPM.

I understand that any information or opinions I provide will be kept confidential to the researcher and that any published or reported results will not identify me.

I understand that data related to the study will be stored for 10 years and will be destroyed after that.

I understand that if I wish, I may receive a report on the findings of this study. I have provided my email details for this.

I understand that a PhD is a public document and can be accessed via the University of Canterbury library database.

I understand that if I require further information I can contact the researcher; Sagaya Amalathas (contact details above). If I have any complaints, I can contact Professor Tanja Mitrovic ([tanja.mitrovic@canterbury.ac.nz](mailto:tanja.mitrovic@canterbury.ac.nz)). By signing below, I agree to participate in this research project.

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Signature: \_\_\_\_\_

Email: \_\_\_\_\_

Please return this completed consent form to Sagaya Amalathas.

## APPENDIX C: HEC APPROVAL



HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen  
Email: [human-ethics@canterbury.ac.nz](mailto:human-ethics@canterbury.ac.nz)

Ref: HEC 2011/84

23 September 2011

Sagaya Amalathas  
Department of Computer Science and Software Engineering  
UNIVERSITY OF CANTERBURY

Dear Sagaya

The Human Ethics Committee advises that your research proposal "Framework for Embedded Intelligent Tutoring Systems" has been considered and approved.

Please note that this approval is subject to the incorporation of the amendments you have provided in your email of 19 September 2011.

Best wishes for your project.

Yours sincerely

A handwritten signature in black ink, appearing to read 'Michael Grimshaw', written in a cursive style.

Michael Grimshaw  
*Chair*  
*University of Canterbury Human Ethics Committee*

# APPENDIX D: PRE-TESTS AND POST-TESTS

## PILOT STUDY FEBRUARY 2011

### Pre-Test

1. For Ladang Sungai Merchong, under the region of TH Plantations (12696), identify the field with the highest yield gap for the month of February, 2006. Calculate the yield gap.

Answer :

Potential yield:

Yield	Yield Gap

2. For Ladang Sungai Merchong, from the MIS we know that the palm trees in this estate are 25 years old. Select suitable fertilizer placement for the following fertilizer combinations:

Urea	a) weeded circle	b) around circle	c) outside circle
Borate	a) weeded circle	b) around circle	c) outside circle
MgKieserite	a) weeded circle	b) around circle	c) outside circle

3. For Ladang Sungai Merchong, based on yield value for May 2006, forecast yield for November 2006.

Yield May 2006	Forecasted Yield November 2006

## PILOT STUDY FEBRUARY 2011

---

### Post-Test

1. For Ladang Ulu Chukai, under the region of TH Plantations (12696), identify the field with the highest yield gap for the month of February, 2006. Calculate the yield gap

Answer:

Potential yield:

Yield	Yield Gap

For Ladang Ulu Chukai, from the MIS we know that the palm trees in this estate are 15 years old. Select suitable fertilizer placement for the following fertilizer combinations:

Urea	a)weeded circle	b) around circle	c)outside circle
Borate	a) weeded circle	b) around circle	c)outside circle
MgKieserite	a)weeded circle	b) around circle	c)outside circle

For Ladang Ulu Chukai, based on yield value for January, 2006 forecast yield for January, 2007.

Yield January 2006	Forecasted Yield January 2007

## Evaluation October 2011

### Test 1

**Question 1:** As an estate manager you need to identify potential yield for estate Ladang Bangka Ulu for the month of August 2008. Estate Ladang Bangka Ulu has three fields. For the given month calculate yield gap.

Answer :

Potential yield:

Yield	Yield Gap

**Question 2:** The palm trees in estate Ladang Bangka Ulu are 3 years old. For the month of August 2008, select suitable fertilizer placement for the following fertilizer combinations:

Urea	a) weeded circle	b) around circle	c) outside circle
Borate	a) weeded circle	b) around circle	c) outside circle
MgKieserite	a) weeded circle	b) around circle	c) outside circle

**Question 3:** For estate Ladang Bangka Ulu based on yield value for August 2008 forecast yield for August, 2009.

Yield August 2008	Forecasted Yield August 2009

## Evaluation October 2011

### Test 2

**Question 1:** As an estate manager you need to identify potential yield for estate Ladang Bukit Belian for the month of April 2007. Estate Ladang Bukit Belian has 3 fields. For the given month calculate yield gap.

Answer :

Potential yield:

Yield	Yield Gap

**Question 2:** The palm trees in estate Ladang Bukit Belian are 10 years old. For the month of April 2007, select suitable fertilizer placement for the following fertilizer combinations:

Urea	a) weeded circle	b) around circle	c) outside circle
Borate	a) weeded circle	b) around circle	c) outside circle
MgKieserite	a) weeded circle	b) around circle	c) outside circle

**Question 3:** For estate Ladang Bukit Belian based on yield value for April 2007 forecast yield for April, 2009.

Yield April 2007	Forecasted Yield April 2009

# APPENDIX E: QUESTIONNAIRES

## Questionnaire DM-TUTOR Pilot Study

**DM-Tutor** – Intelligent Tutoring System (ITS) for training on palm oil plantation decision making.

Thank you for participating in our pilot study—your feedback is crucial to our current and future research. This questionnaire is anonymous and if you wish, you may at any time withdraw your participation, including withdrawal of any information you have provided. However, by completing the questionnaire, you indicate your consent for publication of the generalised results of our research findings.

USER ID :

1. What is your current job designation:

2. How long have you worked with palm oil plantation:

3. What are the different type of roles you have held in palm oil plantations?

4. Have you used any palm oil plantation management software before?

Yes	<input type="checkbox"/>	No	<input type="checkbox"/>
-----	--------------------------	----	--------------------------

(Please tick (✓) where appropriate)



5. If you answered 'Yes' for question number (4.), please state the duration you have used Plantation Management software and for what purpose you have used the software for:



6. Have you used this Management Information System (MIS) before?

Yes		No	
-----	--	----	--


7. What are the analyses that you attempted to answer in DM-Tutor:

--


8. Did you think that the questions in DM-Tutor are easy to attempt?

Not at all		Very much		
1	2	3	4	5

9. Did you think that DM-Tutor is easy to use?

Not at all		Very much		
1	2	3	4	5

10. Did you think that the feedback messages in DM-Tutor was helpful in answering the questions?

Not at all		Very much		
1	2	3	4	5

11. Did you like the interface of DM-Tutor? Please elaborate your answer.

Yes		Somewhat		No	

12. Was DM-Tutor able to teach you any new analyses in palm oil plantation decision making? Please elaborate your answer.

Yes		Somewhat		No	

13. Do you think that by integrating a Management Information System (MIS) with DM-Tutor, you were able to learn these plantation analyses better?

Yes		Somewhat		No	

14. Did you think that DM-Tutor was interesting to use:

Yes		Somewhat		No	

## QUESTIONNAIRE OCTOBER 2011

**DM-Tutor** – Intelligent Tutoring System (ITS) for training on palm oil plantation decision making.

Thank you for participating in our evaluation study—your feedback is crucial to our current and future research. This questionnaire is anonymous and if you wish, you may at any time withdraw from participation, including withdrawal of any information you have provided. However, by completing the questionnaire, you indicate your consent for publication of the generalised results of our research findings.

USER ID:

1. Please explain your current level of study?

2. Do you have experience working in any oil palm plantation?

Yes		Somewhat		No	
<input type="text"/>					

3. Have you used any oil palm plantation management software before?

Yes		Somewhat		No	
<input type="text"/>					

(Please tick (✓) where appropriate)

4. If you answered ‘Yes’ for question number (4.), please state the duration you have used oil palm plantation management software and for what purpose you have used the software for:

Software used duration:

Purpose used for:

5. What are the analyses that you attempted in this study:

6. Did you think that the questions are easy to attempt?

Not at all				Very much
1	2	3	4	5

7. Did you think that the system is easy to use?

Not at all				Very much
1	2	3	4	5

8. Did you think that the feedback messages in DM-Tutor were helpful in answering the questions?

Not at all				Very much
1	2	3	4	5

9. Did you like the interface of DM-Tutor? Please elaborate your answer.

Not at all				Very much
1	2	3	4	5

10. Was the system able to teach you any new analyses in palm oil plantation decision making? Please elaborate your answer.

Yes		Somewhat		No	

11. Do you think that by integrating a Management Information System (MIS) with DM-Tutor, you were able to learn these plantation analyses better? Please elaborate your answer.

Yes		Somewhat		No	

12. Did you think that DM-Tutor was interesting to use? Please elaborate your answer.

Yes		Somewhat		No	

# APPENDIX F: PROBLEMS ATTEMPTED BY THE CONTROL GROUP DURING FULL EVALUATION STUDY

Thank you for your participation. This assessment is anonymous and if you wish, you may at any time withdraw from participating, including withdrawal of any information you have provided. However, by participating in this assessment, you indicate your consent for publication of the generalised results of our research findings. We kindly request you to try to attempt all the questions in this assessment.

USER ID:

## Yield Gap Analysis

**Question 1:** For Estate Ladang Sg Mengah, identify Potential Yield for the month of August 2008. Calculate Yield Gap for each field. Estate Ladang Sg Mengah has been experiencing yield losses over several years due to aging palm trees. Give your recommendation for this condition.

Select the correct formula to calculate yield gap

$$(Y_p - Y_a)$$

Bunch yield/ Fertiliser nutrient

Yield \* Plantation Size

$$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$$

CPO/FFB

FFB/ Hectare

**Solution Workspace**

Yield for each field:


Potential Yield for Estate

Calculate Yield Gap for each field


**Solution Workspace**

Field with highest yield gap

Your recommendations to improve yield at field

**Question 2:** Ladang Bukit Lawiang consists of 8 fields. For the month of August 2007, identify Yield Potential for this estate. Calculate Yield Gap for each field to identify field with highest yield gap. Ladang Bukit Lawiang has been experiencing losses due to excess rain water on the plantation ground. Give your suggestion on how yield can be improved in this condition

Select the correct formula to calculate yield gap

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace**

Yield for each field:

Potential Yield for Estate

Calculate Yield Gap for each field




**Solution Workspace**

Field with highest yield gap

Your recommendations to improve yield at field

**Question 3:** Ladang Gunung Sumalayang consists of 5 fields. For the month of March 2008, identify potential yield (YP) for this estate. Calculate yield gap for each field to identify field with highest yield gap. In estate Ladang Gunung Sumalayang empty fruit bunches have prevented access to FFB. Give your recommendation to improve YP in this condition

Select the correct formula to calculate yield gap

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace**

Yield for each field:


Potential Yield for Estate

Calculate Yield Gap for each field


**Solution Workspace**

Field with highest yield gap

Your recommendations to improve yield at field

## Fertilizer Management

**Question 1:** Partial Factor Productivity or PFP is used to identify how much yield is produced for each kg fertilizer used and Agronomic Efficiency or AE is used to identify how much additional yield is produced for each kg of fertilizer nutrient applied for an estate.. Calculate PFP and AE for Estate Bukit Lawiang for the month of April 2006. Given yield for that period is 2.27 MT/ha and yield without fertilizer is 1.10 MT/ha. Identify palm age and select suitable fertilizer placement and give supporting reason for each selection. Transport of fruit bunches and fertilizer has compacted the soil structure and created a condition of low infiltration of the fertilizer. Give your recommendation to improve condition

Select the correct formula to calculate yield gap

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

Solution Workspace :

For Estate Bukit Lawiang, the age of palm trees within the field is 20 years. Give your recommendation for a suitable fertilizer combination and placement. Give reasons to support your answer.

Solution Workspace :

Palm Age

<b>Fertilizer Type</b>	<b>Fertilizer Placement</b>	<b>Reason</b>
N (Urea, SOA)	Outside circle; Around circle; Inside circle	
P (Rock Phosphate)	Outside circle; Around circle; Inside circle	
K (KCL)	Outside circle; Around circle; Inside circle	
Mg (Kieserite)	Outside circle; Around circle; Inside circle	
B (Borate)	Outside circle; Around circle; Inside circle	

Recommendation:

**Question 2.** Partial Factor Productivity or PFP is used to identify how much yield is produced for each kg fertilizer used and Agronomic Efficiency or AE is used to identify how much additional yield is produced for each kg of fertilizer nutrient applied for an estate. For the month of August 2008, Calculate PFP and AE for Estate Ladang Sungai Buan. Given yield for that period is 1.90 and yield without fertilizer is 0.5. Identify palm age for estate Ladang Sungai Buan. Identify age of palm trees in the estate select suitable fertilizer placement and give reasons to support your selections. Estate Ladang Sungai Buan has a condition where nutrient applied was converted into a non absorbable form due to soil reaction. Give your recommendation to improve this condition

**Solution Workspace :**

Select the correct formula to identify PFP:

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$		CPO/FFB
		FFB/ Hectare

**Solution Workspace :**

For **Estate Bukit Lawiang**, the age of palm trees within the field is 20 years. Give your recommendation for a suitable fertilizer combination and placement. Give reasons to support your answer.

**Solution Workspace :**

Palm Age

Fertilizer Type	Fertilizer Placement	Reason
N (Urea, SOA)	Outside circle; Around circle; Inside circle	
P (Rock Phosphate)	Outside circle; Around circle; Inside circle	
K (KCL)	Outside circle; Around circle; Inside circle	
Mg (Kieserite)	Outside circle; Around circle; Inside circle	
B (Borate)	Outside circle; Around circle; Inside circle	

Recommendation:

**Question 3:** Partial Factor Productivity or PFP is used to identify how much yield is produced for each kg fertilizer used and Agronomic Efficiency or AE is used to identify how much additional yield is produced for each kg of fertilizer nutrient applied for an estate

For Ladang Ulu Chukai, calculate PFP and AE for the month of August 2008, given yield for that period is 1.57 Mt/ ha and yield without fertilizer is 0.12MT/ha. Identify palm age and select suitable fertilizer placement for the estate. Give reasons to support your answer. Ladang Ulu Chukai faces the problem where Nitrogen is found in the water collected 10m from the palm trees. Select suitable recommendation to solve the problem.

**Solution Workspace :**

Select the correct formula to identify PFP:

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace :**

For **Estate Bukit Lawiang**, the age of palm trees within the field is 20 years. Give your recommendation for a suitable fertilizer combination and placement. Give reasons to support your answer.

**Solution Workspace :**

Palm Age

Fertilizer Type	Fertilizer Placement	Reason
N (Urea, SOA)	Outside circle; Around circle; Inside circle	
P (Rock Phosphate)	Outside circle; Around circle; Inside circle	
K (KCL)	Outside circle; Around circle; Inside circle	
Mg (Kieserite)	Outside circle; Around circle; Inside circle	
B (Borate)	Outside circle; Around circle; Inside circle	

Recommendation:

### **Yield Forecasting**

**Question 1:** Given yield for October 2006 for Estate Ladang Bkt Belian as 2.60 MT/ha, Estate Belian suffered the problem when FFB arrived at mill, they were found to be overripe making yield lower than forecasted. Select relevant information to help you to identify the problem. Give your recommendation for a bigger yield gain

Select the correct formula for yield forecasting

### **Solution Workspace :**

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace :**

Information

Recommendation for a bigger yield gain:

--

**Question 2:** Ladang Bangka Ulu has 2 estates. Estate Ladang Bukit Rokan has been experiencing flooding of 5cm rain water in low lying sections of the field. Select relevant information to identify this problem. Give appropriate recommendation for bigger yield gains.

Select the correct formula for yield forecasting

**Solution Workspace :**

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace :**

Information

Recommendation for a bigger yield gain:

--



**Question 3:** For Field 86B of Estate Ladang Gunung Sumalayang, yield has been declining over a period of a year. Select information that will help you to identify the problem. Give recommendation to help gain bigger yield.

Select the correct formula for yield forecasting

**Solution Workspace :**

$(Y_p - Y_a)$	Bunch yield/ Fertiliser nutrient	Yield * Plantation Size
$(\text{Bunch yield}_{+FN} - \text{Bunch yield}_{0FN}) / \text{Fertiliser Nutrient}$	CPO/FFB	FFB/ Hectare

**Solution Workspace :**

Information

Recommendation for a bigger yield gain:

--

## APPENDIX G: PUBLISHED PAPERS

1. Amalathas, S., Mitrovic, A. & Ravan, S. (2009) Towards an ITS for decision making on managing palm oil Plantations. *Workshop Proceedings of the 17th International Conference on Computers in Education (ICCE 2009)*, Hong Kong. APSCE, pp 157-162.
2. Amalathas, S., Mitrovic, A. & Ravan, S. (2010) Towards a framework for embedded ITSs. *DSC Proceedings of the 18th International Conference on Computers in Education ICCE 2010*, Malaysia, APSCE, pp 1-5.
3. Amalathas, S., Mitrovic, A. & Ravan, S. (2010) Intelligent Tutoring System for the Palm Oil Industry. *Proceedings of the International Oil Palm Conference (IOPC) 2010*, Indonesia, IOPRI, pp 245-267.
4. Amalathas, S., Mitrovic, A., Ravan, S & Evison, D. (2011). Evaluation of DM-Tutor, an ITS for Training on Plantation Decision-Making. *Workshop Proceedings of the 19th International Conference on Computers in Education ICCE 2010*, Malaysia, APSCE, pp 1-5.

# Towards an ITS for Decision Making on Managing Palm Oil Plantations

Sagaya AMALATHAS<sup>1</sup>, Antonija MITROVIC<sup>1</sup>, Ravan SARAVANAN<sup>2</sup>

<sup>1</sup>*Department of Computer Science, University of Canterbury, New Zealand*

<sup>2</sup>*RANN Consulting, Malaysia*

ssa106@student.canterbury.ac.nz

**Abstract:** Although intelligent tutoring systems (ITSs) have proven their effectiveness, very few attempts have been made to embed ITSs into existing applications. In this paper, we describe the design of an Intelligent Tutoring System that will be embedded within an active Management Information System (MIS). With the ITS embedded within the MIS, users will be presented with real-life management scenarios and practice with actual operational data. This would help them improve their decision making skills and help them make more effective decisions in their work area. We discuss the architecture and use-case scenario for DM-Tutor (Decision Making-Tutor). We also include plans for system evaluation and future work.

**Keywords:** Embedded ITS, decision making, architecture, use-case scenario

## Introduction

Intelligent Tutoring Systems have been very successful [4] and have resulted in high levels of motivation and learning in many areas researched [11]. ITSs have been build in various teaching and learning domains, SQL-Tutor [10], KERMIT [15], CAPIT [7], CIT [9], UML Tutor [1] and many others. However to date, very few attempts have been made to integrate or embed ITSs into existing systems. This area of research has a lot of potential, especially in increasing the effectiveness of life-long education and training at workplaces [12]. As the saying goes, systems are only as good as the people that use them. It is rather pointless in having good information systems if the users of the systems are not able to fully utilize them. ITS features are ideal for providing tutoring or training to one or many people at the same time. Where other training methodologies such as CBT (computer-based training), WBT (web-based training) and multimedia-based training have encountered limitations in this area [2], ITS could still have the advantage of being able to provide continuous and effective work-place trainings to users and organizations.

Some of the attempts made to embed intelligent learning techniques into existing systems include the Macsyma Advisor [5] which was developed to assist users in using Macsyma, the algebraic manipulation system. Macsyma Advisor focused on assisting rather than

training, which meant the user may not have learned how to solve problems unassisted in future. Embedded Training System (ETS) [3] was developed and integrated with an existing Complex Information System for military tasks and operations. The goal of ETS was to train users on military based scenarios. ETS however was not robust enough to handle variety of student's actions and behaviors within the system.

Excel Tutor [6] was developed to work on top of Microsoft Excel Spreadsheet software program. Excel Tutor provided descriptive explanations and interactive guidance for students to solve excel problems. Excel Tutor provided a higher learning outcome for students. Another similar research was Personal Access Tutor (PAT) [14] which was added into MS Access to help students build reports and forms. Although the students seemed to like PAT, the system has not been fully evaluated.

This paper presents the design and architecture of DM-Tutor (Decision Making-Tutor) that will be embedded into a MIS (Management Information System) to train users on effective decision making. The MIS for palm oil industry [13] provides an integrated software solution for managing oil palm plantation operations. The MIS modules provide for resources management, finance and accounting management and other management concerns. The aim of the MIS is to achieve management objectives of optimum profit and balanced operations costs. Managers carry out yield and cost related analyses using the operational data contained within the MIS database. As the information is highly domain-specific and requires detailed knowledge of the domain and the MIS, managers who are new to the domain or to the MIS face difficulties in making accurate and well defined analyses. This in turn affects the management and operational decisions they would make.

The paper is organized as follows. In the next section we discuss the architecture of DM-Tutor. Section 2 discusses the use-case scenario and section 3 presents conclusion and future work.

## **1. DM-Tutor**

The architecture of DM-Tutor is unique because it presents an ITS embedded within an actively used system, the MIS. Figure 1 presents the architecture of the MIS with DM-Tutor and its components embedded within it. The MIS [13] by itself is a web based system and is accessed via the web browser. When user logs into DM-Tutor for the first time, s/he gets a brief description of the system and the problem solving environment it provides. A log is used to record users' actions in DM-Tutor.

*Student Model:* Constraint based modeling (CBM) [8] approach is used to model the student. Student is modeled by looking through the student's solution and comparing student's solution to the ideal solution. Student model contains student profile that is updated every time s/he uses the ITS. Progress is tracked and the area that s/he requires more help in is identified and retained by the system.

*Pedagogical Module* contains contextualized instructions relevant to the scenario-based problem solving strategy, potential solutions and relevant hints to guide students/users. The pedagogical module interacts closely with the MIS database to provide view of operational data in relation to the scenario-based problem solving. Pedagogical module

observes student's action in the interface and acts to it accordingly, by giving appropriate feedback.

*Domain Model* is represented in the form of constraints [8]. A Semantic Portal developed for the palm oil domain provides additional information in various forms. Using semantic technology, the portal gathers useful and relevant information from the WWW to provide helpful information; based on DM-Tutor user's previous query or doubts. The gathered information would be made available to the user when user logs in the next time.

*Interface Module* for DM-Tutor is unique as it presents the integration within the MIS system. DM-Tutor's users are able to view MIS operational data to simulate real plantation management decision making. The main window of DM-Tutor is divided into 4 areas.

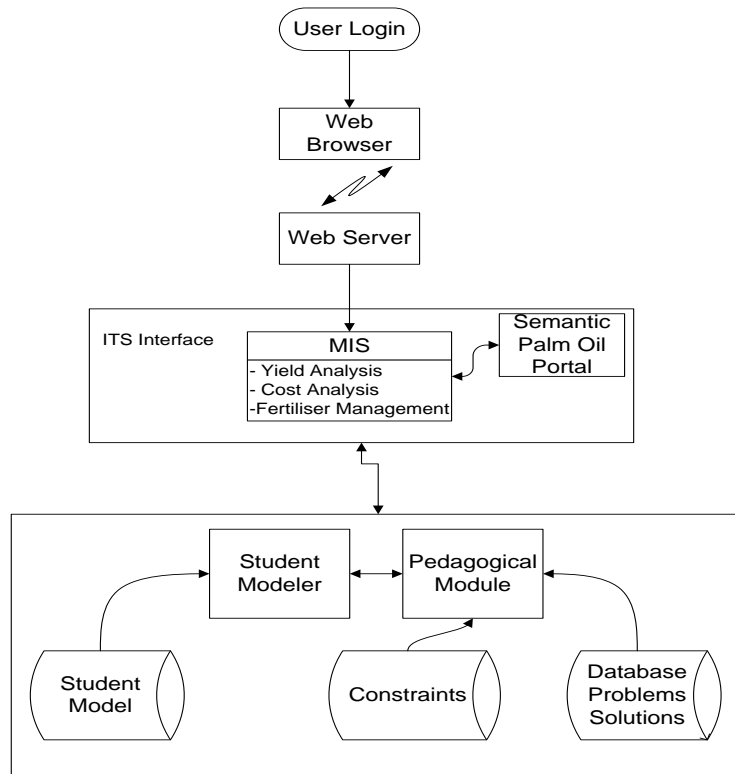


Figure 1: Architecture and main components of DM-Tutor

The top pane displays the given problem; middle pane is workspace area, bottom pane displays relevant view of MIS database and a side pane to display DM-Tutor feedback messages. This design is aimed at reducing memory load of students. Interface module handles communication between system and user.

## 2. DM-Tutor Use Case Scenario

The objective of DM-Tutor is to train users on making effective decisions related to their management area. For palm oil industry managers to make good yield related decisions

they have to first know how to do effective yield analysis. Below is a set of example questions DM-Tutor would provide to users and appropriate user/student actions for this scenario.

*Question 1: Identify yield for estate: Ladang Bukit Lawiang*

The bottom pane of the working area displays screen shot of the yield table to determine if the user can identify yield for the palm oil estate for a given time period (as the first step). Figure 2 shows a screenshot of the MIS yield table, where yield is labeled as *Fresh Fruit Bunch in Metric Tonne (FFB(MT))*. Student has to enter value of the yield in the workspace provided.

*Question 2: List the factors that contribute to yield production in an estate*

User has to list the factors contributing to yield in the workspace area provided. The factors include rainfall, fertilizer consumption, labor utilization and vehicle usage. This question is to gauge the level of understanding the user has of the domain and of the MIS.

ESTATE - YIELD ANALYSIS FOR THE MONTH OF NOVEMBER, 2006						
LADANG BUKITLAWIANG						
	THIS MONTH	THIS MONTH EST.	VARIANCE THIS MONTH	TO DATE	TO DATE EST.	VARIANCE TO DATE
Total Hectare(M,IM,R,NY)	1,989.00	1,989.00	0.00	1,989.00	1,989.00	0.00
Days Offered	26.00	0.00	0.00	0.00	0.00	0.00
Mature Hectare	1,989.00	1,989.00	0.00	1,989.00	1,989.00	0.00
Stands	262,548.00	0.00	0.00	0.00	0.00	0.00
Stand/Hectare	132	0	0	0	0	0
Rainfall(mm)	0.00	0.00	0.00	0.00	0.00	0.00
FFB(MT)	4,730.68	4,351.00	379.68	48,581.54	42,078.00	6,503.54

Figure 2: Sample Yield Table.

*Question 3: Compare the yield of different fields of the estate. Estimate potential yield of the estate and calculate yield gap for each field*

As plantation estates are made up of several fields and management actions are mostly carried out at field levels, user has to access the actual yield ( $Y_a$ ) data of relevant fields. Figure 3 shows screenshot of yield map from where user can obtain the yield data of different fields and compare the yield for several production periods. Yield at above 90th percentile of the estate production is considered as potential yield ( $Y_p$ ). User has to list both field number and yield in the workspace provided. Next, user has to estimate potential yield of the estate and fill in the box provided for potential yield. Lastly, user has to calculate yield gap (using the formula below) for each field and enter this information as well into the provided workspace, using the following formula:  $Yield\ gap = Y_p - Y_a$

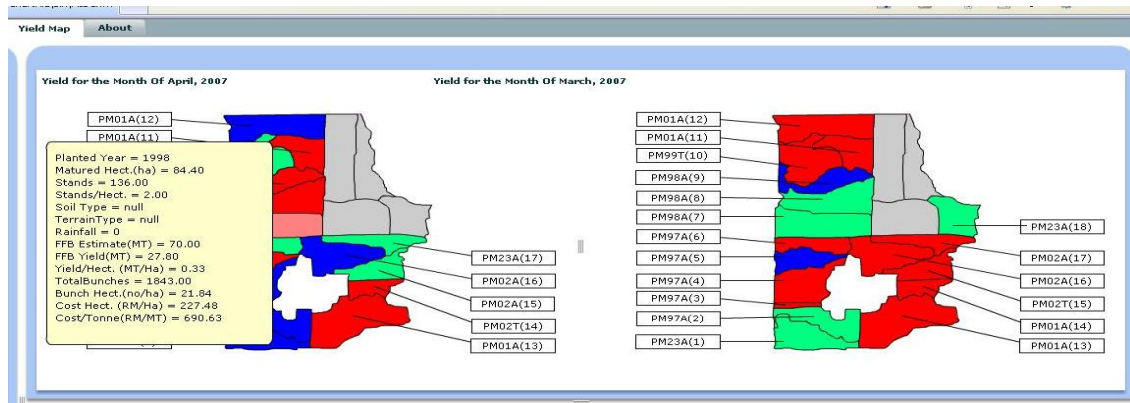


Figure 3: Yield comparison between fields within an estate.

*Question 4: Identify the low performing field(s) and recommend steps to improve yield of the field(s)*

Where yield gap is highest, the field is considered as low-performing and that is the field where management action is most needed to improve yield. As the final step, the user has to identify and name the field(s) that is/are not performing well and in the workspace provided give his/her suggestions to improve yield for the non-performing field(s), based on the earlier understanding of the factors that influence yield production. DM-Tutor compares the user's solution to the ideal solution contained within DM-Tutor's knowledge base and gauges the level of accuracy of the user's solution and gives feedback to the user. If the user's solution is inaccurate DM-Tutor would provide relevant feedback and guide the user through the accurate steps in yield analysis and decision making.

### 3. Conclusion and Future Work

DM-Tutor is a work-in-progress and when completed would be a very beneficial research in many aspects. This would be the first time where a constraint based tutor is being embedded within an existing system and it also would have a very practical benefit of providing life-long workplace learning anywhere and anytime. We have planned for DM-Tutor to be evaluated by users of different levels: managers in palm oil industries, trainee managers and novices. Upon completion of DM-Tutor we hope to be able to present users' evaluation of our system and our belief of the potential of ITS in increasing workplace efficiency.

### Acknowledgements

The first author thanks RANN Consulting, Malaysia for funding this research.

## References

- [1] Baghei, N., Mitrovic, A. and Irwin, W. (2007). Supporting Collaborative Learning and Problem Solving in a Constraint-based CSCL Environment for UML Class Diagrams. *Computer-Supported Collaborative Learning*, 2(2-3), 159-190
- [2] Beck, J., Stern, M. & Haugsjaa, E. (1996). Applications of AI in education. *Crossroads*, 3 (1), ACM Student Magazine. 1528-4972. pp. 11-15
- [3] Cheikes, B.A., Geier, M., Hyland, R., Linton, F., Rodi, L. & Schaefer, H.P. (1998). Embedded training for complex information systems. *Artificial Intelligence in Education*, 10, 314-334.
- [4] Freedman, R. (2000). What is an intelligent tutoring system. *Intelligence*, 11(3), 15-16.
- [5] Genesereth, M.R. (1979). The role of plans in automated consultation. *In Proceedings of the 6<sup>th</sup> International Joint Conference on Artificial Intelligence*, pp. 311-319.
- [6] Mathan, S.A. & Koedinger, K.R. (2002). An empirical assessment of comprehension fostering features in an intelligent tutoring system. *6<sup>th</sup> International Conference: ITS 2002*.LNCS 2363, pp. 330-343.
- [7] Mayo, M. & Mitrovic, A. (2001). Optimising ITS behaviour with bayesian networks and decision theory. *Artificial Intelligence in Education*, 12, 124-153.
- [8] Mitrovic, A. (1998). Experiences in implementing constraint-based modeling in SQL-Tutor. *Proceedings of ITS'98*. pp. 414-423.
- [9] Mitrovic, A., McGuigan, N., Martin, B., Suraweera, P., Milik, N. & Holland, J.(2001).Authoring constraint-based tutors in ASPIRE: a case study of a capital investment tutor. *World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2008)*. pp. 4607-4616.
- [10] Mitrovic, A. (2003). An intelligent SQL-Tutor on the web. *Artificial Intelligence in Education*, 13, 173-197.
- [11] Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *Artificial Intelligence in Education* 10, 98-189.
- [12] Ong, J. & Ramachandran, S. (2003). Intelligent tutoring systems: Using AI to improve training performance and ROI. Stottler Henke Associates. [http://www.stottlerhenke.com/papers/ITS\\_Using\\_AI\\_to\\_improve\\_training\\_performance\\_and\\_ROI.pdf](http://www.stottlerhenke.com/papers/ITS_Using_AI_to_improve_training_performance_and_ROI.pdf)
- [13] Ravan, S. (2007). MIS for Palm Oil. RANN Consulting. <http://www.rann-consulting.com>
- [14] Risco, S. & Reye, J. (2009). Personal Access Tutor-Helping students to learn MS-Access. *In Proceedings of the 14<sup>th</sup> International Conference on Artificial Intelligence in Education*. pp. 541-548
- [15] Suraweera, P. & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modeling. *Artificial Intelligence in Education*, 14, 375-417.



# Towards a Framework for Embedded ITSs

Sagaya AMALATHAS<sup>1\*</sup>, Tanja MITROVIC<sup>1</sup> & Ravan SARAVANAN<sup>2</sup>

<sup>1</sup>*ICTG, University of Canterbury, New Zealand*

<sup>2</sup>*RANN Consulting, Malaysia*

\*sagaya.amalathas@pg.canterbury.ac.nz

**Abstract:** Although Intelligent Tutoring Systems (ITSs) have proven their effectiveness, very few attempts have been made to embed ITSs into existing applications. In this paper, we describe the research, design and progressive development of DM-Tutor (Decision-Making Tutor), the first constraint-based tutor (CBT) to be embedded within an existing system, the Management Information System (MIS) for palm oil plantation management currently being used in Malaysia. We discuss the research and development of DM-Tutor with the help of ASPIRE, an authoring system for CBTs. We also include future work planned for DM-Tutor.

**Keywords:** embedded intelligent tutoring systems, decision making training, management information system

## Introduction

Intelligent Tutoring Systems (ITSs) provide benefits of one-on-one teaching to any number of students automatically and cost effectively through highly interactive environments [3]. Over the years many ITSs including LISP tutor [1], Andes-Physics Tutor [14], PUMP Algebra Tutor [4] and others have been effectively used in many teaching and learning domains. SQL-Tutor [6] and KERMIT [13] are among the many constraint-based tutors (CBT) [7] that have been developed and successfully implemented by the Intelligent Computer Tutoring Group (ICTG). Even though ITSs have been proven as effective teaching tools there have been very few attempts to embed them with other systems, Embedded Training System (ETS) [2], Excel Tutor [5] and Personal Access Tutor (PAT) [11]. Will ITSs embedded within existing systems provide effective instructions? Through this effort we hope to answer that important research question. With this research we aim to make several significant contributions. This will be the first attempt to embed a CBT with an existing system. We also aim to research the significant benefits of providing training through this integration. We aim to develop a framework for embedded ITSs and prove its research contribution through the development of DM-Tutor and its integration with the MIS for palm oil.

This paper presents the research, design and on-going development of DM-Tutor (Decision-Making Tutor), an ITS to train on plantation decision making for the palm oil domain. DM-Tutor will be embedded with a Management Information System (MIS) [10] currently being used to manage palm oil plantations in Malaysia and Indonesia. The

MIS for palm oil contains operational data of yield records and plantation cultivation details. As the information contained is highly domain specific, managers who are new to the domain or to the MIS face difficulties in making accurate operational analyses and this affects the decisions they make at the palm oil plantations. When DM-Tutor is embedded with the MIS for palm oil, students will be able to practice plantation decision making using real life operational data from the MIS. The goal of DM-Tutor is to help students and managers apply theoretical concepts of plantation analyses into real-life plantation decision making. This paper is organized as follows. In the next section we describe DM-Tutor. Section two describes the development of DM-Tutor in ASPIRE [8], an authoring system for CBTs. Section three describes future work planned for DM-Tutor and conclusions.

## 7. DM-Tutor

To the best of our knowledge there has not been an ITS for plantation decision making, DM-Tutor is novel in that respect. Figure 1 presents the overall architecture of DM-Tutor. Our plan is to make DM-Tutor accessible through the MIS interface. The MIS by itself is a web based system and is accessed via a web browser. Students log in to the MIS and then access DM-Tutor through the MIS.

*Student model* contains information of student's knowledge of the domain and is updated every time the student uses DM-Tutor. Constraint based modeling (CBM) [9] is used to model the domain and the student. Student is modeled by looking through her/his solution and comparing it to the ideal solution in DM-Tutor. *Pedagogical module* selects instructions relevant to the scenario-based problem solving strategy used. It also has the role of providing helpful feedback to students when they submit an incorrect solution.

*Interface module* presents the student interface of DM-Tutor. The *problems* and *solutions* component focuses on *Yield Gap Analysis*, *Fertilizer Management* and *Yield Forecasting*, three main analyses of palm oil plantation management. In order to solve problems presented in DM-Tutor, students will have to access relevant reports from the MIS. DM-Tutor would also log information on the MIS reports that students accessed. This will enable DM-Tutor to provide helpful feedback when the student fails to provide correct solution to the problem.

## 8. Development of DM-Tutor

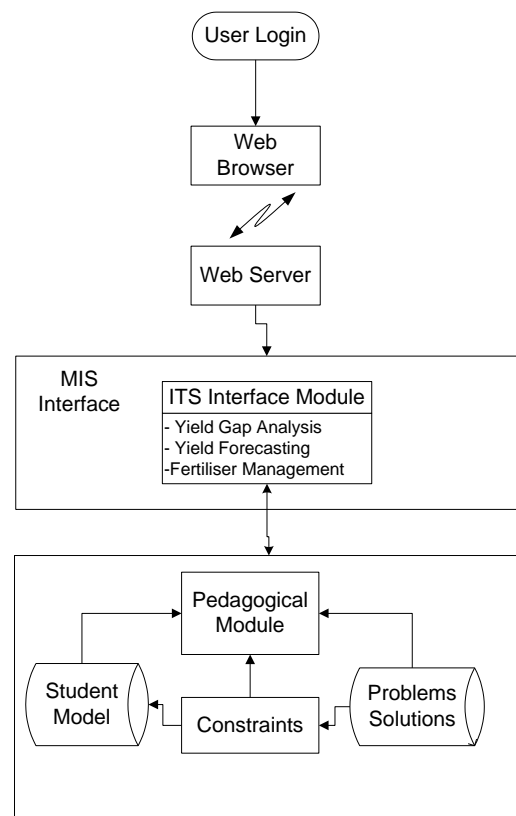


Figure 1: DM-Tutor

DM-Tutor has been developed using ASPIRE, an authoring system for constraint-based tutors. In the first development stage, we identified domain characteristics and problem solving steps required for DM-Tutor. To identify characteristics of the domain, we needed to determine the domain knowledge that we are teaching and the appropriate teaching strategy to be used. We divided the domain into three procedural tasks: *yield gap analysis, fertilizer management and yield forecasting*. In the next stage we developed ontologies for each separate task. Developing domain ontologies was a crucial step in DM-Tutor authoring process as this contributed to building of syntax constraints in the later development stage. In the domain ontologies we identified important concepts for each task and the relationship between these concepts using a hierarchical structure. Each concept could have super-concepts, sub-concepts, properties and relationships with other concepts. Concepts we created in the ontologies are present in the solution structures. By developing the ontologies we were able to study each task within the domain in greater detail and this helped in the development of accurate constraint bases.

In the third step we modeled the problem and solution structures for each task within the domain. For procedural tasks, problem solving is divided into several steps and student's solution is evaluated at every step. Students will not be able to continue to the next problem solving step before submitting the correct solution for the present step. Each solution component comes from the ontology of each task. In the next stage, ASPIRE developed the interface for DM-Tutor. Figure 2 presents the student interface for DM-Tutor. DM-Tutor's student interface is divided into three parts. The top pane is where the problem statement is placed, so that students always know the problem they are attempting. The bottom pane presents the solution workspace where students need to work on their solutions to the problems and the side pane is used to provide feedback to the students on the problems they are attempting. The interface design for DM-Tutor is aimed at reducing memory load of the students. We have planned to replace the current textual interface of DM-Tutor with java applets.

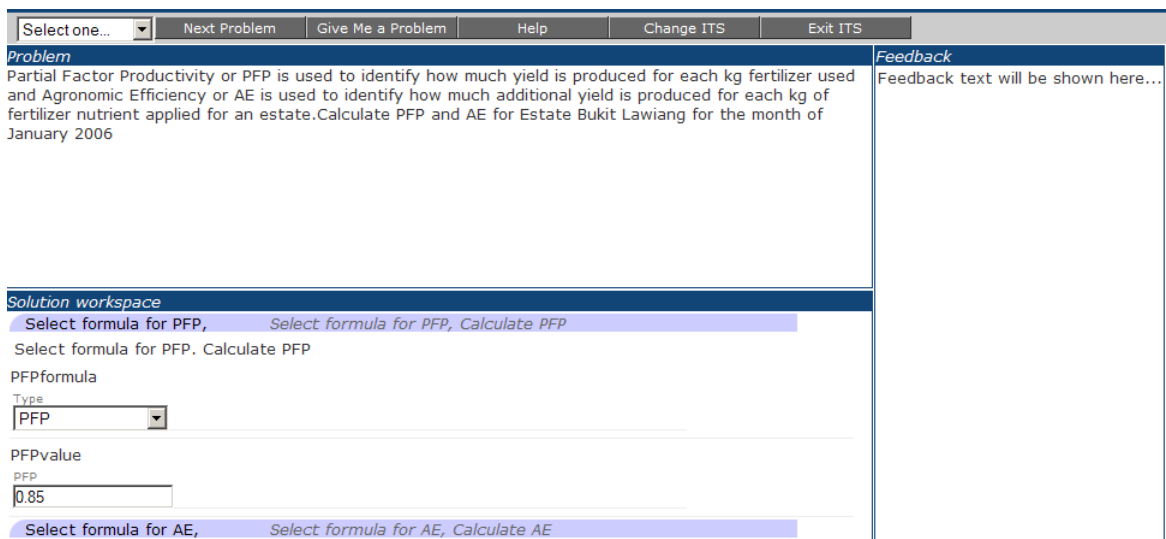


Figure 2: Student interface for DM-Tutor

In the next stage, ASPIRE helped to generate constraints for DM-Tutor. For CBM, knowledge of the domain is expressed as a set of constraints on correct solutions. ASPIRE generated syntax constraints and semantic constraints. Syntax constraints check whether the student's solution follows the syntactic rules of the domain. Syntax constraints were generated based on the concepts found in the ontologies. From the ontologies, there were 21 concepts in *yield gap analysis* task, 30 concepts in *fertilizer management* task and 26 concepts in *yield forecasting* task. Semantic constraints were generated based on the ideal solution for each problem. Semantic constraints checked whether student's solution matched the ideal solution. For DM-Tutor, ASPIRE generated 89 syntax constraints and 127 semantic constraints in total. After ensuring that all the information supplied for DM-Tutor was complete and consistent, we deployed DM-Tutor as an ITS.

## 9. Future Work and Conclusion.

At present we are researching possible ways to make an effective integration between DM-Tutor and the MIS. DM-Tutor will be accessed through the MIS. Students log in to MIS to attempt problems within DM-Tutor. For every problem in DM-Tutor, students would need to obtain relevant information from the MIS as part of their solution. We have planned to make information on the type of reports students looked at in the MIS to be send to DM-Tutor directly. This is to be done so that concise and specific feedback is provided to students when they submit an incorrect solution to a given problem. Currently I'm researching the efforts of Ritter and Koedinger [12], Cheikes, et al [2] and others to further understand and develop the most suitable architecture for embedding ITSs with existing systems.

Upon completion DM-Tutor will be evaluated by different user groups. We have planned for DM-Tutor to be evaluated by the Forestry Department, University of Canterbury. They will be looking at DM-Tutor as novice users. DM-Tutor will also be evaluated by plantation managers and trainee managers from TH Plantations, Malaysia and REAK Plantations, Indonesia. These users would be categorized as expert users for DM-Tutor. Through these evaluations we will be able to analyze how effective DM-Tutor is as a training tool. More importantly we would be able to analyze the benefits of embedding ITSs with existing systems in providing effective trainings.

## References

- [1] Anderson, J. R. and Reiser, B. J. (1985). The LISP Tutor: It approaches the effectiveness of a human tutor. *BYTE* 10, 4. 159- 175.
- [2] Cheikes, B.A., Geier, M., Hyland, R., Linton, F., Rodi, L. & Schaefer, H.P. (1998). Embedded training for complex information systems. *Artificial Intelligence in Education*, 10, 314-334.
- [3] Freedman, R. (2000). What is an intelligent tutoring system, *Intelligence*, 11(3), 15-16.
- [4] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- [5] Mathan, S.A. & Koedinger, K.R. (2002). An empirical assessment of comprehension fostering features in an intelligent tutoring system. *6th International Conference: ITS 2002.LNCS 2363*, pp. 330-343

- [6] Mitrovic, A. (2003) An intelligent SQL-Tutor on the Web. *International Journal Artificial Intelligence in Education*, vol. 13, no. 2-4, 173-197.
- [7] Mitrovic, A., Martin, B., Suraweera (2007), P. Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. IEEE Intelligent Systems, special issue on *Intelligent Educational Systems*, vol. 22, no. 4, pp. 38-45
- [8] Mitrovic, A., Martin, B. Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N. (2009) ASPIRE: An Authoring System and Deployment Environment for CBTs. *International Journal Artificial Intelligence in Education*, vol 19, no 2, 155-188
- [9] Ohlsson, S. (1994), Constraint-based modeling. In Greer, J.E., McCalla, G (eds): Student modeling: the key to individualized knowledge-based instruction, 167-189
- [10] Ravan, S. (2007). MIS for Palm Oil. RANN Consulting. <http://www.rann-consulting.com>
- [11] Risco, S. & Reye, J. (2009). Personal Access Tutor-Helping students to learn MS-Access. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*. pp. 541-548
- [12] Ritter, S. & Koedinger, K.R. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education*, 7(3-4), 315-347
- [13] Suraweera, P. & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modeling. *Artificial Intelligence in Education*, 14, 375-417.
- [14] Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R. H., Taylor, L., Treacy, D. J., Weinstein, A., and Wintersgill, M. C. (2005). The Andes physics tutoring system: Lessons Learned. *Artificial Intelligence in Education*, 15, 147-204



# INTELLIGENT TUTORING SYSTEM FOR THE PALM OIL INDUSTRY

Sagaya SABESTINAL<sup>1</sup>, Antonija MITROVIC<sup>1</sup>, Ravan SARAVANAN<sup>2</sup>

<sup>1</sup>Intelligent Computer Tutoring Group (ICTG), University of Canterbury, New Zealand

<sup>2</sup>RANN Consulting, Malaysia

[sagaya.amalathas@pg.canterbury.ac.nz](mailto:sagaya.amalathas@pg.canterbury.ac.nz)

**ABSTRACT:** This paper discusses the design and development of an Intelligent Tutoring System (ITS) for the Palm Oil industry. The ITS would provide the benefits of one-on-one workplace training to palm oil plantation managers and employees without requiring a personal human trainer. People can be trained from any location, anytime; without additional training cost to the organization. The ITS models the trainee's knowledge level and skills by assessing user's actions and based on the derived model adapts the delivery of knowledge to the user. Initially the ITS would be training users on effective decision making on yield improvement, soil nutrient management and yield forecasting, three global concerns within the palm oil industry. The developed ITS will be integrated with a Management Information System (MIS) developed for the palm oil industry. This enables users to train their decision making skills with real-life plantation scenarios and actual operational data and allows them to fully benefit from scenario based training using problem solving approach.

**Keywords:** workplace training, knowledge level, effective decision making

## 1. INTRODUCTION

One of the main factors contributing to the success and failure of oil palm plantations management is effective decision making. In the early days, plantation management dealt with relatively smaller-scale operations in less complex environments and estate managers ran plantations with sheer experience and skills. As the palm oil industry rapidly emerged as a global industry and many plantations are remotely managed, making effective decisions in plantation management becomes difficult but more crucial [11]. Adding to this challenge are the dynamic conditions and ever changing decision variables within the plantations operations.

With this concern at heart, many oil palm industries have embraced plantation management solutions where vendors provide training as part of the system implementation procedure [6]. Organizations have readily invested huge sums of money for training each year. But to what effect? The sad truth is that most training methods and technologies generally produce, at best, "trained novices." That is, they introduce facts and

concepts to trainees, present them with relatively simple questions to test this new knowledge, and provide them with a few opportunities to practice using this knowledge in exercises or scenarios [9]. However, becoming truly proficient in decision making requires extensive practice solving realistically-complex problems in a wide range of situations, combined with coaching and feedback from managers, more experienced peers, or other types of experts.

Current training technologies can help create trained novices but they are really not up to the job of creating expert decision makers. For example, multimedia computer-based training (CBT) systems are good at presenting information and then testing factual recall using multiple choice or fill-in-the blank questions. However, traditional CBT systems are not capable of providing intelligent, individualized coaching, performance assessment and feedback for students who need to acquire broad and deep expertise. Although web-based training (WBT) reduces logistical difficulties related to distributing and installing educational or training software, its pedagogical limitations are just as

pronounced. When using web based learning systems, students typically rely on asynchronous communication to get help from instructors to solve their problems. When faced with a problem, the student must wait for help from an online instructor, who, depending on the instructor-to-student ratio, could take a long time to respond.

Computer-based simulations allow students to learn and experience the effects of different actions in a variety of situations. Although training simulations were pioneered by the aviation industry and the military, they are being adopted by more and more businesses to help students learn diverse subjects such as equipment and software operations and maintenance, business and analytical skills, and interpersonal skills. Today, many of these simulations are relatively simple and prompt the student to select from just a few choices at each step in the simulation. As these simulations become more sophisticated, presenting many possible actions to the student and modeling a variety of cause-and-effect relationships, it becomes more and more difficult for the student to determine exactly what they did well or not so well that led to their success (or failure) during the scenario based training [9].

This paper presents the research and development of an Intelligent Tutoring System (ITS), DM-Tutor (Decision Making-Tutor) that is being developed for the palm oil industry. The goal of DM-Tutor is to provide training on plantation decision making for palm oil employees and managers. DM-Tutor will be embedded into an existing MIS (Management Information System) for Palm Oil. Through this integration DM-Tutor would provide scenario based training using the problem-solving approach.

## **2. INTELLIGENT TUTORING**

Intelligent tutoring systems (ITSs) have been very successfully adopted in various teaching and training domains [5]. The goal of intelligent tutoring or training systems is to provide one-on-one training automatically and cost effectively. ITSs assess each learner's actions within its interactive learning environment and develop a model of the learner's knowledge, skills and expertise.

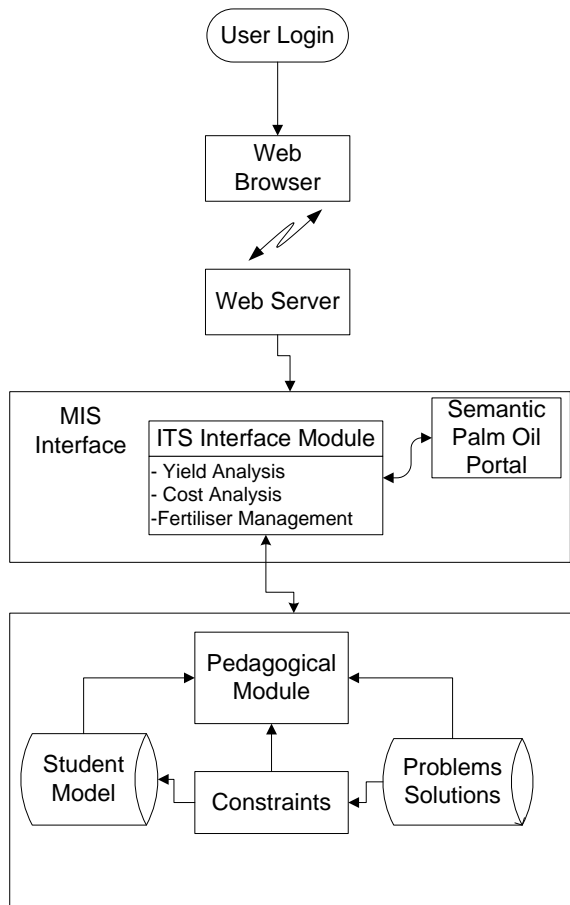
Based on the user's model, ITS tailors its instructional strategies, in terms of both content and style [5]. A variety of Artificial Intelligence (AI) techniques are used to capture how problem based learning approach is incorporated into scenario based training within ITSs [2]. Successful embedded ITSs have been implemented by the U.S. Navy [9] and Military [3] to train senior officers and personnel on effective tactical and operational decision-making.

## **2. MIS FOR PALM OIL**

The MIS for palm oil industry [10] provides an integrated software solution for managing oil palm plantation operations. The MIS modules provide for resources management, finance and accounting management and other management concerns. The aim of the MIS is to achieve management objectives of optimum profit and balanced operations costs. Managers carry out yield and cost related analyses using the operational data contained within the MIS database. As the information is highly domain-specific and requires detailed knowledge of the domain and the MIS, managers who are new to the domain or to the MIS face difficulties in making accurate and well defined analyses. This in turn affects the management and operational decisions they would make.

## **3. DM-TUTOR**

DM-Tutor is unique because it presents an ITS embedded within an actively used system, the MIS. Figure 1 presents the architecture of the MIS with DM-Tutor and its components embedded within it. The MIS [9] by itself is a web based system and is accessed via the web browser. When user logs into DM-Tutor for the first time, s/he gets a brief description of the system and the problem solving environment it provides. A log is used to record users' actions in DM-Tutor. Through being embedded within the MIS, DM-Tutor is able to present real life operational data that provides scenario and problem based training to the user.



**Figure 1:** Architecture of DM-Tutor

Constraint based modeling (CBM) [7] approach is used to model the user. The user is modeled by looking through their solution and comparing it to the ideal solution. User model is updated every time s/he uses the ITS. Pedagogical module contains instructions relevant to the scenario-based problem solving strategy used. Domain module is represented in the form of constraints [7]. A semantic portal will be used to provide additional information from WWW in the form of documents, video and images. Integration module presents the integration of DM-Tutor with the MIS. Through this integration users are able view actual MIS operational data to simulate real plantation decision making.

The Student model in DM-Tutor models the user based on the answers the user provides to the given questions. Answers given by users will be compared to the ideal solutions within DM-Tutor’s knowledge base and appropriate feedback is provided for the user. DM-Tutor would log the

user’s actions at every stage. As the questions require user to access relevant information in the MIS, DM-Tutor would need to know if the user did access the MIS and the reports accessed. This will enable DM-Tutor to provide helpful feedback when user faces difficulties in answering the questions.

### 3. DM-TUTOR USE CASE SCENARIOS

We would be discussing three use case scenarios of DM-Tutor in this paper. We have selected these scenarios based on the areas that are the main focus within palm oil plantation management.

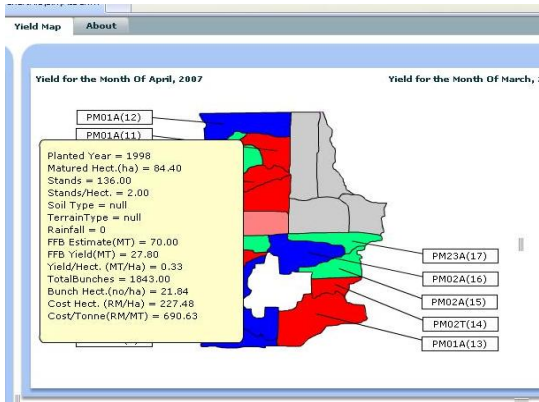
#### 3.1. Yield Gap Analysis

Yield in a particular year is always affected by climatic and plantation conditions of past production periods. Potential yield ( $Y_p$ ) can be identified by making use of past yield records from several production months or years. Potential yield is what the management hopes to ideally achieve in an estate or plantation area. The difference between actual yield ( $Y_a$ ) and potential yield ( $Y_p$ ) is an important criterion for making the decision if management actions could be taken to improve yield of the particular field or should replanting be considered. An estate achieves greatest return to the inputs of management staff when attention is focused on fields where the yield gap ( $Y_p - Y_a$ ) is greatest [4]. Below is a set of example questions DM-Tutor would provide to users and appropriate user/student actions for this scenario.

*Question: Compare the yield of the different fields in Estate Bukit Lawiang. Estimate potential yield of the estate and calculate yield gap for each field of this estate*

Figure 2 shows the screenshot of yield map in MIS from where user can obtain the yield data of different fields and compare the yield for several production periods. Yield at above 90th





**Figure 2: Estate Yield Analysis**

percentile of the estate production is considered as potential yield ( $Y_p$ ). User has to list both field number and yield in the workspace provided in DM-Tutor. DM-Tutor will require the user to estimate the potential yield of the estate and fill in the box provided for potential yield input. Next, the user will be required to calculate yield gap (using the formula) for each field and enter this information as well into the provided workspace.

Lastly, DM-Tutor requires the user to identify field with the highest yield gap and give his/her recommendation on how to improve yield in that particular field. As DM-Tutor is procedural based, it will ensure that user solves each problem step before going to the next. For example, if the user calculates yield gap inaccurately, DM-Tutor tries to identify what caused the error in user's response and guides the user to arrive to the correct solution.

### 3.2. Nutrient Management

Fertilizers are usually the largest variable cost item in oil palm production. Improvements in oil palm nutrition are required to improve site yield potential. In order to have a balanced fertilizer program, potential needs of the crop at various stages of growth must be determined. Focus is also emphasized on the most suitable nutrient combination and the appropriate placement of the fertilizer relevant to the crop/palm age [3]. For DM-Tutor we selected several analyses that would help the MIS user to identify nutrient efficiency of the plantation and fertilizer needs of the estate or field in question. Partial

Factor Productivity (FPF) formula is used to identify how much yield is produced for each kg fertilizer used and Agronomic Efficiency (AE) is used to calculate how much additional yield is produced for each kg of fertilizer nutrient applied for an estate [4]. For this scenario the questions would be as below:

*Question: Given below the Upkeep and Cultivation analysis of Estate Bukit Lawiang. (Figure 3). Calculate PFP and AE for this estate. For Estate Bukit Lawiang, identify the age of palm trees and give your recommendation for a suitable fertilizer combination and placement.*

Mature Planted Area	: 219.00
Total Stand	: 14652
Stand/Hect	: 67
Expenditure Code	: PM0303 - MANURING - MATERIAL
Material Cost (RM)	: 24,928.00

Material Desc	Unit	Qty	Qty/Stand	Cost
MIX THP2	50KG/Bag	760.00	0.05	24,928.00
				24,928.00

**Figure 3: Estate fertilizer consumption**

DM-Tutor would ask the user to select correct formulas to calculate both PFP and AE [4]. The user would have to use the formulas to do the calculations and identify the nutrient efficiency of the estate. Next, DM-Tutor would require the user to identify palm age of the particular estate and select suitable fertilizer combination that best meets the palm tree's needs. Users would then be presented with a potential fertilizer management problem and will be expected to provide a suitable solution to the problem.

### 3.2. Yield Forecasting

Numerous attempts have been made to devise a reliable yield prediction method for palm oil plantations. From counting fruit bunches to using econometric regression, various methods have been used to forecast yield for both short and long term [12]. The regression methods could provide forecast with a higher degree of accuracy but they require development of a

multiple linear models which are time consuming and need expert statistical analytical skills to develop them. For DM-Tutor we will be using a practical approach for managers to do yield forecasting with data available within the MIS. Figure 4 shows a screenshot of a yield analysis report in the MIS. Using the current yield of a given estate or field we forecast yield for a future period within the MIS. Yield forecast = current yield (MT/Ha)\*future plantation size (Ha) [1]. For this scenario the questions would be as follows.

*Question: Based on yield for April 2006 for Estate Bukit Lawiang, forecast yield for April 2007. Analyse the effect of weather conditions on yield that caused the difference between forecast and actual yield for Estate Bukit Lawiang by selecting the relevant reports you would need to access and the information you would need from each report.*

DM-Tutor would ask user to select yield forecast formula from the list of available formulas. DM-Tutor will require the user to access the relevant estate yield analysis information for the particular estate for that period of time and calculate yield forecast for the given period. Next the user would compare the actual yield of the forecasted period with his/her forecasted value. DM-Tutor would provide a problem scenario where a plantation condition contributed to the difference between

LADANG BUKIT LAWIANG				
	THIS MONTH	THIS MONTH EST.	VARIANCE THIS MONTH	TO DATE
Total Hectare (M,IM,R,NY)	1,989.00	1,989.00	0.00	1,989.00
Mature Hectare	1,989.00	1,989.00	0.00	1,989.00
Stands	260,095.00	0.00	0.00	260,095.00
Stand/Hectare	131	0	0	131
Rainfall(mm)	0.00	0.00	0.00	0.00
FFB(MT)	4,034.40	3,038.00	996.40	11,111.11
Oil(CPO)(MT)	815.56	580.97	234.59	2,222.22
Kernel(MT)	254.16	151.90	102.26	714.29
Palm Product(MT)	1,069.72	731.00	338.72	2,941.18

**Figure 4:** Estate yield analysis

forecasted yield and actual yield. Next DM-Tutor would test the user's knowledge of the plantation and management considerations that contribute to yield, for example rain, fertilizer, vehicle management and workers' factors [12]. Next DM-Tutor would require users to provide their recommendation to improve yield for the particular given scenario. As DM-Tutor updates user's model based on his/her progress it is able to

provide contextualized instruction on problem solving. A semantic portal is also integrated to DM-Tutor to provide additional declarative knowledge to the user. For example if through DM-Tutor it is evident that the user is not clear on the factors contributing to yield forecasting in palm oil plantations, DM-Tutor identifies this information and the next time user logs into DM-Tutor would make this additional information available through articles, journals or videos that are relevant to the area user needs to improve on.

#### 4. CONCLUSION

DM-Tutor is aimed to provide employees in palm oil plantations the required theoretical knowledge through the analyses within DM-Tutor and the practical skills required to make effective decisions in managing palm oil plantations.

DM-Tutor is a work-in-progress and upon completion would have a very practical benefit of providing life-long workplace learning anywhere and anytime. We have planned for DM-Tutor to be evaluated by users of different levels: managers in palm oil industries, trainee managers and novices. Upon completion of DM-Tutor we hope to be able to present users'

evaluation of our system and our belief of the potential of ITS in increasing workplace efficiency.

#### 4.4. Acknowledgements

The first author thanks RANN Consulting, Malaysia for funding this research.

#### 5. REFERENCES

- [1] Ballard, S & Miramontez, D. (2009). Forecasting FTES using a yield projection model. RP-CISOA Conference, U.S.A
- [2] Beck, J., Stern, M. & Haugsjaa, E. (1996). Applications of AI in education. *Crossroads*, 3 (1), ACM Student Magazine. 1528-4972. pp. 11-15
- [3] Cheikes, B.A., Geier, M., Hyland, R., Linton, F., Rodi, L. & Schaefer, H.P. (1998). Embedded training for complex

- information systems. *Artificial Intelligence in Education*, 10, 314-334.
- [4] Fairhurst, Thomas (2003). Efficient management of natural resources in oil palm plantations. Proceedings of the International Planters Conference, June 2003. Pp 135-174.
- [5] Freedman, R. (2000). What is an intelligent tutoring system. *Intelligence*, 11(3), 15-16.
- [6] Kuruvilla, K.J., Rankine, I., Lee, C.T. & Mohd. Hashim, T.(2003). Innovation and the plantation industry. Proceedings of the International Planters Conference, June 2003. Pp 215-218.
- [7] Mitrovic, A., Martin, B., & Suraweera, P. (2007). Constraint based tutors: past, present and future. *IEEE Intelligent Systems, special issue on Intelligent Educational Systems*, vol.22, no.4, pp 38-45
- [8] Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *Artificial Intelligence in Education* 10, 98-189.
- [9] Ong, J. & Ramachandran, S. (2003). Intelligent tutoring systems: Using AI to improve training performance and ROI. Stottler Henke Associates. [http://www.stottlerhenke.com/papers/ITS\\_Using\\_AI\\_to\\_improve\\_training\\_performance\\_and\\_ROI.pdf](http://www.stottlerhenke.com/papers/ITS_Using_AI_to_improve_training_performance_and_ROI.pdf)
- [10] Ravan, S. (2007). MIS for Palm Oil. RANN Consulting. <http://www.rann-consulting.com>
- [11] Tan, K.T., (2003). Challenges Facing Plantation Management. Proceedings of the International Planters Conference, June 2003. Pp 45-54.
- [12] Weng, C.K., (1999). Towards Higher Yield Potential, Production and its Prediction in Oil Palm. PORIM Bulletin Number 39. MPOB Publications.

# Evaluation of DM-Tutor, an ITS for Training on Plantation Decision Making

Sagaya AMALATHAS<sup>1</sup>, Antonija MITROVIC<sup>1</sup>, Ravan SARAVANAN<sup>2</sup> & David EVISON<sup>3</sup>

<sup>1</sup>*Department of Computer Science, University of Canterbury, New Zealand*

<sup>2</sup>*RANN Consulting, Malaysia*

<sup>3</sup>*Department of Forestry, University of Canterbury, New Zealand*

sagaya.amalathas@pg.canterbury.ac.nz

**Abstract:** Over the years many Intelligent Tutoring Systems (ITSs) have been used successfully as teaching and training tools with proven results through evaluation studies. Even though ITSs are effective in providing individualized tutoring to many students at the same time, very few attempts have been made to embed them with existing systems. This area of research has a lot of potential in providing life-long learning and work place training anytime and anywhere. We present the evaluation of DM-Tutor, an ITS that provides training on plantation decision making. DM-Tutor has been embedded with a Management Information System (MIS) to provide scenario-based training using real life operational data and actual plantation conditions. The pilot study conducted in February 2011 shows that DM-Tutor is an effective teaching tool; the performance of the participants on the post-test was significantly higher than their performance on the pre-test. The questionnaire responses show that the participants found the system easy to use and useful.

**Keywords:** Intelligent tutoring systems, embedded ITSs, management information system, scenario-based training.

## Introduction

Rapid technological advances have created profound implications for all levels of education and training around the world. Intelligent Tutoring Systems (ITSs) have become an important class of educational technology that is capable of playing a crucial role helping learners of any age acquire the skills needed to succeed. ITSs have been developed and effectively used for teaching and learning for many years and have been proven in the past to be successful in providing tutoring to any number of students

through highly interactive environments [8]. The ultimate goal of ITSs is to achieve the learning gain of 2 standard deviation, the same level achieved by master human tutors in one-to-one interactions with students [5]. Over the years many ITSs including LISP tutor [3], Andes [22], PUMP Algebra Tutor [11] and others have been effectively used in many teaching and learning domains. SQL-Tutor [13], NORMIT [14], KERMIT [21], UML Tutor [4] and J-Latte [9] are among the many constraint-based tutors (CBT) [15] that have been developed and successfully implemented by the Intelligent Computer Tutoring Group.

Even though ITSs have been proven as effective teaching tools, there have been very few attempts to embed them within other systems. ETS (Embedded Training System) [6], Macsyma Advisor [10], Geometer Sketchpad Tutor [20], Excel Algebraic Tutor [12] and PAT (Personal Access Tutor) [19] are some of the attempts made to embed ITSs into existing systems. Macsyma Advisor was developed to assist users in using Macsyma, the algebraic manipulation system. Macsyma Advisor focused on assisting rather than training, which meant the user, may not have learned how to solve problems unassisted in the future. Although various parts of Macsyma Advisor were implemented, they were not combined into a full working system; and evaluation studies were not carried out to assess the effectiveness of Macsyma. Excel Tutor provided descriptive explanations and interactive guidance for students to solve excel problems and Geometer Sketchpad Tutor taught students how to sketch geometric diagrams. For Geometer Sketchpad Tutor and Excel Algebraic Tutor, users' actions were observed and monitored through the system's interface. Initial evaluations showed that Excel Tutor provided a higher learning outcome for students. However, to our knowledge no detailed empirical evaluation was done to analyze students' interactions with the system. The tutors were not assessed for their effectiveness as educational systems. ETS was developed and integrated with an existing Complex Information System for military tasks and operations. The goal of ETS was to train users on military based scenarios. ETS however was not robust enough to handle a variety of student's actions and behaviors within the system. Evaluation studies were not conducted for the system and the learning outcome was also not evaluated. PAT was added into MS Access to help students build reports and forms. Although the students seemed to like PAT, the system has not been fully evaluated.

In this paper, we present DM-Tutor (Decision-Making Tutor) [1], an ITS that provides training on plantation decision making for the palm oil domain. We aim to make several significant contributions in our research. This will be the first attempt to embed a CBT with an existing system. Secondly, we will investigate the benefits of providing on-the-job training through this integration. Thirdly, we aim to develop a framework for embedded ITSs and prove its research contribution through the development of DM-Tutor and its integration with the MIS for palm oil.

DM-Tutor has been embedded with a Management Information System (MIS) [18] that is currently being used to manage palm oil plantations in Malaysia and Indonesia. The MIS for palm oil contains operational data of yield records and plantation cultivation details. As the information contained is highly domain specific, managers who are new to the domain or to the MIS face difficulties in making accurate operational analyses and this affects the decisions they make at the palm oil plantations. As DM-Tutor is

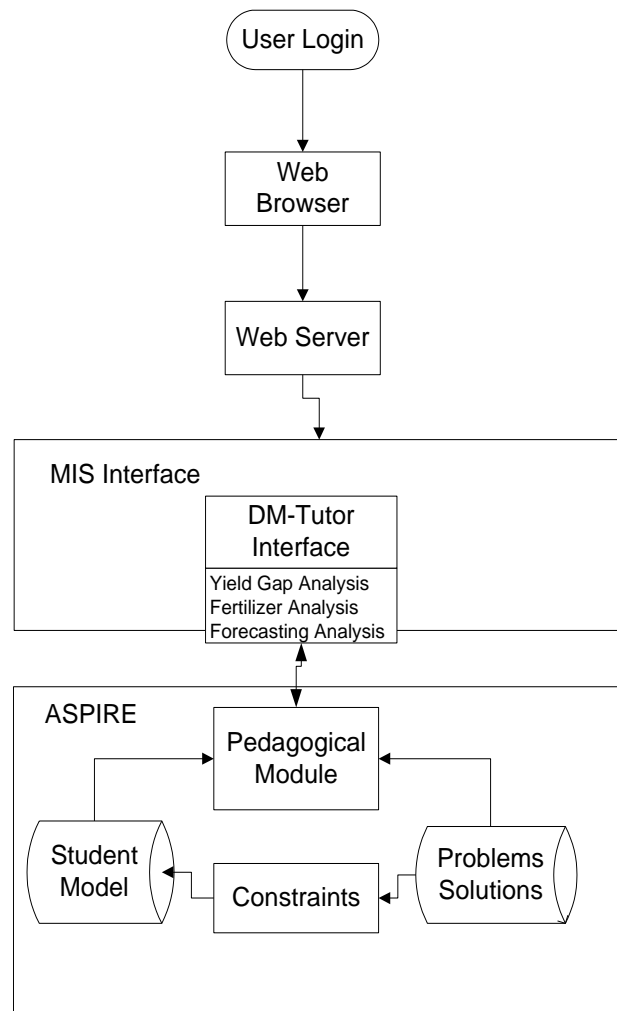
embedded within the MIS for palm oil, users will be able to practice plantation decision making using real-life operational data from the MIS. The goal of DM-Tutor is to help users apply theoretical concepts of plantation analyses into real-life plantation decision making.

This paper is organized as follows. In section 1 we describe the architecture and components of DM-Tutor. Section 2 describes the evaluation study conducted for DM-Tutor and in section 3 we present conclusions and future work.

## 1. DM-Tutor

To the best of our knowledge, there has not been an ITS for plantation decision making, and DM-Tutor is novel in that respect. It is also the first CBT to be embedded with an existing system. DM-Tutor was developed using ASPIRE [16], an authoring system and deployment environment for constraint-based tutors. DM-Tutor is a procedural based tutor. This means that problem solving is divided into several steps and student's solution is evaluated at every step. Students will not be able to continue to the next problem solving step before submitting the correct solution for the present step. Figure 1 presents the overall architecture of DM-Tutor. DM-Tutor consists of a student model, pedagogical module, interface module, constraints that represent domain knowledge and a database of problems and solutions [2].

Student model contains information of student's knowledge and is updated every time the student uses DM-Tutor. Constraint based modeling [17] is used to model the domain and the student. The student's solution is matched to the constraints in order to identify any mistakes, and the student model is updated. The pedagogical module selects instructional actions relevant to the scenario-based problem solving strategy used. It also has the role of providing helpful feedback to users when they submit an incorrect solution.



The interface module presents the student interface of DM-Tutor. The problems and solutions component focuses on *Yield Gap Analysis, Fertilizer Management and Yield Forecasting*, three main analyses for palm oil plantation management [1]. DM-Tutor's interface is designed to reduce the cognitive load so that the student could concentrate on how to solve the problems rather than trying to understand the interface. The interface is developed using Java applets. In order to solve problems presented in DM-Tutor, users need to access relevant reports from the MIS. DM-Tutor logs information on the MIS reports that users accessed. This enables the system to provide helpful feedback about student mistakes.

The MIS is a web-based system and is accessed via a web browser. When users log into the MIS, they would see a menu on the left side of the screen. The menu consists of the various information that MIS contains. This includes maps, graphs and various reports, including yield analysis, upkeep and cultivation analysis, store analysis and vehicle management analysis. An MIS user would access the type of information she/he needs by selecting the type of analysis required, estate name, year and month to view the particular report she/ he needs to work with. Figure 2 illustrates the MIS interface, with the menu of information on the left pane, and a sample of analysis showing *Yield Analysis* for a particular estate on the right. When user selects to view a particular report, a window opens up showing the report (center window of right pane).

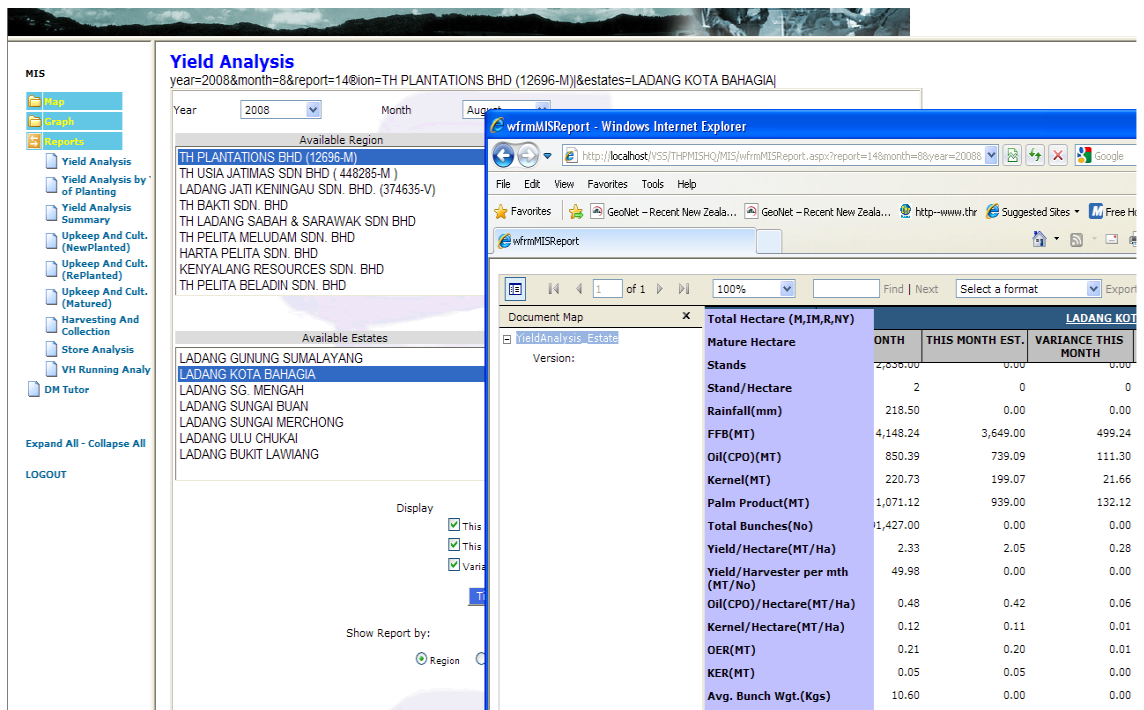


Figure 2: MIS Interface showing a yield analysis report

MIS was modified to allow the integration with DM-Tutor. The menu pane of MIS was changed by adding a selection button (visible in Figure 2) for DM-Tutor. When the

student clicks on this button, he/she will be logged into DM-Tutor automatically. The tutor is served by ASPIRE, which assigns a user ID to the user. The interface of DM-Tutor is displayed in a separate window. Once logged in, DM-Tutor users are presented with the three tasks within DM-Tutor: *Yield gap analysis*, *Fertilizer management analysis* and *Forecasting analysis*. The tasks posed by DM-Tutor contain real life plantation situations and problems. Each task in DM-Tutor focuses on one area of the plantation management's concern and requires students to access various plantation areas and specific reports from the MIS in order to answer the questions in DM-Tutor. Students also need to use the correct formulas for each analysis.

For *Yield gap analysis* [23], students first have to identify relevant plantation areas and their potential yield in the MIS. Next they need to calculate yield gap between plantation areas using the correct formula. Lastly, they need to give their recommendation on how to improve yield of that particular area. The focus of this task is to train students on how to focus management's efforts on improving conditions in plantation areas that needs management attention the most.

For *Fertilizer management analysis* [7], students need to calculate the *partial factor productivity* (PFP) and *agronomic efficiency* (AE) for a given plantation area in the MIS using a formula. They need to select the accurate nutrition combination required for the palm trees based on the age of the tree (identified from the MIS). Lastly, they need to solve a fertilizer management problem to improve yield for the given plantation location. This task teaches users on how to efficiently manage plantation nutrition.

For *Forecasting analysis* [23], students are required to calculate a future yield value based on the present yield value of a given plantation area in the MIS. They will then need to provide their solution to a given plantation scenario problem where yield losses have occurred due to various plantation conditions. This task teaches students to plan the plantation area's future yield based on current yield conditions.

When students select a task, they are presented with the window that shows the problem and solution workspace. DM-Tutor evaluates the student's solution step-by-step, immediately after the student submits his/her solution for the current step. If the submitted solution is incorrect, DM-Tutor provides various levels of feedback that guides the student towards the correct solution. Only when the solution is correct, the student is allowed to move to the next step of the problem. Figure 3 shows a screen shot of a problem step in the Forecasting Analysis task in DM-Tutor.



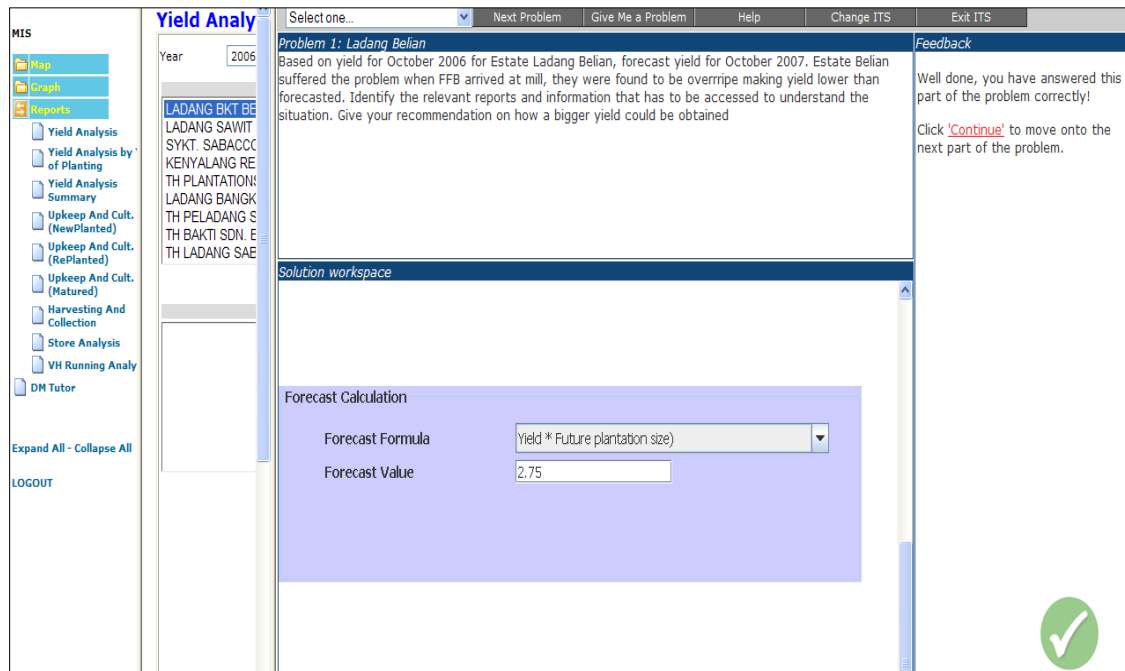


Figure 3: MIS interface with DM-Tutor's problem-solution workspace

In the example shown in Figure 3, the student has selected the correct formula to calculate yield forecast and has calculated the forecast accurately. As a response, DM-Tutor has provided a feedback message for the student to continue to the next problem step in Forecasting analysis task. The MIS and DM-Tutor frame can both be resized by the student at any time to allow more visibility for either system. All of the user's actions within DM-Tutor are recorded and used to model the student's knowledge of the domain. Once the student has completed a problem correctly she/he can choose to practice another problem or a different analysis.

## 2. Evaluation Study

Evaluation of ITSs should focus on the educational impact on students. We conducted a pilot study of DM-Tutor in February 2011 with the goal of evaluating its effectiveness as a teaching and training tool for the palm oil plantation decision making domains. We also wanted to identify any technical issues, interface problems and system usability concerns that needed to be solved. The study was conducted with a group of 22 employees working for a palm oil plantation company in Kuala Lumpur, Malaysia. All the participants were already familiar with the MIS for palm oil as they use a similar version of the system for their everyday work, so their knowledge of the MIS system was not evaluated.

We provided the participants with a demonstration on DM-Tutor, describing the different tasks contained inside DM-Tutor, problem selection and the various levels of feedback messages available. They were informed during this demonstration that their

participation in the study was on a voluntary basis. The participants were given a pre-test before interacting with the system. The pre-test contains three questions, one for each task of DM-Tutor. Each pre-test question is a shorter and simpler version of the actual problems in DM-Tutor. Participants spent 15 minutes doing the pre-test, after which they interacted with the system for one hour. Lastly, they worked on the post-test for another 15 minutes and also completed the questionnaire. The post-test also consisted of three questions, of similar complexity to those in the pre-test. Out of the 22 participants who initially volunteered, only 19 stayed through the whole study. Table 1 presents the basic statistics about the participant's interaction with DM-Tutor.

Table 1: DM-Tutor Interaction Results

User log data from DM-Tutor	Mean (sd)
Interaction time (min)	27.20 (24.43)
Number of problems attempted	2.85 (1.23)
Number of problems solved	2.05 (1.05)
Number of submissions made	38.05 (8.96)
Number of feedback messages seen	19.80 (8.25)
Pre-test result (%)	50 (16)
Post-test result (%)	78 (21)
Gain (%)	28 (0.15)

Only three participants interacted with DM-Tutor for one hour or more. On average, the participants interacted with the system for about 27 minutes. The possible reason for this could be that the study was held during office hours and the participants had to complete their pre-assigned daily work load as well. The number of problems attempted by participants is higher than the number of problems they managed to solve. To solve the given problems, the participants made around 38 submissions of answers to DM-Tutor (please note that a submission is a partial answer, which covers only the current step of the task). From Table 1, we can observe that the participants have utilized the feedback messages in DM-Tutor to help them to solve the given problems.

Out of the 19 participants, only 8 completed both the pre-test and post-test for this study (the others completed only one test). Therefore, we only report the pre/post test results for those 8 participants in Table 1. We found that the performance on the post-test is significantly higher than the performance on the pre-test ( $t= 1.89$ ,  $p= 0.005$ ,  $df=7$ ). From the pre-test and post- test results we could observe that the participants' knowledge in palm oil plantation decision making increased after using the system; the amount of increase is strongly correlated with the time the participants spent with the system ( $r=0.92$ ).

From the 19 questionnaire responses obtained, we found out that 74% of the participants thought that DM-Tutor was easy to use. 52% of the participants found the feedback messages in DM-Tutor to be helpful in answering the questions. When they were asked if they liked DM-Tutor's interface, 63% of the participants said that they did. One of the participants commented that he liked this new version of the MIS. When

asked if DM-Tutor was able to teach them any new palm oil plantation decision making analyses, all the participants answered 'yes'. One participant said that DM-Tutor helped to teach her how to use the information in the MIS to make better decisions. Another participant said that she knew now what yield gap analysis and fertilizer efficiency were, after using DM-Tutor. When the participants were asked if they felt that by integrating the Management Information System (MIS) with DM-Tutor they were able to learn the plantation analyses better? All of them answered 'yes'. One participant said that he liked to see the MIS and the teaching system together. Another participant stated that she liked the idea that she could stop learning and continue to work with MIS when she needed to and another participant was happy that she could choose the analyses that she needed to learn and that she could check the analysis again if she was not sure.

### **3. Conclusions and Future Work**

ITSs have proven to be successful in many instructional domains. ITSs are better than computer-based training or multimedia-based training methods that only present information and test recall of factual information through multiple choice questions. ITSs are effective in providing individualized one-to-one teaching or learning to many people at the same time. By embedding DM-Tutor, a constraint-based tutor, into a MIS for palm oil we hoped to provide scenario-based training (SBT) on palm oil plantation decision making using real life operational data and actual plantation conditions. Our goal was to make several significant contributions through this research. DM-Tutor is the first CBT embedded within an existing live system. Through this research we aim to develop a framework for embedded ITSs. We aimed to investigate the benefits of providing training through this integration. We hoped to prove our research contribution through the development of DM-Tutor and its integration with the MIS for palm oil.

From the pilot study, we found that the users have interacted well with DM-Tutor. They have used the information from the MIS to answer the questions posed by DM-Tutor. The participants have also utilized the feedback messages in DM-Tutor to help them answer the given questions. From the pre- and post-test results it is proven that the users have improved their knowledge of palm oil plantation decision making after using DM-Tutor. It was also very encouraging to get positive comments on DM-Tutor from the participants of the study. However, our study was small in terms of the number of participants, and therefore more evaluation is needed.

Currently, we are making further enhancements to DM-Tutor. We have modified the MIS to automatically send information about which reports the user has looked at in MIS directly to DM-Tutor, thus making interaction easier for users (as they do not have to type report details). We plan to conduct a full evaluation study of DM-Tutor in 2011 with a group of students currently enrolled in the Master's in Plantation Management program at the Putra University of Malaysia. We believe that our research has many potential benefits. Apart from the research contributions discussed above, DM-Tutor also has the potential benefit of providing life-long workplace learning, anywhere and anytime.

## Acknowledgements

The first author thanks RANN Consulting, Malaysia for funding this research. We also thank the people who participated in the evaluation study.

## References

- [1] Amalathas, S., Mitrovic, A. & Ravan, S. (2009) Towards an ITS for decision making on managing palm oil Plantations. Workshop Proceedings ICCE 2009, Hong Kong. APSCE . pp 157-162.
- [2] Amalathas, S., Mitrovic, A. & Ravan, S. (2010) Towards a framework for embedded ITSs. DSC Proceedings ICCE 2010, Malaysia, APSCE. pp 1-5.
- [3] Anderson, J. R. & Reiser, B. J. (1985). The LISP Tutor: It approaches the effectiveness of a human tutor. *Byte* 10(4), 159- 175.
- [4] Baghei, N., Mitrovic, A. & Irwin, W. (2007). Supporting Collaborative Learning and Problem Solving in a Constraint-based CSCL Environment for UML Class Diagrams. *Computer-Supported Collaborative Learning*, 2(2-3), 159-190
- [5] Bloom, B.S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6), 4-16.
- [6] Cheikes, B.A., Geier, M., Hyland, R., Linton, F., Rodi, L. & Schaefer, H.P. (1998). Embedded training for complex information systems. *Artificial Intelligence in Education*, 10, 314-334.
- [7] Fairhurst, T. (2003). Efficient management of natural resources in oil palm plantations. Proceedings of the International Planters Conference, pp. 135-174
- [8] Freedman, R. (2000). What is an intelligent tutoring system, *Intelligence*, 11(3), 15-16.
- [9] Holland, J., Mitrovic, A. & Martin, B. (2009). J-LATTE: a constraint based tutor for Java. Proceedings of ICCE 2009, Hong Kong. APSCE. pp 27-34
- [10] Genesereth, M.R. (1979). The role of plans in automated consultation. Proceedings of the 6th International Joint Conference on Artificial Intelligence, pp. 311-319.
- [11] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *Artificial Intelligence in Education*, 8, 30-43.
- [12] Mathan, S.A. & Koedinger, K.R. (2002). An empirical assessment of comprehension fostering features in an intelligent tutoring system. Proceedings of the 6th International Conference ITS 2002, LNCS 2363, pp. 330-343
- [13] Mitrovic, A. (2003). An intelligent SQL-Tutor on the Web. *Artificial Intelligence in Education*, 13(2-4), 173-197.
- [14] Mitrovic, A. (2005). The effect of explaining on learning: a case study with a data normalization tutor, Proceedings of AIED 2005. Amsterdam, IOS Press 2005, pp. 499-506
- [15] Mitrovic, A., Martin, B. & Suraweera, P. (2007), Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring. *IEEE Intelligent Systems* 22(4), 38-45
- [16] Mitrovic, A., Martin, B. Suraweera, P., Zakharov, K., Milik, N., Holland, J. & McGuigan, N. (2009) ASPIRE: An Authoring System and Deployment Environment for CBTs. *Artificial Intelligence in Education*, 19(2), 155-188
- [17] Ohlsson, S. (1994), Constraint-based modeling. In Greer, J.E., McCalla, G (eds): Student modeling: the key to individualized knowledge-based instruction, 167-189
- [18] Ravan, S. (2007). MIS for Palm Oil. RANN Consulting. <http://www.rann-consulting.com>
- [19] Risco, S. & Reye, J. (2009). Personal Access Tutor-Helping students to learn MS-Access. Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009). pp. 541-548
- [20] Ritter, S. & Koedinger, K.R. (1996). An architecture for plug-in tutor agents. *Artificial Intelligence in Education*, 7(3-4), 315-347
- [21] Suraweera, P. & Mitrovic, A. (2004). An intelligent tutoring system for entity relationship modeling. *Artificial Intelligence in Education*, 14, 375-417.

- [22] Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R. H., Taylor, L., Treacy, D. J., Weinstein, A. & Wintersgill, M. C. (2005). The Andes physics tutoring system: Lessons Learned. *Artificial Intelligence in Education*, 15, 147-204
- [23] Weng, C.K., (1999). Towards Higher Yield Potential, Production and its Prediction in Oil Palm. PORIM Bulletin Number 39. MPOB Publications, 25-37