A Learning Support Web-based System using Robert Gagné Theory and a Rules-Based Approach

Jeerauch Khawbang, Sarefa Thoyom, Duenpen Kochakornjarupong
Department of Computer and Information Technology
Faculty of Science, Thanksin University,
Phatthalung, Thailand

***Abstract*—**This paper presents the design, development, and evaluation of a web-based system for supporting propositional logic learning using Robert Gagné theory and a rules-based approach. The system design uses UML, its development utilizes Open Source software under the windows OS, and the evaluation of the system with 30 users was assigned a high satisfaction level (4.35).

Keywords- tutoring program, propositional logic, web-based system, learning support, rules-based approach

I. INTRODUCTION

A web-based tutoring program for learning propositional logic can help students understand and solve problems more easily. They can practice iteratively at any time and at any place with internet connectivity.

Robert Gagné [1]'s nine concepts of instructional theory is a popular learning support mechanism in Mathematics [2], and our system adopted Gagnè's nine events for effective learning: 'Gain attention', 'Specify Objective', 'Activate prior knowledge', 'Present the content', 'Guide learning', 'Elicit performance', 'Provide feedback', 'Assess performance', and 'Review and transfer'. In addition, an Intelligent Tutoring System (ITS) model based on a rules-based approach [3] was applied to the design.

II. LEARNING SUPPORT WEB APPLICATIONS RELATED TO PROPOSITIONAL LOGIC

There are several learning support web applications for Logic or Propositional Logic [4-7]. For example, Wongchosiri's [4] application summarizes each topic and displays overall exercises not limited to a chapter or topic. Missing knowledge related to a question is explained after the end of the exercise. With regards to Gagnè [1]'s nine concepts, the application adopts 'Gain attention', 'Present the content', 'Provide feedback', and 'Assess performance'. Another example is Nokkeaw's [5] application which divides its content into the sections: which consists of "propositional logic", "conncectives of propositional logic", "equivalence of propositional logic", "tautology", "open sentence", "quantifier", and "argument". It utilizes three of Gagnè's ideas: 'Gain attention', 'Present the content', and 'Assess performance'. Kannin [7]'s work applies 'Gain attention', 'Present the content', and 'Review and transfer' because, although the content is read only, pre-test and post-test are also provided. Kruheem[6]'s application has a human tutor presents via YouTube videos and so utilizes 'Activate prior knowledge', Present the content', 'Assess performance', and 'Review and transfer'.

These learning support web applications for related to Propositional Logic are summaized in Table 1 which highlights how Gagné's nine ideas are suited for designing instructional web-based learning systems. The concepts highlight the interaction between lessons learners (e.g. through stimulation and encouragement). Although [4-6] contain 'Assess performance', they display test items in the same sequence, not randomly chosen items from a test bank.

TABLE I. A COMPARISON OF LEARNING SUPPORT WEB APPLICATIONS FOR PROPOSITIONAL LOGIC USING SOME FEATURES CONSISTENT WITH GAGNÉ'S THEORY

WEB-APPLICATION / GAGNÉ'STHEORY	1	2	3	4
GAIN ATTENTION	✓	✓	×	✓
SPECIFY OBJECTIVE	×	×	×	×
ACTIVATE PRIOR KNOWLEDGE	×	×	✓	×
PRESENT THE CONTENT	✓	✓	✓	✓
GUIDE LEARNING	×	×	×	×
ELICIT PERFORMANCE	×	×	×	×
PROVIDE FEEDBACK	✓	×	×	×
ASSESS PERFORMANCE	✓	✓	✓	×
REVIEW AND TRANSFER	×	×	✓	✓

✓ = features which consistent with Gagné's theory
× = features which not consistent with Gagné's theory
1 = Wongchosiri's [4] application
2 = Nokkeaw's [5] application
3 = Kruheem's [6] application
4 = Kannin's [7] application

None of [4-7] provide features or modules where the learner can construct their own questions in order to test his/her understanding. For this reason, we include randomly selected tests exercises items from the relevant banks and have the learner build questions to evaluate their understanding.

III. DESIGN OF A PROPOSITIONAL LOGIC LEARNING SUPPORT WEB-BASED SYSTEM

The system design use UML consisting of:

1. A use case diagram (see Figure 1) which displays the relationship between a user and all the system processes.

2. Use case specifications to describe the working system processes,

3. Activity diagrams to present system situations,

4. Sequence diagrams to show the event ordering in the system.

According to the use case diagram in Figure 1, the tutor can be a teacher, an instructure or a lecturer and the students are actors of the system who must first login. The manages the system data such as the learning contents, exercises, tests, and student's data, which are inserted, deleted and updated in the exercises and tests banks (see Figure 3). The pre-test and the post-test are generated randomly from the tests bank. In addition, the tutor can view the learner's progress, search/delete data and answer to the student questions. The student member can edit his/her profile, initially carry out a pre-test, then study all the lessons and exercises. Thereafter, he/she can select practice questions which will be answered automatically by the system. (see Figure 4) The learner can send questions or opinions to the tutor, who can anwer them later (see Figure 5). The system design structure is shown in Figure 8, and is suitable for primary school, high school, and undergraduate students, or any one who wants to learn and practice propositional logic.

The relationships between the entities can be expressed as the ER diagram in Figure 2. The normalization of the system database is presented below.

teacher = (ID_Teacher, username , password , T_Name, E-mail, T_surname)
test = (id, type, chapter, question, choice1, chioce2, choice3, choice3, answer)
member = (M_id, M_Name, M_Surname, M_Email, M_class, M_Study, M_Provinc, M_Username, M_password)
Upload = (ID, Filename, name_chapter)
Score = (ID_Score, S_Id, pretest, posttest, day, B_assessment, A_assessment, Exercises1, Exercises2, Exercises3, Exercises4)
tlb_question = (ques_id, title, detail , name, date_ques , view)
tbl_answer = (ans_id, detail, name, date_ans, ref_ques_id)

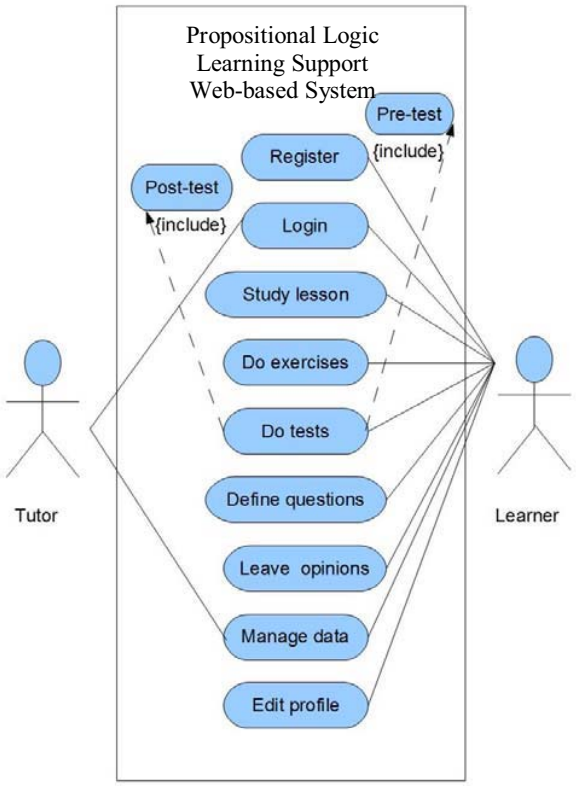


Figure 1. Use-case diagram of our Propositional Logic Learning Support Web-based System

IV. Applying Gagné's Theory and a Rules-Based Approach to a Propositional Logic Learning Support Web-Based System

Robert Gagné [1]'s nine concepts of instructional design for effective learning are as follows:

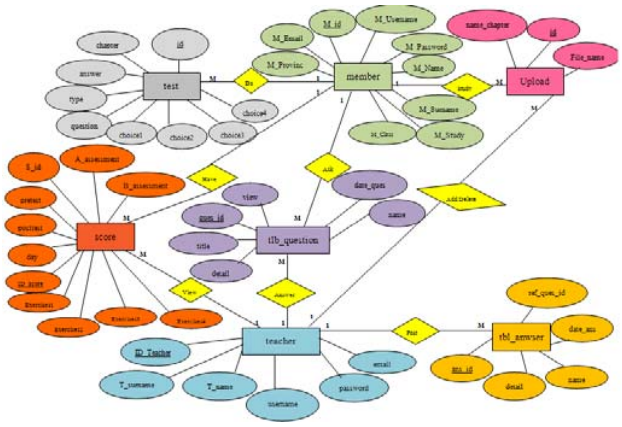


Figure 2. ER diagram of our Propositional Logic Learning Support Web-based System

A. Gain attention

A student's attention towards a lesson can be focussed inseveral ways in a new situation. It is important to motivate the student so they will learn better.

B. Specify objective

A student's thoughts can be organized by specifying the objectives of the lesson. A description of the goals of each lesson should be displayed so the learner can better understand the knowledge that is presented.

C. Activate prior knowledge

A learner's previous skills or knowledge can be activated by performing a pre-test for the lessons before starting those lessons.

D. Present the content

Bloom [8]'s taxonomy and learning strategies for sequencing content can help order a lesson, so the student can comprehend how to solve problems in propositional logic. We utilize four system topics: 'Introduction to Truth-value in Propositional logic', 'Connectives of Propositional logic', 'Equivalences of Propositional logic', and 'Negation'.

E. Guide learning

A student shold be given guidance on how to learn, while avoiding mixing the guidance with the subject matter. We do this by providing examples for each topic.

F. Elicit performance

Elicit performance involves having the learner practice their newly acquired behavior, skills, or knowledge through exercises.

G. Provide Feedback

We provide feedback for the learner by explaining the correct of the student's responses[9].

H. Assess performance

A post-test helps the learner assess their performance after studying all the topics and performing all the exercises.

I. Review and transfer

A report containing the learner's results is produced before the students is transfered to the next part of the system. The report includes the results of the pre-test, the post-test, and the exercises for each lesson (see Figures 6-7) .

Our rules-based approach is based on an Intelligent Tutoring System (ITS) design for enhancing AI problem solving [3]. The architecture (see Figure 8) consists of five components:

A. Learner model

The model contains knowledge about the individual learner, including their weaknesses and given feedback.

B. Expert model

The model contains rules for solving the student's questions.

C. Domain model

The domain model contains contents, exercises, and tests banks.

D. Pedagogical model

The model contains rules for providing feedback which based on Draper's feedback type [9], rules for Gagné's nine concepts which are used for controlling the process of presenting a topic to the learner, and rules for randomness, marking, and solving tests and exercises.

E. Knowledge management model

The knowledge management model contains modules for the tutors and students. The tutor module contains manages content, tests, exercises, the learner's profile, and opinions. In student module contains questions manages their profile.

V. IMPLEMENTATION

The system was developed with PHP under the windows OS, while the tutorial presentation was created using the Open Source software OpenOffice.org Impress. The tutorial contains sections on 'Introduction to Truth-value of Propositional logic', 'Connectives of Propositional logic', 'Equivalences of Propositional logic',

and 'Negation'. At the end of each topic, five questions are randomizing selected from the exercises bank to display on the webpage. Likewise, 20 questions are selected from the tests bank for each pre-test and post-test. There are also practice questions which will be answered automatically by the system (see Figure 4).

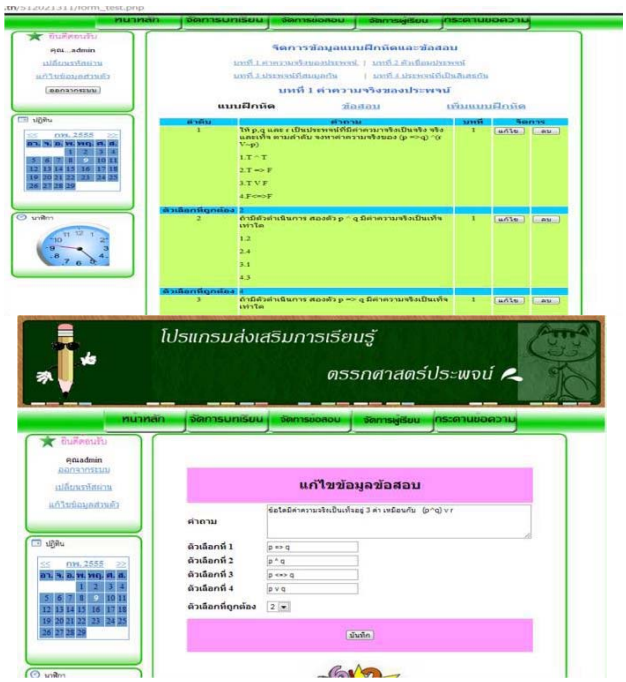


Figure 3. The interfaces for managing the exercises and tests banks.

VI. FINDINGS

User satisfaction is evaluated with respect to four topics:

1) Attractiveness of the user interface
- The screen has appropriate objects
- Screen colors are used appropriately.
- Font styles and colors are appropriate.
- System satisfactory.

2) Ease to use
- The system is linked to various screens systematically.
- The system description is obvious.
- A menu leads users to a log-in screen
- Relevant screens are easy to follow.

3) The accuracy of the system
- The system records information accurately.
- The system displays information clearly.

4) Coverage.
- The system works as planned
- The system allows for easier work.

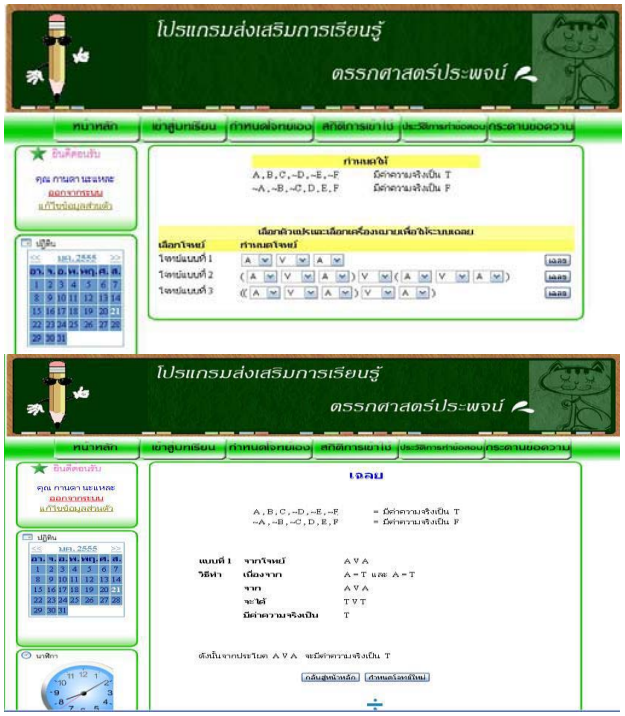


Figure 4. The interfaces for constructing learner questions which the system answers automatically.

The questionnaire was developed using "Likert scales" which contain multiple items and are therefore likely to be more reliable than single items [10]. The criteria for measuring satisfaction levels are as follows:

"Very Good": the highest satisfaction level; the score is 5.
"Good": a high satisfaction level; the score is 4.
"Moderate": an average satisfaction level; the score is 3.
"Fair": a reasonable level of satisfaction; the score is 2.
"Least" a minimum level of satisfaction; the score equal to 1.

The average ranges for all the scores assigned the following criteria:

4.50 to 5.00 is the highest assessment.
3.50 to 4.49 is a high assessment.
2.50 to 3.49 is a moderate assessment.
1.50 to 2.49 is a fair assessment.
1.00 to 1.49 is a poor assessment.

The average evaluation score for the system was 4.35 given by 30 student users in Thaksin University at Phatthalung Campus. This score is in the top-end of th highassessment range.

VII. CONCLUSIONS

This paper presents a web-based system to support the learning of propositional logic,developed using Robert Gagné theory and a rules-based approach. The system consists of two main components for the tutor and students. The tutor part

allows the management of the pre-test, postest, and exercises; tutorial contents can be inserted and deleted; student information can be searched and deleted; and student questions can be viewed and answered. The student part let the users perform the pre-test, the post-test, and exercises; the content can be viewed; profiles can be viewed and edited; test scores can be viewed; questions can be sent the to the teachers; and system auto-checking is defined.

The result of our user satisfactory evaluation survey assessed the system as being at the top of the high satisfactory range (with an average score of 4.35).

ACKNOWLEDGMENT

Our thanks go to the Faculty of Science, University of Thaksin, Phatthalung Campus for supporting our research. We also thank to the National Electronics and Computer Technology Center, Office of Science; and The Software Industry Promotion Agency for partially funding some of the research via the National Software Contest of Thailand 14 (NSC2012) during Round 1 (for the proposal) and Round 2 (for the program).

REFERENCES

[1] R. M. Gagné, "The conditions of Learning and the Theory of Instruction" (4th ed.), New York: Holt, Rinehart, and Winston. 1985.

[2] D. Kochakornjarupong, "A Learning Support Web-based System Design in Mathematics Based on Robert Gagné's Concepts of Instructional Theory". 2nd National Conference on Applied Computer Technology and Information Systems (ACTIS 2). Bangkok. 2011.

[3] D. Kochakornjarupong, "A Web-based System Design for Enhancing Learning Problem Solving in Artificial Intelligence". Special Issue of the International Journal of the Computer, the Internet and Management, 18*(SP1),62.1-62.6*, December, 2010 and in The Seventh International Conference on eLearning for Knowledge-Based Society (eLearningAP 2010) Bangkok Metropolitan Area, Thailand. 16-17 December 2010.

[4] R. Wongchosiri, "Logic" (online). Available from http://www.tewlek.com/anet-mar48-2-6.html. Accessed on 8 June 2011.

[5] A. Nokkeaw, "A Room for Basic Logic" (online). Available from http://www.il.mahidol.ac.th/e-media/logic/logic.html. Accessed on 8 June 2011.

[6] Kannin. "Basic Logic" (online). Available from http://www.bloggang.com/viewblog.php?id=krulemon&dat. (Accessed on 8 June 2011).

[7] Kruheem. "Finding Truth Value of propositional logic" (online). Available from http://www.trueplookpanya. com/true. Accessed on 8 June 2011.

[8] B. S. Bloom, "Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain". New York: David McKay Co Inc., 1956.

[9] S. W. Draper, "Feedback", 1999. Available: www.psy.gla.ac.uk/~steve/feedback.html. Accessed on 7 March 2003.

[10] J. D. Brown, "Likert items and scales of measurement?" SHIKEN: JALT Testing & Evaluation SIG Newsletter. March 2011, 15(1), 10-14.

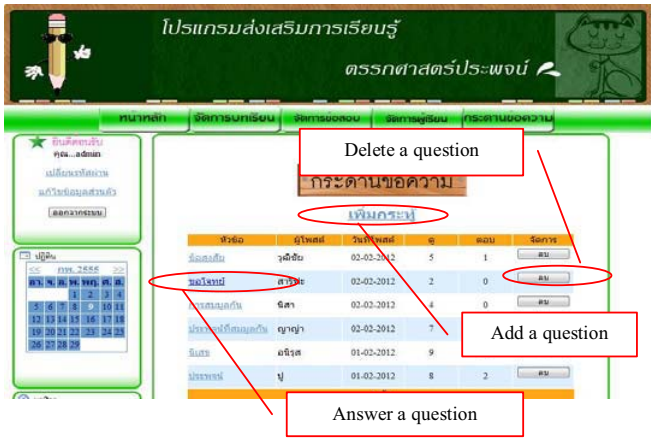


Figure 5. The web board for leaving questions and opinions, and for answering the questions.

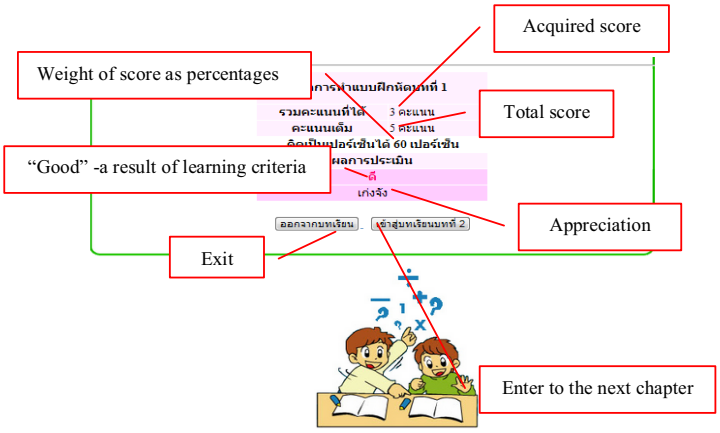


Figure 6. A short learning results report, generated after the student finishes an exercise.

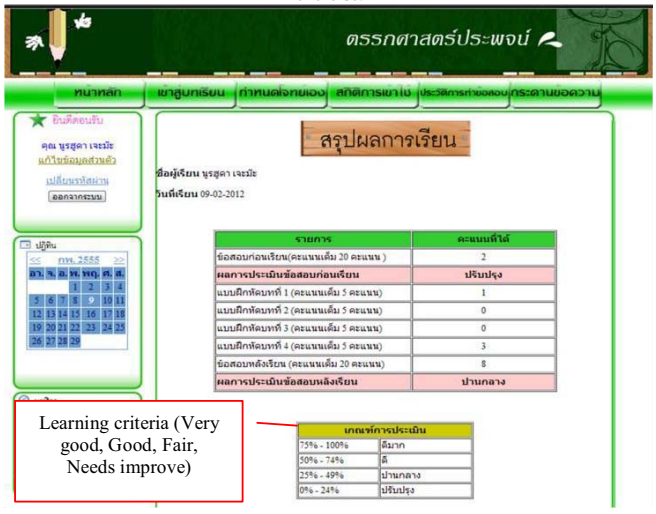
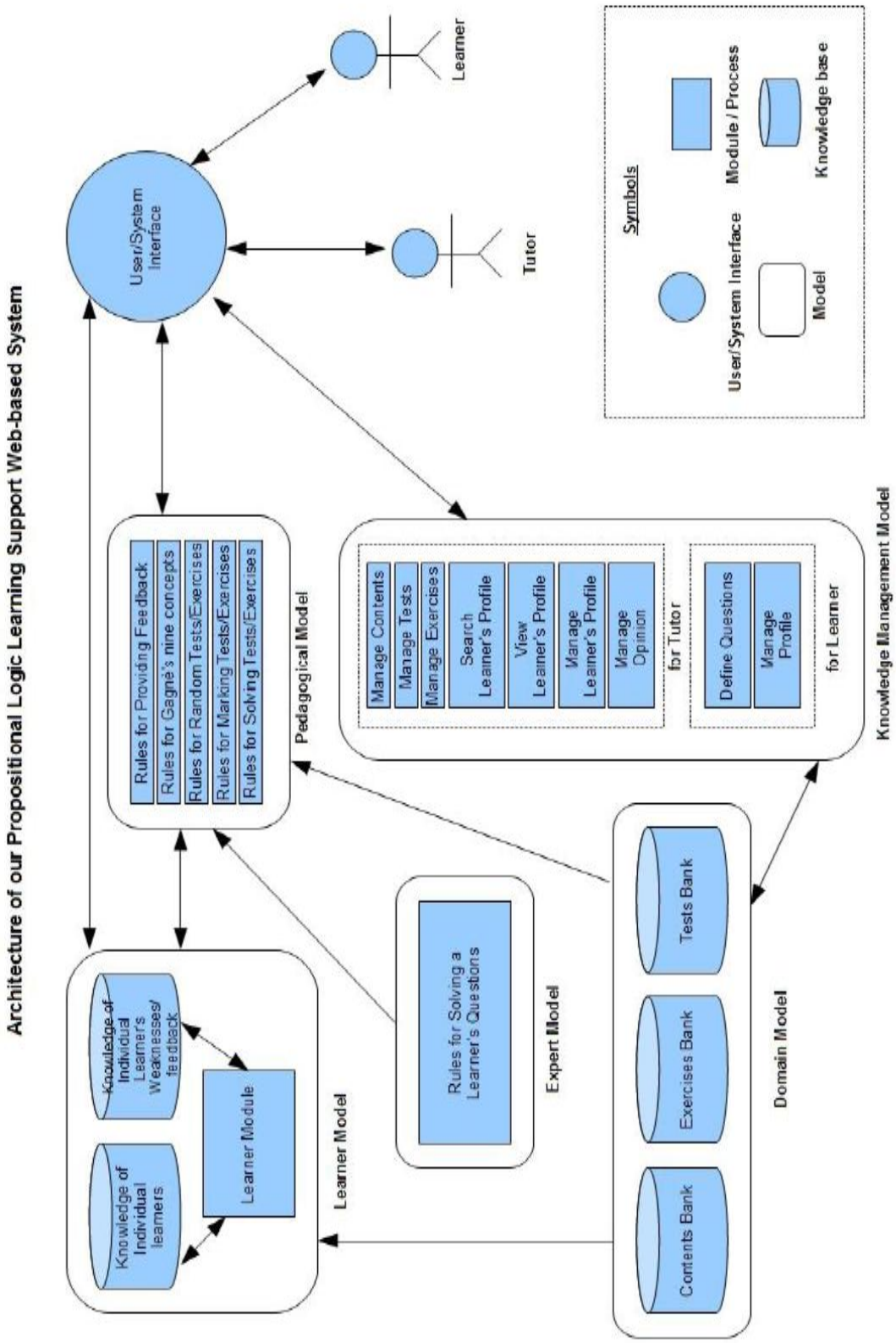


Figure 7. A full learning results report for a studunt (covering the pre-test, all exercises, and the post-test).


Architecture of our Propositional Logic Learning Support Web-based System
User/System Interface
Learner
Tutor
Symbols
User/System Interface
Module / Process
Model
Knowledge base
Pedagogical Model
Rules for Providing Feedback
Rules for Gagné's nine concepts
Rules for Random Tests/Exercises
Rules for Marking Tests/Exercises
Rules for Solving Tests/Exercises
Knowledge Management Model
for Tutor
Manage Contents
Manage Tests
Manage Exercises
Search Learner's Profile
View Learner's Profile
Manage Learner's Profile
Manage Opinion
for Learner
Define Questions
Manage Profile
Learner Model
Knowledge of Individual Learners Weaknesses/ feedback
Knowledge of Individual learners
Learner Module
Expert Model
Rules for Solving a Learner's Questions
Domain Model
Tests Bank
Exercises Bank
Contents Bank

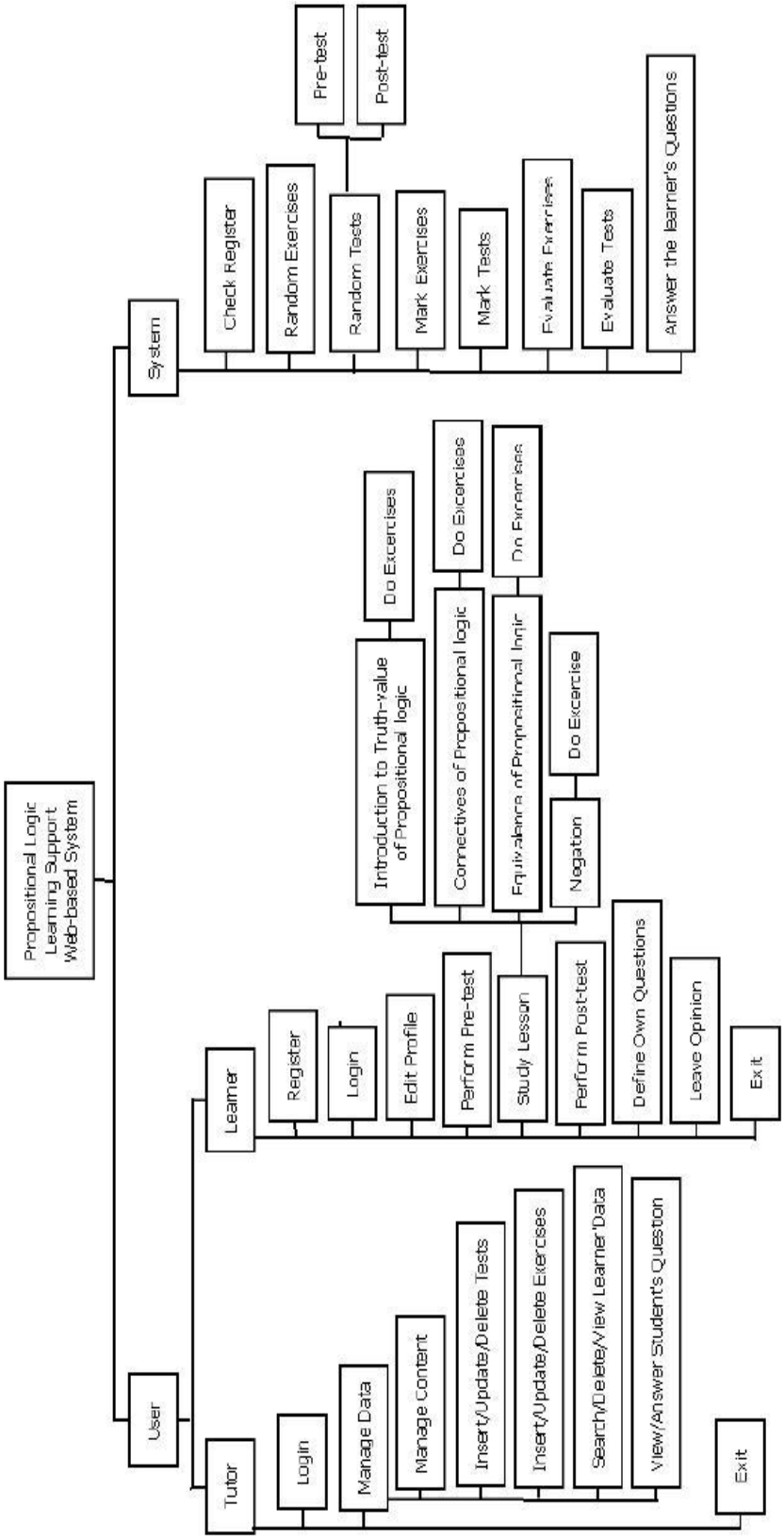


Figure 9. System Design Structure