HKDA – DESIGN PATTERN SYSTEM ARCHITECTURE

Lal Bihari Barik
Shrimad Rajandra Inst of Mgmt & Computer Application
Uka Tarsadia University, Bardoli, Surat,
Gujarat, India
lalbihari@hotmail.com

Bankim Patel
Shrimad Rajandra Inst of Mgmt & Computer Application
Uka Tarsadia University, Bardoli, Surat,
Gujarat, India
bankim.patel@utu.ac.in

Omprakash Chandrakar
Shrimad Rajandra Inst of Mgmt & Computer Application
Uka Tarsadia University, Bardoli, Surat,
Gujarat, India
op_chandrakar@hotmail.com

Ankita Barik
Vidyabharti Trust College of BBA & BCA
Umrakh, Bardoli, Surat,
Gujarat, India
ankita175@rediffmail.com

Abstract— **Software Design Patterns (DPs) have been recognized as very important and useful in real software development since they provide an elegant way of getting around problems that often occur. Researcher has built an ITS known as Human Knowledge Discovery Agent System- HKDA, which provides adaptive learning environment for the students by providing personalized instructions. The HKDA System makes inferences about student knowledge and interacts intelligently with students based upon individual representations of their knowledge. The design pattern system architecture of HKDA consists of eight main components. In this research, researchers are going to discuss about Student Modeller and Pedagogical Module. However, to provide a clear idea about design patterns, a small description about other components such as Domain Knowledge/Domain Model, Knowledge Management, Student Knowledge-base, Presentation Planner, Pedagogical Experts and Communication Knowledge are also given.**

Keywords-Design Pattern; Knowledge Component; Multi-agent HKDA System

I. INTRODUCTION

Software design patterns (DPs) are common conceptual structures that describe successful solutions to common and recurring software design problems [7]. They can be applied over and over again when analyzing, designing, and developing software applications in diverse contexts [1]. DPs are aimed at seamless reuse of software designs and architectures that have already proven their effectiveness in practice. In addition, their application guaranties high quality software solutions that are easy to maintain and extend.

It also exploits the use of multilingual and multimodal contents, which offers the students with a variety of materials, with different media and alternative modes of explanation. Researcher adapted the classical structure of an ITS with four basic components [2, 3] domain model, student model, teaching model and student's interface with other components as design pattern system architecture of HKDA.

The HKDA System makes inferences about student knowledge and interacts intelligently with students based upon individual representations of their knowledge. The design pattern system architecture of HKDA, as shown in Figure 1.1, consists of eight main components. The necessary components of an ITS as HKDA System are Domain Knowledge/Domain Model, Communication Knowledge, Knowledge Management, Student Modeller, Student Knowledge base, Pedagogical Module, Pedagogical Experts, Presentation Planner and an User Interface. Typically these components combine to generate a tutoring system's intelligence.

In this research, researchers are going to discuss about Student Module and Pedagogical Module. However, to provide a clear idea about design patterns, a small description about other components such as Domain Knowledge/Domain Model, Knowledge Management, Student Knowledge-base, Presentation Planner, Pedagogical Experts and Communication Knowledge are also given, which is as under.

II. DOMAIN KNOWLEDGE/DOMAIN MODEL

Domain Model is designed as a network of subjects. What researcher refers to as a 'Subject' is named differently in different research papers – attribute, topic, knowledge element, object, and learning outcome. Researcher uses this term to refer to set of topics that describes a single design pattern. Subject could be related to each other with prerequisite relations. One subject might be a prerequisite for another subject, i.e., the former should be learned before the latter can be presented.

For each subject, there is a knowledge threshold (measured through the score obtained on assessment mechanisms) that must be reached by a student before the system can assume that the student has learned that subject. Each subject is decomposed into units/topics and sub-units/sub-topics – content elements that correspond to a particular web page to be presented to students. The number of units/topics for a particular subject is defined by the teacher/tutor.

Whether or not it is explicitly stated, all ITS's must have a domain model. The Domain Model uses the knowledge of experts to solve domain specific problems. The Domain Knowledge contains knowledge regarding the subject being taught as well as the actual teaching material. The quality of

978-1-4577-1386-611$26.00©2011 IEEE

the domain knowledge may range from expert knowledge sufficient to solve the problem. Expert knowledge has been represented in various ways, including semantic networks, frames, production systems and constraints.

Domain Model contains the knowledge about design patterns and the actual teaching material. It organizes instructional units into a hierarchy of topics, lessons, fragments and tests, which are related by prerequisite, part-of and other relationships. Domain Model of Design Pattern Tutor is made up of subjects, which correspond to one pattern. Each subject is divided into units/topics – the elementary pieces of Domain Knowledge. There are a fixed number of units in particular concept/subject, but the size of unit is not fixed. The number of units for a particular subject is defined by the teacher.

The system uses unit variants technique [4, 5], which consists of keeping two or more alternative pages with adapted content for each knowledge level - beginner, intermediate and expert. This means that the system keeps a set of alternative units for each subject and makes a selection based on the student's characteristics (e.g., based on the student's knowledge level: very low, low, medium, high, and very high).Each unit has an arbitrary number of fragments - a chunk of information that should be presented to the user.

The Student Modeller and the subject relationships of the Domain Model provide the information that allows the system to determine which chunk of information should be presented to the user. The chunk of information may also consists of fragment variants, i.e. fragments related by an "or" relationship. A lot of work has been done with AI techniques to model student's reasoning with Back Propagation networks.

III. Student Modeller

The Student Modeller is the most critical part of an Intelligent Teaching System. If the student is modelled in such a way that the characteristics of the student are not even approximated, then the quality of decisions made by the pedagogical module will be poor. Since the Pedagogical Module depends on the Student Modeller, a wrong approximation (subject, subject units, evaluation) of the student would directly affect the pedagogical decisions made, regardless of the quality of the Pedagogical Module.

The Student Modeller evaluates the student's solutions and dynamically develops a representation of the current state of the student's knowledge and skill. The representation is called a Student Modeller, and is developed by deducing the student's knowledge from their interactions with the system. It includes long-term knowledge, such as an estimation of the student's domain mastery, as well as short-term knowledge, such as the errors that the student has made in their most recent attempt.

The Student Model also dynamically indicates the system's views of the student's strengths and weaknesses, as well as currently misunderstood knowledge including corrective advice. Other factors such as motivation can also be modelled. Typically, the long-term student model is saved when a student logs out of a teaching system, and is reloaded when he logs in again.

The Student Modeller may keep any number of students' characteristics, depending on the system requirements. In HKDA System, four basic categories of the students' characteristics are used:

A. Personal Data

Personal characteristics of a student (e.g., name, ID, and e-mail). This information represents the static part of the Student Modeller, and is collected during the student's first learning session with the system, through a questionnaire.

B. Evaluation of Performance

Cognitive and individual characteristics of a student. This part of the student model represents a mixture of static and dynamic data. Static data, such as the desired detail level, the experience level or the preferred programming language, is collected during the registration procedure (through a questionnaire). Dynamic data is derived from the learning sessions and is changed as the student progresses through the course material.

C. Manage Student Activities

Data related to the student's interactions with the system during learning sessions. This part of the Student Modeller keeps track about everything that the student has done during the learning process. In particular, it keeps data about each student's session with the system, such as the time spent on solving tests and the student's success on a particular test. This data is less important for adaptive presentation than performance data, but it is very important for reflective learning which often plays considerable role in a learning process.

D. Student Motivation

System uses this data in order to provide the student with feedback about what he has done well and where he failed, what should be revised and how to make the learning process more successful.

The student's knowledge of the teaching subject constitutes the most important student characteristic. Several representation methods are available for this [6]. Researcher has used the KAUS Model in the Design Pattern System.

Objective behind to develop such type of system is to check the student/trainer and teacher/tutor Knowledge, Understanding, Application and Skills. Therefore, whenever there is a demand of skill person in this model, Exam Agent/Expert Agent provides the percentage level to check his/her skills. Similarly, Expert Agent changes the percentage (%) of KAUS Model to keep track of Learner Agent in right path. The Objective wise weightage is given below Table 1-1 for normal University/College/School Examination:

Sr. No.	Objective	Marks	Percentage (%)
1	Knowledge	30	30
2	Understanding	30	30
3	Application	30	30
4	Skills	10	10
	Total	100	100

Table 1-1: Weightage Distribution

KAUS Model Provides Following Functionalities:

- Which topic of the various subject of syllabus should be given how much weightage can be decided.
- On the basic of the Chapters of book, the weightage of different types of question can be decided.
- Various objectives can be given sufficient justice.
- The weightage can be changed every time.
- The form of question is formed in the accordance of difficulty level.
- The measurement of remembrance ability, retention ability, obtaining skills, application etc. can be done more clearly by the Learner Agents.
- An ideal question paper can be formed.

When the student registers to the system for the first time, the newly created student model is initialized with default values from a stereotype. It is the system that selects the stereotype, based on the student's initial interaction with the system. Design Pattern gradually introduces other characteristics into the student model based on the estimated student's knowledge, such as degree of mastery, experience level, learning style, detail level, etc. Attribute values in the student model are calculated by applying a dedicated group of rules and simple functions from Pedagogical module. The values are updated throughout the session.

At the end of each session, the system stores the Student Modeller as an XML document. The next time the student logs onto the system, the data from the stored XML document are used to initialize the Student Modeller.

IV. Student Knowledge-base

Knowledge acquisition is an incremental process in which the discovered knowledge is added to the existing knowledge base. A Knowledge-base may contain questions, examples, analogies and explanations. It should also contain tutoring primitives such as topics, and also reasoning on how and when to present each primitive.

The Student Knowledge-base is always kept up-to-date and supports adaptive work of all the environment modules. This form of Student Modeller helps to define, at any moment of the learning process, what and to what extent the student has acquired (or not acquired) the material, and to accurately tune in to his level of knowledge and requirements.

V. Presentation Planner

Adaptive presentation means that students with different performance levels get different content for the same domain area. During each learning session, HKDA observes the student's progress and adapts the course presentation accordingly. Likewise, the student's performance is used to adapt the visual representation of hyper-links to related topics. The Tutoring Agent is responsible for forming an adaptive presentation of the lesson to be taught or present to the leaner as student/trainer.

Web-base Adaptive Testing is an online assessment system that can generate questions adaptively. The students will be given a pre test with different difficulty levels. The scores obtained from this pre-test will be used as the starting estimated ability value. This ability level is computed based on the KAUS Model. The question level is from one to five. Topic level is also varies from one to five. Questions are given to students based on question level and topic level. The Expert Agent, Question Agent and Answer Agent are responsible for this.

The assessment system uses the weight of the question to provide the incremental score contributed by the answer to this question. The knowledge key information about each question is used by the adaptive questioning system, together with student input and the current state of knowledge, to determine which question to pose next. After each question is answered, the question management system updates the knowledge map. This module gives results by choosing appropriate languages, tools, services and technologies helpful to all users to get all queries.

VI. Pedagogical Module

The Pedagogical Module (PM) acts as the driving engine of the whole teaching system. The primary task of the PM is to construct a plan of the course/lesson to be presented to the student, related to the learning goal(s) selected by the student. A course/lesson plan actually consists of an ordered list of subjects. To construct a plan, the subject structure and the Student Modeller are used. Due to the existence of various relations between subjects, alternate plans, teaching the same learning goal(s), can be derived for different students.

The PM represents aspects of the teaching process and is closely linked to the Student Modeller. It provides the knowledge infrastructure in order to tailor the presentation of teaching content according to the information contained in the Student Modeller. The main pedagogical tasks it is called to represent and handle are:

A. Query Mode:

When the student initiates an input query or request to its Pedagogical Agent, appropriate Web Service Agents in the system is helpful to provide the required response. Pedagogical Agents help very much in locating, browsing, selecting, arranging, integrating, and otherwise using educational material on the Web.

B. Choice of Teaching Operation:

This task provides the selection of teaching strategy and selection of teaching content. When the student enters a solution, the PM sends it to the Student Modeller, which analyzes the solution, identifies mistakes (if any) and updates the Student Modeller appropriately. On the basis of the student model, the PM generates an appropriate pedagogical action (i.e. feedback). When the current problem is solved, or the student requires a new problem to work on, the PM selects an appropriate problem on the basis of the student model.

C. Performance Analyzing:

The core competent area is to provide student support and student evaluation. It determines the timing and content of pedagogical actions, and a Student Modeller, which analyzes student answers.

D. Realization of Teaching Operations:

This task helps students for teacher support and change in course/lesson plan. This module organizing teaching for coping with individual student differences and tries to develop strategies for critical thinking.

Another task of the PM is the selection of the instructional strategy for each student. It is important for an HKDA System to offer more than one tutoring strategy, because it gives flexibility. So, there should be an effective mechanism for selecting the appropriate strategy.

Each subject is related to more than one learning unit. Based on the constructed plan and the selected strategy, the learning units are selected, ordered and presented to the student. For this purpose, the characteristics of the Student Modeller (e.g. presentation preferences) as well as the meta-description of the learning units are taken into account. In order to increase the system's teaching effectiveness, construction of the plan as well as selection and ordering of learning units should not be static but updated according to the student's performance.

Another important aspect of the PM is to evaluate the student's performance. The system is able to identify on the one hand what is wrong or incomplete in the student's answers and at the same time missing knowledge or misconception causing the error. The system is developed to analyze the final solutions to the problems, each individual solution step, or several solution steps and offer student-adapted assistance accordingly.

VII. Pedagogical Experts

Pedagogical Experts' main role is to make decisions that are used by Pedagogical Module for generating teaching plans and adaptive presentation of the teaching material. In many traditional ITS systems these roles are in the charge of the pedagogical module, but researcher decided to separate them in two distinct modules in order to increase the flexibility of the system.

One of the main tasks performed by this module is selection of instruction plan items that best suit to a particular student's knowledge level (Figure 1.1). This is a three-step process which starts with the creation of subjects plan. To create such a plan, Pedagogical Experts uses a set of facts about the student {e.g. teaching history} (from the Student Modeller), the set of available domain subjects {e.g. provide solution} (from the Domain model), and a rule for subjects selection {solution comparison}.

Pedagogical Experts selects subjects having minimal (i.e. entry level), knowledge level lower or equal to the student's current knowledge level and having all of the prerequisites satisfied. In other word, Pedagogical Experts will not choose those subjects that it believes the student is not ready to learn yet, neither will it select a subject whose prerequisite (some other subject) the student has not passed yet. This process repeats each time the student knowledge level is changed. The created subjects plan is displayed to the student in the form of adaptive subject menu.

This module is the driving engine of the teaching system and is closely linked to the Student Modeller, PM and Student Knowledge-base. It designs and controls the instructional interactions with the student. It uses the Student Modeller and the Domain Knowledge to make its pedagogical decisions. It is similar to the Domain Knowledge as keep track of the student/trainer teaching history that it must contain the information being taught to the learner.

However, it is more than just a representation of the data; it is a model of how someone skilled in a particular domain represents the knowledge. Most commonly, this takes the form of a runnable Knowledge Management, i.e. one that is capable of solving problems in the domain. By using Pedagogical Experts, the System can compare the learner's solution to the expert's solution, pinpointing the places where the learner had difficulties.

VIII. Communication Knowledge

This module observes the user/student interactions with the system, provides users with help for using tools; clarifies learning goals; gives navigation orientations; trouble shoots user's local system configurations and so on. Also it controls all such dialogues, screen layouts, etc. For instance, how should the material be presented to the student/trainer in the most effective way?

From the implementation point of view, the communication problem has been achieved by using Web Services (WS). Web Services are software components located somewhere on the Internet that are accessible through standard protocols. The syntax to call a web service can be described using standard XML-based language: WSDL (Web Service Description Language).

A WSDL document describes what a service does, how it can be accessed and where it is located, so that different clients can understand automatically how to interact with it. The integration of different modules must be transparent for the students, the platform must show a unique view of the curriculum and unified student model that all the data from the student model.

The system consists of two main modes of communication. They are:

A. Interactive Mode

The interactive mode allows the student's to practice.

B. Assignment Mode

The assignment mode allows the student/trainer to assess the competence achieved in the interactive mode.

IX. Sequence Diagram of Agent

The Sequence Diagram in Figure 1.2 shows the procedures when students answer questions from the HKDA System. The system provides to learn, give examination and performance evaluation to help students who have difficulties to improve during study. Test Paper(s)/Question Paper(s) is generated by Expert Agent as Teacher/Tutor who teaches the student/trainer will be attempted by Learner Agent.

In HKDA System, all modules combine contextual learning and constructivism methods for learning and evaluating. The steps are given below.

Learner Agent

Step 1: Select Institute where Agent belongs
Step 2: Select Course (Area) where Agent Registered
Step 3: Select Subject as Agent wish to Learn or Test
Step 4: Select Topic as Agent wish to Learn or Test
Step 5: Select Sub Topic (If Topic has subtopic else go to step 6)
Step 6: Create/Select Question along with Question type which is defined by System
Step 7: Create Question bank with answers
Step 8: Generate Question paper along with different levels of difficulty by Expert Agent
Step 9: Test generated by Expert Agent will be attempted by Learner Agent
Step 10: Learner Agent will give Answer in either Right answer or Wrong answer or Skip Question
Step 11: Generate Result according to Test answer
Step 12: Define Grade to Learner Agent according to Test Result
Step 13: If Grade level is below the expectation then goes to step 9 else goes to step 14
Step 14: Change Level of Difficulty and go to step 8

During the registration process students' valid information like institute name, course, semester etc. are filled up and on successful registration student will get user id/ member id and password from administrator. The process of generating user id is as follow.

If the user is Student and he belongs to registered institute of bpkm.org.in, then generation of userid/memberid:
user id/member id = "T"+ Expertise + year of joining + "0" + institute id + course id + (total count of user of these type + 1) e.g.

Similarly, user as Student belongs to other institute then generation of userid/memberid:
user id/member id = "T"+ Expertise + year of joining + "0" + institute id + course id + (total count of user of these type + 1) e.g.

If the user is a Trainer then generation of userid/memberid:
user id/member id = "T"+ Expertise + year of joining + "1" + institute id + course id + (total count of user of these type + 1) e.g.

For Teacher, generation of userid/memberid:
user id/member id =year of joining+"F"+instituteid + category+Department+Designation + (total count of user of these type + 1) e.g.

Similarly for Tutor, generation of userid/memberid:
user id/member id = "S"+ Expertise + year of joining + Current Industry + Function Area + Department + Designation + (total count of user of these type + 1) e.g.

These steps are helpful to Agents to know his/her knowledge space present in his/her expertise area. Almost all the modules are interacts with each other and these are regular process of Agent activities.

X. Conclusion

System provides an easy and the best way to teacher/tutor members to monitor the growth of the student/trainer by going through the vast array of useful reports generated by the system. The system provides some of the powerful features, which are basically not found in CBT, CAI systems. It traps individuals learning patterns and gives feedback based on their performances. It sets questions automatically based on such performances rather than randomly setting questions that is normally found in online examination simulators well.

This System is made in more generous that it can be used in any domain at the cost of little bit configuration and tuning. The system is reviewed by some of the industry experts and experienced academicians and has been able to win their appreciations.

REFERENCES

[1] A. Harrer, and V. Devedzic, “Design and analysis pattern in ITS architecture,” In Proceedings of the ICCE 2002 Conference, (CA: IEEE Computer Society 2002),523-527.

[2] B. F. Skinner, “The Technology of Teaching”, Appleton Century Crofts, New York, 1968.

[3] E. Wenger, “Artificial Intelligence and Tutoring Systems,” Los Altos, CA: Morgan Kaufmann, 1987.

[4] I. Beaumont, “User modeling in the interactive anatomy tutoring system ANATOMTUTOR. User Models and User Adapted Interaction,” 1994, Vol.4 Issue 1,21-45.

[5] J. Kay, and R. Kummerfeld, “An Individualized Course for the C Programming Language,” Available At: http://www.cs.su.oz.au/~bob/kay-kummerfeld.html [Site Visited: 10 Jan. 20, 2009].

[6] O. Conlan, “Novel Components for supporting Adaptivity in Education Systems – Model-based Integration Approach,” In Proc. of the 8th ACM Int. Conf. On mult., (2000), pp. 519-520.

[7] Z. Jeremic, J. Jovanovic, and D. Gasevic, “Evaluating an Intelligent Tutoring System for Design Patterns: the DEPTHS Experience,” Educational Technology & Society, 2009, Vol. 12 (2), pp. 111–130.

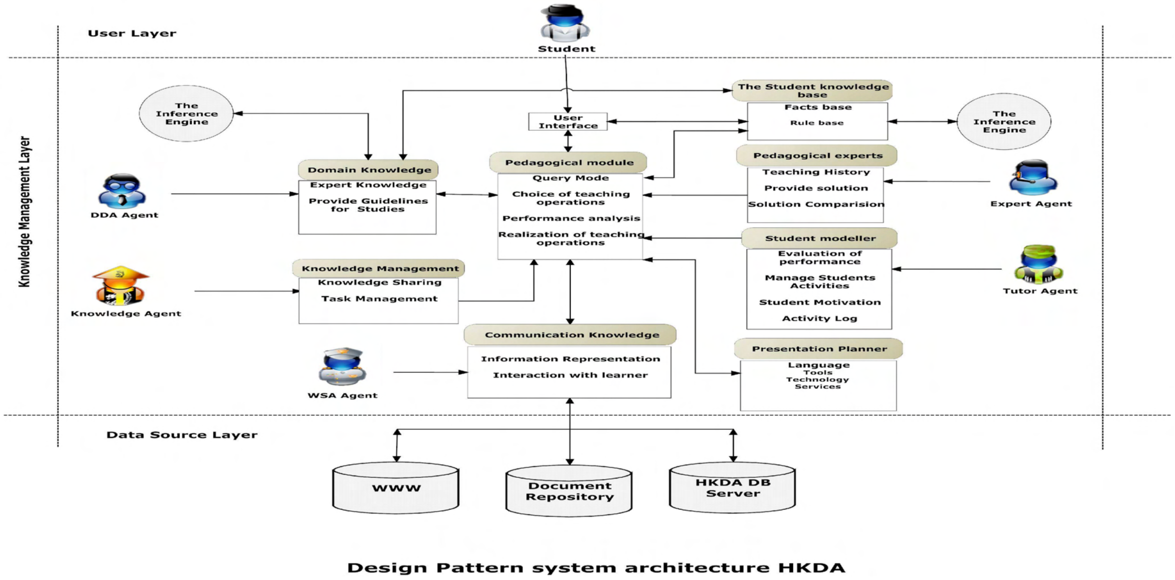

Figure:1 HKDA Design Pattern System Architecture

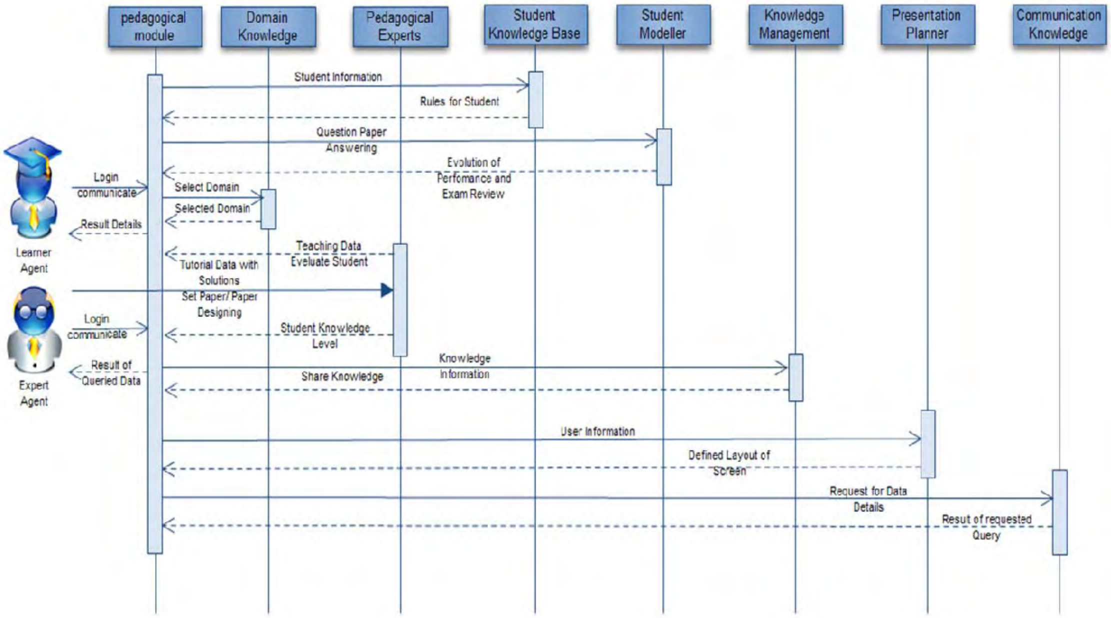

Figure: 2 Sequence Diagram of Agents