An Adaptive Model Customized for Programming Learning in E-learning

Jinghua Zhang
Department of Computer
North China Electric Power University (Baoding), NCEPU
Baoding, China
zjh_bd@tom.com

Abstract—There have been many researches on adaptive learning model or system, which are mainly for common model, common strategies and common system architecture for adaptive e-learning system. But some courses have particular characters which deserve more researches. This paper discusses the courses of programming learning for novices, such as C++ programming. We try to give a further discussion on adaptive learning support, and provide an adaptive learning model customized for programming learning course. The characters of our model are the separation of the syntax knowledge and program design knowledge, the further course resources design and organization, and then more dynamic adaptive strategies are provided based on these.

Keywords—e-learning; adaptive learning model; programming learning; adaptive learning strategy

I. INTRODUCTION

There have been many researches on adaptive learning model or system. These researches are mainly for common model[1,2], common strategies[3,4] and common system architecture[5] for adaptive e-learning system, which almost can be used by every course, though many of them give an experimental system on programming learning[5].

However some courses have particular characters which deserve more research. This paper discusses the courses of programming learning for novices, such as C++ programming, Java Programming. We try to give a further discussion on adaptive learning support, and provide an adaptive learning model customized for programming learning course. Of course, this model bases on the common model.

Research on programming psychology points out two challenges that a novice programmer has to handle [6]:

- Learning a new programming language: the student has to learn the syntax and semantics of a new programming language.
- Learning how to program a solution to a given problem: the student has to learn how to construct the solution with thoughts program used and transform the solution into a computer program.

So it has two sets of knowledge and skills, which are related but really different. Typical programming instruction tries to cover the first side—syntax knowledge, and the second side—skills in program design is instilled with the syntax learning. The knowledge of the second side is implicit which need the students perceive by himself, but we often find that many students can not grasp this kind of knowledge well with the result that some didn’t know how to design the program when facing a new problem and some gave a poor solution even it can really run correctly.

This paper tries to extract the knowledge of program design and strengthen the learning on this side. So a model with the two routines is constructed; the teaching materials are redesigned; and more adaptive methods are provided.

Next section, we do the analysis for the programming learning course and describe our ideas. An adaptive model customized for programming course is introduced in section 3. In section 4, we discuss the use of the model.

II. ANALYSIS FOR PROGRAMMING LEARNING COURSE

A. Domain Knowledge Analysis

In e-learning, the domain knowledge is divided into topics and the relations between topics are described.

The traditional partition bases on the syntax knowledge of programming language, but the program design knowledge is not mentioned explicitly or independently. This traditional method regards the program design knowledge as associate product of syntax, so such knowledge is not identified, summarized, categorized and can not impress the students, even some programming methods emerge in many examples in different syntax topics, students are still vague with them.

So this paper divides the knowledge in two ways. The one is syntax knowledge (Later abbreviated as SK) of the programming language, and relates concept, semantics, usage and run manner in the specific context. The other is program design knowledge (Later abbreviated as PDK) on how to design program which contains some programming theories, methods, skills, some usual program patterns and algorithms. An example on topic of array in C++ programming is showed in TABLE 1.
Since the PDK must have some syntax as its basis, the syntax topics are usually the main routine, and the program design topics are usually the associate routine during the learning process. So a unidirectional relation from syntax topic to program design topic is used to express the suitable point for design topic introducing.

### B. Learning Resource Analysis

In programming learning course, the course must be supported by program examples and exercises which can be seen everywhere in a lecture, so the examples and exercises are more important in the learning resources which need to be designed and organized carefully.

In e-learning, traditional programming learning resource first explains the syntax, then gives several program examples, next gives some exercises. These contents are fixed in a syntax topic lecture. The need for more materials such as examples or exercises by students will not be supported, not to mention the suitable materials according to the student states. The other hand it is also difficult to do further pedagogy design such as analogy or contrast between examples or exercises.

1) Differentiate the materials

In this paper, we differentiate these materials for the lecture. At first, we give the basic lecture material to explain the syntax which does not reference the other topics as far as possible. Every topic of PDK is also given a basic material to explain the method, the applicable problems and an example which demonstrate the application of the method to solve a real problem. The material is demanded as simple as possible. These are the basic resources of a course.

Another important part is extended resources which are categorized into two types.

The one is resources for syntax practice. These materials are pure syntax practice for syntax topics which is permitted to have not any actual meaning. The resources are given in the form of questions we usually used in the examinations including choosing, judging, reading program and giving results, reading program and filling blanks, reading program and finding errors. Every resource will give the inspection-point statements, the question, the answer and the explanation. It can test the master degree of syntax topic.

The other is resources for programming method practice. These materials are programming practice for program design topics, which demonstrate the theories, methods, skills of program design. The form of this type of resources is mainly program examples which must have actual meaning in order to demonstrate how to do when facing a real problem. Every material will give the theme, program, and all the related program design topics with program segments marked (for sometimes a resource may relate several topics). Two type’s explanations are also given: the one is program analysis; the other is design instruction step by step. Both explanations are according to the design topics the resource relates.

2) Building the relations between resources and topics

Every syntax topic should be given many resources to practice the details of the syntax topic. These questions have different difficulty levels from basic concept to complicated usage. Sometimes a syntax resource may involve several syntax topics, but should have one or more emphases. The relations between syntax resources and syntax topics must be identified for dynamic choosing.

Every design topic should be designed many materials (examples) in diversiform circumstances but actually using the same method in order to widen the sought of the students. The resources will have different difficulty levels and complexity levels. Of cause, some materials are synthesized with other design topics, but will have one or several main topics; on the other hand, the resources are programs which need the syntax support, so both the relations between resources with syntax topics and design topics must be identified.

3) Building the relations between the resources

Except the relations with topics, the extended resources have relationships with each other which are also useful on selection during learning.

Between the syntax materials, the relation is similar degree. There is many detail knowledge in a syntax topic which is not suitable to divide, and it is differentiated in the materials, so some materials are for the same details with similarity.

For the program design materials, if they are relevant from each other, it will be helpful for student to do the contrast, analogy which can strengthen the impression. So we try to design some materials with relations deliberately. The relation can be vertical or horizontal. Horizontal relation means for the same topic, such as analogy resource group (different subject with same arithmetic); vertical relation sometimes is across many topics, such as contrast group (same subject with different arithmetic); extended group (the same context but extend the subject with new topics).

### III. THE ADAPTIVE MODEL CUSTOMIZED FOR PROGRAMMING LEARNING

In order to construct a web-based adaptive learning system for programming course, the adaptive model is the most important work. According to the common adaptive model, we customize a model especially for programming learning, which includes domain model, resource model, student model and adaptive strategy.

#### A. Domain Model

Domain Model describes the division of domain knowledge and the relations of knowledge topics. We use DM to express the domain model, then DM is a two tuple: $\text{DM} = (\text{TS}, \text{R-TTS})$. $\text{TS}$ is a set of topics, $\text{R-TTS}$ is a set of relations between topics. $\text{TS}$ is divided into two sub-sets:
TS = SKTS \cup PDKTS, and SKTS is a set of topics for SK and PDKTS is a set of topics for PDK. We use t-SK to express the element of SKTS, t-PDK to express the element of PDKTS, tp to express the element of TS, r-TT to express element of R-TTS, then:

\[ tp = (ID, name, type, tp-attr) \]

\[ tp-attr = (keywords, difficulty, necessity, demand-level) \]

type is the type of the topic, type = \{atomic, composite\}; tp-attr is the attributes of tp; difficulty is difficulty level of tp, difficulty = \{easy, middle, high\}; necessity describes whether tp must be learned, necessity = \{necessary, optional\}; demand-level describes the level the student should acquired, demand-level = \{bad, lower, low, middle, high\} which is suitable to student model.

r-TT = \{relation-type, source, object, degree\}.

r-TT must describe the source topic, object topic, relation type, and relation degree; relation-type express the allow topics to select also according to prerequisite learning according to prerequisite relationships, or show all tree menu but disabled the items which are not suitable to the student model.

C. Student Model

Every student is given a model to describe his learning track and the ability-level. We use RM to express the student model, then SM = (TM, AM). TM is a track model which records learning process for every topic, including learning state, resource practiced and feedback. We use topicAM to express the track information for a topic. We use topicAM to express the learning result information for a topic.

\[ \text{topicTM} = (\text{topicID, state, learned-resources-list}) \]
\[ \text{learned-resource} = (\text{resourceID, feedback}) \]
\[ \text{topicAM} = (\text{topicID, energy, ability-level}) \]

The feedback of e-RE is different with er-type. If er-type is syntax-practice, the feedback is evaluated with two levels \{correct, incorrect\}; if er-type is design-practice, the feedback is evaluated with three levels \{not-understand, understand, program\}, which can be evaluated by student himself: whether the resource he understands or not understands or can program the example by himself.

AM is a model we create by referencing and simplifying from adaptive test thoughts and cognitive ability model. Our thought is that the student cumulates the energy during practice. For the theory that quantity change arouses quality change, so the energy accumulates to some extent, the student’s ability enters a new degree. So energy describes student’s cumulated energy during the learning which is a number between 1 and 100; ability-level describes the student mastery degree for a topic which is divided into five level \{bad, lower, low, middle, high\}.

D. Adaptive Strategy

1) Basic strategy

We first give a basic strategy on learning:

- First select a t-SK in SKTS to learn; if passed, select the proper t-PDKs in PDKTS to learn.
- For every topic, learn the b-RE in BasicRES first, and then do the practice with e-RE in ExtendedRES.
- The e-RE is not only practice but also a test, so we can adjust student model and next resource during practicing process.
- If the student’s ability-level is too low, then do the remedy.

So the strategy has 4 levels: topic selection, practice resource selection, remedy recommendation, student model update.

2) Topic selection strategy

Syntax topic (t-SK) learning is the main routine. It can have three control patterns: predefined sequence, select with tree menu but disabled the items which are not suitable to learning according to prerequisite relationships, or show all the allowed topics to select also according to prerequisite relations.
So the topic selection rule is: if all the prerequisite topics have reached the demand-level then the topic is able to learn.

Design topic (t-PDK) learning is the associate routine. When just finishing a syntax topic, the suitable topics are showed by the relation—introduce between syntax topic and design topic sequenced by relation degree but must satisfy the support and prerequisite relation. Anyway, the design topic also be supported with the tree menu if request.

During learning for a topic, first learn the basic resource (b-RE) and then do the practice (e-RE). When doing the practice, start up the practice selection strategy.

3) Practice resource selection strategy

The strategy uses the student current ability-level of the topic and the feedback of last resource and selects the most suitable resource for the next learning. The principle of selection is: from easy to difficulty, from width to depth, from simple to complex, from single to synthesize.

A topic may be found many related extended resources (e-RE) according to the related relation of r-RT in R-RTS sequenced by relation degree, but the candidate resources must be tested for every related topic must have been learned.

First check the student ability-level and confirm the resource difficulty level. According to the student’s ability on a topic which is divided into 5 levels, give a strategy for resource selection. If the level is higher than low (including low), then: if the level is low, resources with easy difficulty level will be recommended for him; if the level is middle, middle level resources will be recommended, if the level is high, high level resources will be given.

Then According to the feedback of last resource choose the next resource.
If the student ability-level is lower or bad, then start up the remedy strategy.

a) Special for syntax practice selection

If the feedback of last resource is incorrect then choose a resource with similar relation in high relation degree to this resources, if the feedback is correct then choose the resource not similar with the resource already practiced, if the learning result is better (such as three correct continuously) then can choose resource with many relations to other topics. If the similar resources have been failed with many times (such as 3), then choose a resource with low similar degree to the previous.

If the ability-level is promoted to the demand-level, then the student can choose to study the design topic or practice continue.

b) Special for design practice selection

This principle of strategy according to feedback comes from the analogy teaching method. Many similar resources can support repeated practice, which can intensify the comprehension and memory, and test the learning result till grasp. If last resource’s feedback is not-understand then recommend the similar resource especially from analogy group; If the feedback is understand, recommend the similar example especially from analogy group also, but demand the student to do programming practice; even if the ability-level change, but analogy example is preferential. If the feedback is program, give other example with low or no similar relation to widen application. If the analogy resources have been failed with many times (such as 3), then choose a resource with low similar degree to the previous.

If there are still many candidate resources, there are some principles to choose.

Resources in extended group have priority. Extended resources extend the other resources a little. This principle comes from the incremental teaching method which is apt to be accepted with grasping the new topic quickly.

Resources in contrast group have priority. This principal comes from the contrast teaching method. Contrast resources can help getting in-depth comprehension on arithmetic or widen the thought on resolving problems.

If learning a resource in group, then give the link with other resources have learned in the group.

The complexity of resource is from low to high by the student energy increasing.

The count of related topics to the resource is from low to high by the student energy increasing. The high count expresses the resource relates many other topics. The purpose of the principle is recommending resources from single problems to synthesize problems.

The principles could conflict sometimes, so a basic priority sequence is given according to their importance to the learning: difficulty, analogy group, extended group, contrast group, complexity, relation to topics, similar relations, others.

If the ability-level is promoted to the demand-level, then the student can choose to practice continue or study a new topic.

4) Remedy strategy

If the student ability-level is lower then recommend the basic resource of the topic; if the student ability-level is bad then recommend the prerequisite topics sequenced by the ability-level had acquired;

a) Special for syntax topic

Check all resources practiced with incorrect feedback, find the sub details and related topics and give hint to student.

b) Special for design topic

Check all the resources learned with not-understand feedback, find out the topics these resources include most, and give hint to student. Recommend some resources with these related topics especially in vertical relation group, such as extended group, sequenced from low to high on ability value had acquired.

5) The strategy for updating student model

If Student learns an e-RE, then an estimate is given. According to the feedback from student, update the student’s energy value. The energy’s change is interrelated to the feedback and resource difficulty. The value is experiential for the idea that different difficulty level gives different contribution for energy, and different feedback expresses the different state of student level. The value for energy up and down is detailed in TABLE 2. When the energy accumulates to some extent, the student’s ability will enter a new degree. The student’s ability has 5 levels. We give an experiential setting on the energy and ability level mapping: if the energy is less than 20, then the ability level is bad; if the energy
reaches 20, the ability enter lower; if the energy reaches 40, the ability enter low, if the energy reaches 70, the ability enter middle, if the energy reach 90, the ability enter high. During learning with a topic, a student’s energy value is set 55 at first.

<table>
<thead>
<tr>
<th>TABLE II. LEARNING ABILITY COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>not-understand (incorrect)</td>
</tr>
<tr>
<td>understand</td>
</tr>
<tr>
<td>program (correct)</td>
</tr>
</tbody>
</table>

IV. USING THE MODEL

We had given a paper on program example recommendation for student practice [7], this time we try to give a total model for web-based adaptive learning on programming. The prototype is still in developing, for there are some factors which are challenging the successful implementation and application of the model. The factors are coming from the performance aspects and the pedagogical aspects. The knowledge division, resources design and relation construction, all are long time works and need constantly add and improve.

V. CONCLUSION

The paper discusses the programming learning in adaptive e-learning system. We try to give a further discussion on adaptive learning support for programming learning course. With the analysis of the characters programming course has, we get some ideas. So we design an adaptive model customized for programming learning course, and give a detailed description on the domain model, resource model, student model and adaptive strategy.

ACKNOWLEDGMENT

The work is supported by North China Electric Power University intramural research funds (No: 200811020).

REFERENCES