Model-Based Computer Science Curricula Design

AZLINAH MOHAMED, NOOR HABIBAH ARSHAD, NORHAMIMI MOHD HIDZIR
Faculty of Information Technology & Quantitative Sciences
Universiti Teknologi MARA
40450 Shah Alam, Selangor
MALAYSIA

Abstract: - Knowledge representation is a very important aspect of curricula design. It involves dealing with qualitative data. One of the best solutions to represent the qualitative data for strategic and tactical decision is by using a model. Therefore this paper deals with model-based technique to model the domain knowledge that corresponds to Computer Science curriculum design. It would integrate the evaluation and adaptation that takes place in light of structural and/or functional constraints. Here the model is intended as interpretations of the computer science curriculum structure. This domain is suitable since curriculum designing is a complex task involving vast amount of information and requires many different types of expert knowledge. The conventional knowledge based systems lack the robustness needed to cope with the variety of combinations and requirements that can be encountered during curriculum design. This paper introduces recursive methodology in domain modelling to model Computer Science undergraduate curriculum design.

Keywords: - Model-based technique, recursive methodology, qualitative data, expert knowledge, computer science curricula design.

1 Introduction

Curriculum design is particularly challenging in academia where resources are limited and not many faculty members achieve depth in all areas of major fields. Furthermore, curriculum design requires knowledge and understanding of the domain in order for the designers to be able to develop the most appropriate curriculum model. Curriculum design also requires the designer to adhere certain procedures that are more or less structured. This paper presents an appropriate technique of modelling a structured domain area. This paper identified the use of Model-Based Technique (MBT) [2] to model this more or less structured domain. MBT is an old but still very fruitful and important area in artificial intelligence research. Nowadays MBT is used as a basic technology in several different application areas, including tutoring systems such as learning C++ [5] and Mathematical Tutoring [9], software debugging, monitoring and control of technical system in industrial sector [3], mobile and autonomous systems. Although some of the basic ideas can be taken from all these areas but there are some differences because of the different nature of all domains.

MBT requires a model of the domain to be constructed first [5]. In this case, this would be a model of the curriculum structure. This model is used to simulate the appropriate and expected basic behaviour of the curriculum structure. This behaviour would than be compared with the predicted behaviour by the user (curriculum designers). The discrepancies between these two behaviours are used to hypothesize structural discrepancies in the mental model of the curriculum designers and generate feedback to advise the curriculum designers. Finally as the domain model is complete, MBT is comprehensive in its coverage of possible behavioural (and hence, structural) discrepancies. This is not necessarily true of Rule-Based systems, which cannot address behavioural discrepancies [5]. Rule-Based systems are good at replicating the heuristics experts' use. Similarly, Case-Based Reasoning systems are primarily constrained to the types of cases already stored in the knowledge base [11]. It only may recall similar, previously solved cases and adapt their solutions to the current requirements.

For every new CS undergraduate program that would be introduced, there should be certain criteria to be looked into and guidelines to be followed. In developing and designing curriculum structure it is very important to ensure that the program has strong and clear objectives. It is time consuming and extremely difficult to incorporate all criteria into one solid model of a curriculum structure. Besides, a good knowledge-based design would require a ready to be used general rules knowledge to solve a huge class such as curriculum design. Therefore, this study would analyse and assist the curriculum designers in determining the best practices model based on their requirements. In addition, this research would discuss the recursive methodology produced in designing and generating the model.

The next section analyses related work on the application of MBT for domain modelling in an analyzing and advising applications. Section 3 would describe the domain model and how it is used to produce the solution and generate feedback. The implementation area would be discussed in section 4. Section 5 concludes the paper discussing the prototype design, contributions and limitations of the case study and further work being concluded.

2. Domain Modeling Using MBT

A model-based representation is an intuitive paradigm, which has been shown to be theoretically sound and possess some computational advantages over reasoning with formula-based representations of knowledge [12]. MBT [3] is based on explicit, behavioural models of components, which are free of any assumptions about their intended use. To perform diagnosis, these models are manipulated by appropriate reasoning routines. For example, one classic approach is to use a "perfect" model to generate the ideal operation, and then introduce faults into the model to reproduce the behaviour of the real world system. When the model's behaviour matches the actual behaviour, the faults in the model are candidates for the real world faults.

Domain Modelling does not need to include the answer (solutions) in the curriculum development. This is practical for the curriculum structure development since they can occur in a multiple different combinations. The model knows the best structure. However not every combination of states would be possible in the real world. Combination rules as a constraint for controlling purposes were used to prevent the decision support system from considering inconsistent/invalid combinations of states in earlier stages. Certain scenario fragments may contain this as a consequent state, indicating that the combination of the elements requested and prerequisite states it contains is inconsistent. Therefore, the domain model doubles as the runnable expert module. Constraint-Based Modeling [10] does not require the inclusion of a runnable expert module either. However, it targets the knowledge that prescribes user's actions whereas MBR targets the knowledge that describes the domain's behaviour. In this sense, the two could co-exist in assisting and advising mechanism. There have been more recent efforts to extend Constraint Based Modeling to include a runnable expert module [8].

Furthermore, this paper introduces the recursive methodology during the modeling process. This methodology would make the process of modeling the domain more structured, effective and efficient. The whole process indicates how the modeling process is narrowed down. It also can identify any mistake in the early phase.

3. Recursive Methodology

The aim of the overall process is to produce a complete/solid and matching model of CS curriculum structure using recursive methodology. This methodology requires the constructed model to go through two phases as shown in figure 1 below. The first phase is to generate an initial robust model. In generating a robust model, the main framework in a template form is designed and classified earlier. This model is intended as early interpretations of the target structure.

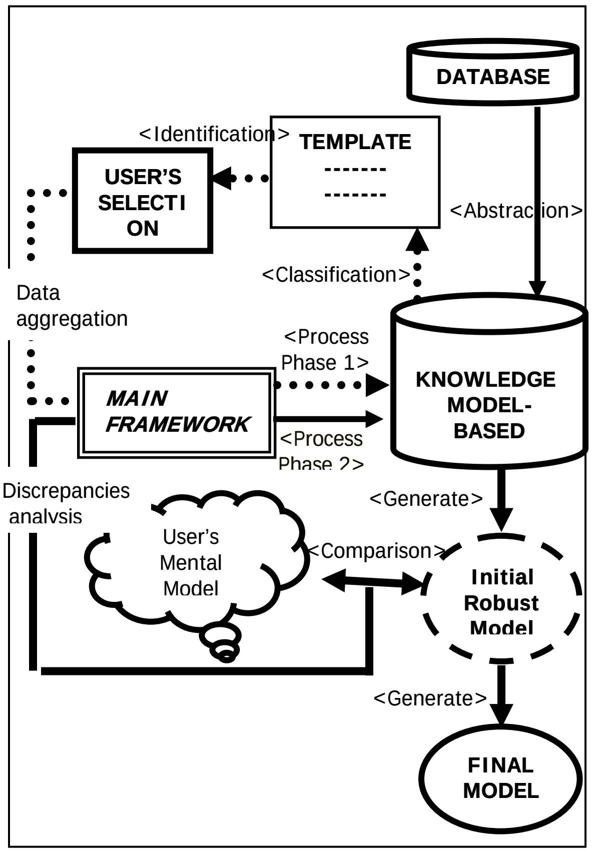

Figure 1 Knowledge Modeling Using Recursive Methodology

Users would do the selection of specific kind of template they need from the knowledge model-based. This template contains the rule of combination elements from the database that is intended as preliminary interpretations that indicates the correct behaviour of target structure. In addition, it would requires some other input regarding the curriculum criteria along with the modeling process. Subsequently, the template containing all the information gathered is analysed and verified. At this point, in order to meet the requirements such as specific range of credit hour, there would be some changes in terms of added and removed subjects. The initial model is generated and simulated to the user. The next process is to compare this initial model with the user's expected model. It is to identify the discrepancies between these two models. Beside that, the simulation model contributes some ideas to the user's mental model about their earlier selection.

Meanwhile the second phase gives the flexibility of choices whereby the user is allowed to filter the selection of the initial model's structure. The flow of second phase is as indicated using circulation arrows of 'Process Phase 2' in figure 1. However, this selection would not change the main structure as defined in the first phase. The current selection would return to the knowledge based and the process is repeated where the inference engine would analyse and interpret the changes. The whole process of modeling the curriculum structure is limited to two possible choices because the system would finally determine and view the final model. This final model is generated according to user's mental model and knowledge-based.

This methodology is different from generative architecture of a model-based tutor [5], which the hypothesis of both models is the final result of the process. Meanwhile robust MBT in crime investigation [4] directly uses the hypothesis that was created from the domain without executing the comparison process. They apply problem independent technique to those models.

4. CS Curricula Model

In this section, details about the modeling technique and recursive method to model the domain knowledge that corresponds to CS curriculum design would be discussed. There are several groups of elements in the suggested knowledge based. During knowledge acquisition, the essential elements for the standard structure of CS curriculum have been identified.

CODE:

I – University/Program Requirements
II – CS Core Subjects
III – CS Electives
IV – Free Elective
V – Internship
VI – Thesis
VII – Additional Maths

Figure 2: CS General elements in codes form

Those elements consist of university/program requirements, CS core subjects [1], CS electives subjects [1], other discipline core subjects, free electives, internship, thesis and additional mathematics which are represented in codes as shown in Figure 2.

From the analysis of internal and external universities curriculum structure and the standards put forward by the professional bodies [1,6,7], it is found that there are several common patterns of program offered. In order to determine those common structures, the pattern has been grouped and classified in models form. Each of the models constructed has it's own criteria and characteristics. There are four main models of CS curriculum that is categorised into two different modes of studies. For single major program, three different structures identified are CS (General), CS and CS (Minor), CS and Other/Option (Minor). The element of CS and CS (Minor) consists of elements from figure 2 plus CS Specialisation. Meanwhile, CS and Other/Option (Minor) consists of elements from figure 2 plus Other Discipline Core and Interdisciplinary. On the other hand, double major program offered a structure of CS (Major) and Other/Option (Major) consists of elements from figure 2 plus Other Discipline Core and Interdisciplinary.

All these elements are compiled using aggregation technique. Those elements are analysed and transformed into rule combinations. Therefore the rule combinations are the abstraction of the domain knowledge. The rule combinations are used for producing the initial model and later would represent the template for the final model. Some of the combination rules are as shown in figure 3. Figure 3 shows the example combination list of a model of CS (General) program.

Rule Combination:

I + II + III + VIII
I + II + III + VIII + V
I + II + III + VIII + VI
I + II + III + VIII +V + VI
I + II + III + VIII + VII
I + II + III + VIII + V + VII
I + II + III + VIII + VI + VII
I + II + III + VIII + V + VI + VII

Simplify

*** I + II + III + VIII = SGENERAL**

SGENERAL
SGENERAL + V
SGENERAL + VI
SGENERAL +V + VI
SGENERAL + VII
SGENERAL + V + VII
SGENERAL + VI + VII
SGENERAL+ V + VI + VII

Figure 3: Rule Combination

Based on user's choice, the initial model is intended to interpret target structure of CS design. It indicates the basic structure of rule combination as the following examples in Figure 4. This initial model is used to simulate the correct behaviour in order to get the main idea and would be compared with user's expectation. The evaluation and adaptation take place in light of structural and functional constraints.

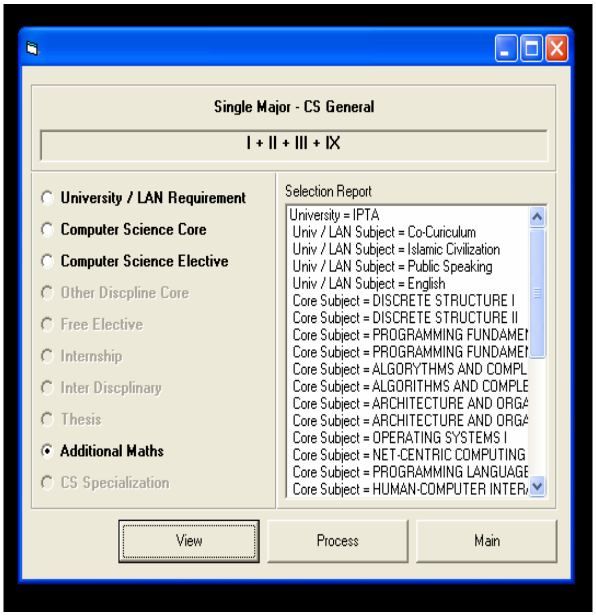

Figure 4: Initial model

This is to control from considering such inconsistent combination in earlier stage. Even though there would be multiple of different combinations, it is important to realise that not every combination would be possible in the real world. There is a complete description and rationale for each element to give the user a strong understanding. In fact there are also functions provided for the user to make changes in the initial model. Figure 4 indicates snapshot for the complete elements selection together with the list of subject.

The next phase is to generate a hypothesis according to the discrepancies between both initial model and user's expectation. Here there is a need for question-answer session to clarify the requirements. The system would then enter the second phase of the recursive technique whereby the hypothesis would go through the domain knowledge to expand the model. The system would begin to verify and integrate the constraints to the combination subjects. At this stage, the inference engine would refer back to the common structure selected by the user. Each of the common structure has its own constraints' specification such as the subject components, its percentage and credit hours. Meanwhile the inference process also includes the calculation task of overall credit hours and percentage of curriculum structure. Besides that the system would consider the curriculum designing prerequisites guideline in generating the final model as shown in figure 5.

Finally it would generate the solid/complete curriculum structure model in a framework form that contains details about course information, objectives, selection made and curriculum structure complete with the detail subject list to be displayed to the user as shown in figure 5.

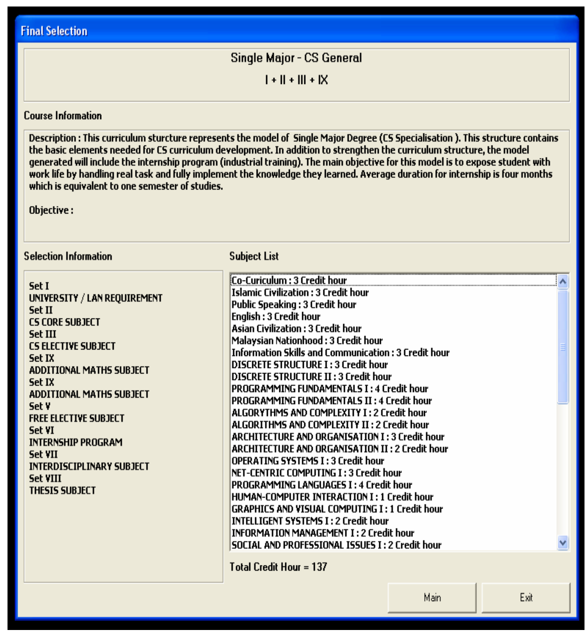

Figure 5 Sample Complete/Final Model of CS Curriculum Framework

5. Conclusion and future work

This research has shown that by implementing MBT using recursive methodology would identify conflict at an early stage. Therefore other processes in producing the knowledge model would be more effective. In addition, the constraints for template designing are important to ensure the combination is valid for the real world situation. Aside from these, a number of important theoretical issues remain to be addressed in future research. In particular, an approach to compute with crude probabilities, suitable for curriculum development and design, must be formalized. The combination of compulsory elements contained in the template may be constrained by time durations in addition their relative orders. Moreover, this research can be enhanced by considering to add another technique known as Case-Based Reasoning once the system is executed. This is to recall similar, previous solutions and putting learning elements as well.

The current situation shows that the number of Computer Science (CS) undergraduate programs being offered whether by public or private universities and colleges in Malaysia has increased each year. The CS curriculum structure has also been focused and given priority due to market requirements and government policies towards producing knowledgeable graduates in the field of Information Technology (IT).

The development of the system in assisting the aforementioned domain is expected to improve the understanding of the domain structure. The system would help and assist curriculum designers by giving them the knowledge required in developing the CS undergraduate curriculum structure for the public and private higher learning institutions. Besides, the methodology and technique suggested could also be deployed on any domain knowledge related to curricula design.

References:

[1] Computing Curricula 2001 Computer Science (CC2001), The Joint Task Force on Computing Curricula, IEEE Computer Society and Association for Computing Machinery, Final Report , December 2001.

[2] Davis, R., Diagnostic Reasoning Based on Structure and Behavior. Artificial Intelligence, 24, 1984, pp. 347-410.

[3] Hunt, J.E., Model Based and Qualitative Reasoning: Industrial Application, 1st International Workshop on Model-Based and Qualitative Reasoning --- Perspectives for Industrial Applications, ECAI '96, Budapest, 1996, http://ai.di.fc.ul.pt/Ecai-96.htm (25 February 2003).

[4] Jeroen, K. and John, Z., On the Role of Model-based Reasoning in Decision Support in Crime Investigation, 2002, http://homepages.inf.ed.ac.uk/jeroen/lawtech2002.pdf (3 March 2003).

[5] Kumar, N. A., Model-Based Reasoning for Domain Modeling, Explanation Generation and Animation in an ITS to help Students Learn C++, 2002, http://monet.aber.ac.uk:8080/monet/docs/pdf_files/its_02/kumar.pdf (21 February 2003).

[6] LAN-TC-01 2nd Edition: Procedures for Approval of Certificate, Diploma and Degree Courses of Study (PHEI), http://www.lan.gov.my/english/index2eng.html (January 2006).

[7] LAN-TC-02 1st Edition: Procedures for Approval of Postgraduates Courses of Study (PHEI), http://www.lan.gov.my/english/index2eng.html (January 2006).

[8] Martin, B. and Mitrovic, A., Tailoring Feedback by Correcting Student Answers. Proceedings of Intelligent Tutoring Systems (ITS) 2000. G. Gauthier, C. Frasson and K. VanLehn (eds.), *Springer,* 2000, pp. 383-392.

[9] Neuper, W. and Wotawa, F., Model-based Reasoning in Mathematical Tutoring Systems – Preliminary Report, 2002, http://monet.aber.ac.uk:8080/monet/docs/pdf_files/its_02/nueper_wotawa.pdf (21 January 2003).

[10] Ohlsson, S., Constraint-based Student Modeling. In J.E. Greer, G. McCalla (eds.) Student Modeling: The Key to Individualized Knowledge-Based Instruction, 1994, pp.167-189.

[11] Reyes, R.L. and Sison, R., A Case-Based Reasoning Approach to an Internet Agent-Based Tutoring System. Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future. J.D. Moore, C.L. Redfield and W.L Johnson (ed.), *IOS Press*, Amsterdam, 2001, pp. 122-129.

[12] Thomas, E., Computing Intersections of Horn Theories for Reasoning with Models, 1998, http://citeseer.nj.nec.com/537461.html (23 March 2003).