Exploring effective approaches in teaching *Principles of Compiler*

ZhaoHui Li
Computer Science Department
Guangzhou University
People's Republic of China

Lizhaoh888@hotmail.com

Mentor
Bing Bing Zhou
School of Information Technologies
Faculty of Science
The University of Sydney

Abstract

In this paper, the problems that we are facing in teaching *Principles of Compiler* are analysed and some modifications on this course to incorporate effective contemporary student-centred teaching approaches are discussed. These include concept mapping, problem-based learning (PBL), case study and e-learning. The possible challenges in implementing these teaching methods are also investigated.

Introduction

It is well known that our world is changing rapidly and our students are facing a changing world. Higher education needs to adapt to such trends to ensure that students are well trained to fulfil the needs of the industry.

As a visiting scholar of China, I was fortunate to have the opportunity to take part in the *Teaching Science in English* program at The University of Sydney from June to October 2006 and have learned a lot about these issues and contemporary teaching and learning approaches such as concept mapping, case study, PBL, e-learning and so on. What we observed at The University of Sydney made us understand why the contemporary teaching approaches are powerful and effective. These approaches make students think critically and be responsible for their own learning.

This paper first describes the course of *Principles of Compiler* and its present situation. It introduces what I have learned and observed at The University of Sydney and my reflection on the current teaching and learning approach. Some modifications I should make on this course and the possible issues I will encounter are also discussed.

Course introduction and the present situation

Principles of Compiler is a compulsory course in computer science. It introduces the theories used to build compilers for high level programming language and algorithms. Students are required to have prior knowledge like discrete mathematics, programming language, data structure, assembly language, operating system, computer organisation and structure. The course covers lexical analysis, syntactic analysis, syntax directed translation, intermediate languages, code optimisation, code generation, error handling, run-time systems and memory allocation.

The course at the Computer Science Department of Guangzhou University in China is offered to the third year undergraduate students. It consists of 56 teaching hours in one semester including two sessions: lectures (36 lecture hours) and laboratory work (18 hours). About 250 students take this course each year. Three teachers share the teaching load. Each has about 80 students in one lecture.

It is a challenging course for the students since most topics are quite theoretical, and are often conceptually difficult for students to understand. The algorithms covered in this course are more complex than those in other courses. Moreover, the course does not have a strong engineering perspective that can link itself directly to a career in industry. Sometimes students may doubt the value of a course that is quite difficult and at the same time too theoretical to benefit a career in industry. It turns out that many students spent insufficient time on this course and each semester the failure rate is over 20% each semester.

A high level of difficulty and certain features of the course content contribute to the lack of student motivation. However, what I was not aware of before was that the teaching approach might have contributed to it also.

What I have learned and observed in The University of Sydney

At The University of Sydney, we have learned a lot of contemporary teaching theories and approaches.

There is a trend from behaviourist view of teaching and learning to a more constructivist and developmental view. This leads to the movement from a strongly teacher-centred approach to a more student-centred approach. It is important to recognise that students must be active participants in the learning process wherever possible, in order to promote deep level processing of knowledge and encourage them to take greater responsibility on their own learning. (King 2006)

Thus, higher education should focus on students' intellectual and imaginative powers development. It should also focus on improving students' problem solving and communication skills.

Accomplished teachers are expected to have a rich understanding of the subject knowledge they teach and appreciate how knowledge of their subject is created, organised, linked to other disciplines and to real world settings. They must also try to develop critical and analytical capacities in their students. They should command specialised knowledge on conveying subject matter to their students. They should adopt a wide range of teaching strategies that enable them to organise, adapt and present the curriculum in ways that take account of the specific teaching and learning context. They must understand where difficulties are likely to arise and modify their practice accordingly.

To improve teaching and learning, there are some effective approaches such as concept mapping, problem solving, PBL, case studies, workshop tutorial, e-learning and so on.

The reflection on the current teaching and learning approach

During my study at The University of Sydney, I realised that several issues in the *Principles of Compiler* course might be related to my teaching approaches. This course was taught and learned in a traditional way, with little interaction and little use of contemporary approaches. Usually, there is only one-way communication in the lecture with the lecturer talking while the students listen through out the whole lecture hour. There was little response from the students to questions deliberately asked by lecturer. The lecturer puts more emphasis on explaining concepts, theories, algorithms and explained them as clearly and detailed as possible, but seldom gave students practical examples. Students were required to do exercises but were seldom given real-world problems. During laboratory hours, most students worked individually on set tasks. They seldom cooperated.

This approach is not effective in motivating and engaging students with their study. They passively accept selected information provided by their lecturers but soon forget much of what they have learned.

But in The University of Sydney, the students learn in a different way.

Firstly, the university curriculum gives students flexibility to arrange their learning and choose courses. They can choose between full-time learning and part-time learning; they can take units of study from other faculties; they can get credit for previous tertiary study; students that perform well have the opportunity to enrol in advanced units of study or take the Talented Student Program (TSP) tailored to meet students' individual needs; they can enrol in various combined degrees such as science/law, science/commerce, nursing/science, education/ science, and engineering/medical. They can take many electives courses in which they are interested. In contrast, at Guangzhou University, the students can only undertake full-time study; they have much less choice in taking electives from other faculties or from their own discipline; there is no advanced units of study option or TSP for them; usually students enrolled in the same year and the same department will form one or a few relative stable classes and the class will do courses together through out the four year study. Also, there are no combined degrees.

Secondly, at The University of Sydney, the students are more motivated to self-education. They are required to consult a lot of references on the Internet or from the library, not just from textbooks, many students do not even have a textbook. In Guangzhou University, every student is required to buy textbooks with most teachers teaching only the contents from the textbooks and students mainly learn from the teachers or from textbooks. In lectures, the teachers at The University of Sydney don't teach students every detail as we do in China. They just focus on selected key content. The students can ask and be asked questions at any time during class. During classes, the teachers use more demonstration and practical questions to help teaching.

Lastly, in The University of Sydney, the teachers pay more attention to practical application problems. The students are asked to do problem-based assignments, practical projects and presentations. Their critical thinking, problem solving and communication ability can be improved greatly. During laboratory class, there are some tutors chosen from senior students to guide students in solving problems. This can not only alleviate the teachers' labour but also improve the students' abilities in problem solving and communication.

Based on the above reflection and what I have learned and observed at The University of Sydney, I will try to implement some modifications in my course.

Modifications I should do on this course

In order to trigger the students' passion and make learning relevant to students and improve the students' abilities, I would like try to make some modification in this course by using some contemporary teaching approaches and show the practical application of the discipline to students.

Table 1. Teaching plan in *Principles of Compiler*

No.	Chapter	Main contents	Teaching Approaches
1	Introduction	Overview of compiler The process and structure of compiler The exploitation of compiler	Concept mapping E-learning
2	Lexical analysis	Designing method of lexical analysis A simple example of lexical analysis Regular expression (RE) and finite state automata (FSA) The construction from RE to FSA	Concept mapping PBL (case study) E-learning
3	Syntax analysis	Grammar and language Induction and syntax tree Recursive-descent parsing LL(L) parsing Operator priority parsing LR parsing	Concept mapping PBL (case study) E-learning
4	Semantic analysis	Overview Attribute grammar Several intermediate language Expression translation Control sentence translation Array element translation Procedure and function translation Recursive down syntax-directed translation	Concept mapping E-learning
5	Code optimisation and generation of intermediate code	Local optimisation Recycling optimisation Example of optimisation	Concept mapping E-learning
6	Management of run-time storage space	Static storage allocation Simple stack storage allocation Stack realisation of nesting procedure Dynamic stack storage allocation	Concept mapping E-learning
7	Generation of target code	A simple code generator	Concept mapping Case study E-learning
8	Symbol table management and error handling	Symbol table Error handling	Concept mapping E-learning

Table 1 is the detailed plan of the teaching approaches which will be used in this course

Using concept mapping

Concept mapping used in this course is for representing knowledge graphically in a network of interconnected concepts. It can be used to: generate ideas, design complex knowledge structures, communicate complex ideas, aid learning and assess understanding. It can also help to explain the importance of a particular aspect of a topic so that students can see how particular pieces of information fit into the overall schema. It can help students retain a mind map of the information they are studying and understand why they are learning it (King 2006).

I plan to use concept mapping at the beginning of the course to show the relationship of this course with the other courses (see Figure 1). This is important to help students establish the whole professional schema. I will also use concept mapping to show the relationship of the main topics with this course (see Figure 2).

I will use concept mapping in every chapter. At the start of a chapter, it is used to help students establish their preconceptions. It will also be used at the end of a chapter to help students figure out what they have learned and what they still do not understand, and thus help the lecture to know whether or not the students understand the concepts.

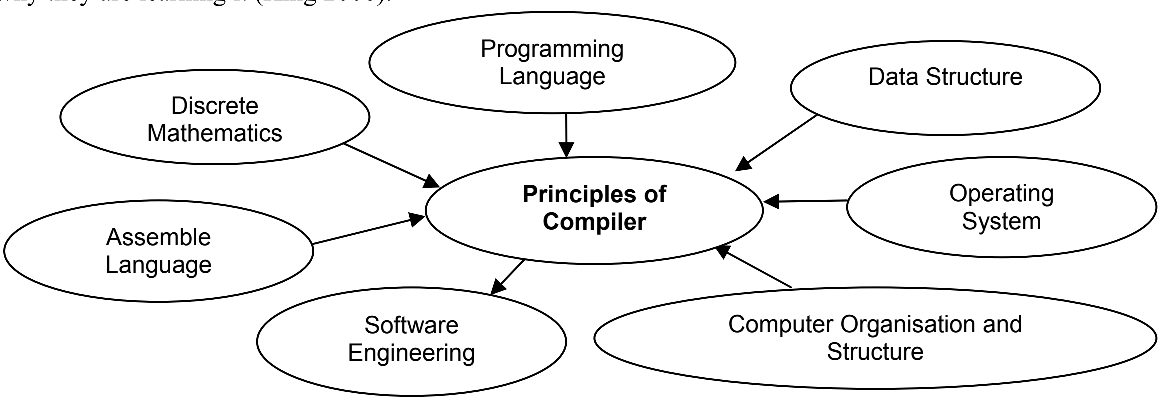

Figure 1. Concept mapping of *Principles of Compiler* with the other courses

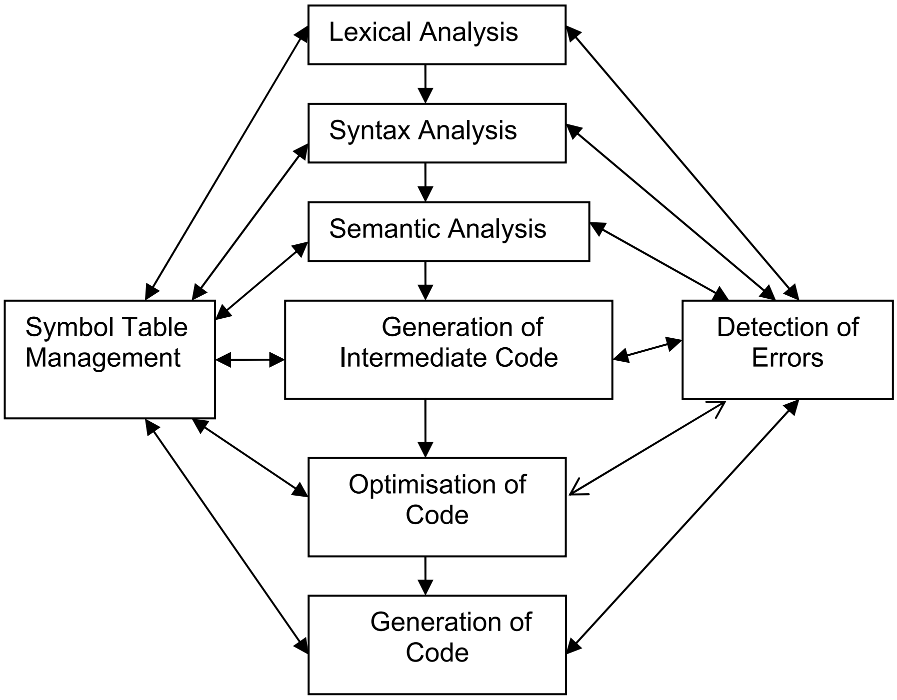

Figure 2. The concept map of main topics

Using PBL

How to get the students to think and 'learn to learn'? PBL is an instructional approach which challenges students to learn by working cooperatively in groups to seek solutions to real problem. PBL used in this course can engage students in collaborative learning. Students acquire and apply knowledge in their quest for solutions. Guided by teachers acting as tutors, they develop critical thinking, problem solving, and collaborative skills. PBL enables them to establish relevance between knowledge and real problems in their learning. This would in turn enhance their creativity and problem solving skills. It can increase their interest in the course; increase their motivation to learn science; make them more active in learning; improve their problem solving skills and lifelong learning skills (King 2006).

Problems must come from the real world or from a real engineering situation. I think that it seems that it is not easy to find a real-world problem in computer science, it is more reasonable to find a real engineering situation problem. However, whether the problem is from the real world or from a real engineering situation, it must stimulate the students' curiosity and motivate them to learn the course. The problems must also lead the students to identify the necessary skills for solving the problem effectively. This learning provides realistic situations where knowledge is applied, and thus, encourages a better understanding of the knowledge (Khoumsi and Gonzalez-Rubio 2006). In this way, PBL encourages active learning, students are more responsible in the learning process.

This approach is a student-centred teaching approach and we can use it to improve our teaching. However a pure PBL approach is not always suitable for all students. The PBL teaching experience in School of Information Technologies at The University of Sydney showed that the result of PBL was not satisfactory in first year undergraduate students. So we have to be careful to plan PBL teaching.

In this course, although our students are third year students, we do not plan to use PBL in the whole course. We will just use it in Chapters 2 and 3, because the content in these chapters is very important and is sufficient for learning many of the basic principles of a compiler.

A sample problem

> *From an advertisement in the newspaper, you know there is a computer company who is willing to pay $2000 dollars for the design and construction of a lexical and syntax analyser. In this company, a high-level programming language is used for the development of programs. A compiler for the target environment has been developed. For some reasons, the existent compiler can not be used. The manager of the company has decided to replace the target environment, so a new compiler must be designed. In order to avoid having to design a whole compiler, the manager commissions somebody to design and construct the front end of the compiler – the lexical and syntax analyser.*

In this program, the solution to the problem is not unique. There is some space for students to explore this problem. Below are a few sample problems that might be given to the students.

1. How to describe lexical units formally by using regular expressions and finite state automata?
2. How to describe the syntax formally by using a grammar? How to analyse and manipulate grammar?
3. How to design and realise a lexical analyser (to design an analyser that reads a source code, checks whether the lexical units in the source code are accepted and returns the recognised lexical units)?

4. How to design and realise a syntactical analyser (to design an analyser that reads a source code and checks whether the source code is accepted by the grammar)?

Student activities

Group discussion

The students can be divided in groups (6-8 persons). Each group will have a tutorial meeting. Under tutor guidance, students use acquired knowledge and collaborate to find solutions to the problem. The tutor's role consists of asking questions, making comments, validating students' solutions, reflect on what they have learned and so on but not presenting solutions to the problem.

Experimental work

This activity aims at practicing knowledge related to the solution alternatives to apply the theory knowledge into practical situation.

Independent study

In order to solve the problem, the students have to do independent study. They have to search materials and learn new knowledge and seek solutions to the problem, thus they develop their self-directed and lifelong learning skills. Skilled learners are more in control of their own learning; create intrinsic motivation rather than extrinsic. They learn in a more active and varied way. They are aware of how to learn so that they can continue to learn into the future.

Mini-lectures

During the PBL teaching, I will give students some mini-lectures. The main purposes of giving mini-lectures are to introduce some basic conceptions, explain some difficult theories which can not be understood by students and answer some questions asked by many students and so on.

Using case study

A case study is a student-centred teaching strategy beginning with a story and educating students through the story. In this strategy, students learn particular concepts, issues or topics through a complete and real-world case.

For example, a case study will be introduced in chapter 7.

> *Last Summer Vacation, I was introduced to work on a program for a computer company by my friend. On the first day, I met my boss who told me to analyse an existing program of generation of code which was part of a compiler. Then I was introduced to an experienced systematic programmer, Alice, who gave me a 'guided tour' of the actual program code. After my tour, I had to analyse: Could this program fulfil the task of translation from intermediate code to target code i.e., generation of code? Was this program optimisation ? The students are asked to study this case by discussing, questioning and practicing.*

I will also introduce case studies at the beginning of PBL in Chapters 2 and 3, for students may think it is very hard for them and feel confused at the beginning and don't know how to start. After a case study, they may feel relaxed and be confident to carry out PBL.

E-learning

E-learning is a good approach for teaching and learning. I found *WebCT* at The University of Sydney was a good example. The learning and referencing materials of every course can be found on *WebCT*. This helps students' self-directed learning. I am very impressed by the *WebCT* platform. When I go back to China, I hope I can construct an e-learning platform for this course. With this platform I will put up relevant materials such as the lecture notes, assignments, practical problems and research papers on the Internet. In lectures I can then focus on key concepts and students can access the rest of the materials from the e-learning platform. I will also try to design a self-assessment tool on the Internet to obtain feedback from students. This learning will be arranged in every chapter in this course.

Some other approaches

There are some other approaches which might be used to stimulate students' interest during class. For example, at the beginning of class, sing a song to keep teacher and students' in a good mood; I may also design some interesting games which are related to what they have learned. It is also helpful to give students some questions for them to prepare for the next lecture at the end of class.

Challenges

Changing teaching approaches will be a big challenge.

The teachers should have a deep and wide understanding of what they will teach. They should gain practical experiences in addition to textbook knowledge. This is especially important to IT teachers. In order to catch up with the newest development of IT, they should do more practical projects to improve their own practical abilities, otherwise, how can they instruct the students in PBL teaching? It will take teachers a lot of time and energy to prepare the course. But the teachers time and energy are limited, take me as an example, the teaching task for me is heavy, usually having 8-12 lecture hours per week, and I also have to do research work. So, how can I have enough time and energy to prepare myself and lots of materials for the new teaching and learning approaches? Besides, the teachers should have good organisation of the students' activities during PBL teaching, otherwise it may give rise to chaos. Moreover, because of the large size of classes, we should train more tutors to help. The cooperation of other teachers to keep up with the teaching pace is also very important. But how to motivate the other teachers?

For students, they may not like the new teaching and learning approaches. They get too used to being spoon-fed by teachers from kindergarten, they expect to be told what to do and how to do it. They lack the independence and self-discipline. They will think it is too hard for them to take more time and energy to learn this course.

For the administration, teaching reform is not an individual undertaking, it must be supported by the administration of the university. If I do not have their permission to try the new teaching approaches, they may think I am not teaching!

Therefore, it is very important that all parties involved, including administration, teaching staff and students to cooperate.

Conclusions

This paper discussed some possible modifications on *Principles of Compiler* course. Some effective contemporary student-centred teaching approaches such as concept mapping, PBL, case study, e-learning will be introduced to this course. These approaches will make the students active and responsible for their own learning and thus make the course more interesting, relevant and motivating. It will improve students' skills for communication, problem solving and lifelong learning. But when we try to change our teaching approaches, we will encounter some big challenges, the challenges come from administration, teaching staff and students. We can only make the change gradually.

Acknowledgements

First of all, I would like to express my sincere acknowledgements to The University of Sydney and the China Scholarship Council for their support. I would like extend my warmest thanks to Associate Professor Mike King, Dr Tom Hubble, Associate Professor Mary Peat and Tony Sperring, for their effort to introduce contemporary education theories to us and share with us of their years of experience in teaching; I'd like to thank my mentor Professor Bing Bing Zhou for his instruction and all the teachers especially our English teachers Asher, Cecelia, Luise, Kathy, Sophie and those professors who gave us interesting seminars; I'd like to thank Janet Conroy, Didi, Fiano and the other officers for their useful help. I also appreciate Consulate-general of China in Sydney for their kind help and support. Finally, I'd like to give my best wishes to The University of Sydney.

Reference

Ben-Ari, M. (2001) Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, **20**(1).

Boud, D. (1985) *Problem-based learning in education for professions.* Sydney: Higher Education Research and Development Society of Australasia.

Hamalainen, W. (2004) *Problem-based learning of theoretical computer science*. 34th ASEE/IEEE Frontiers in Education Conference. Savannah, GA.

Khoumsi, A. and Gonzalez-Rubio, R. (2006) Applying a Competency – and Problem-Based Approach for Learning Compiler Design. *Journal of STEM Education*, **7**(1,2).

Khoumsi, A. and Hadjou, B. (2005) Learning Probabilities in Computer Engineering by Using a Competency-and Problem-Based Approach. *Journal of STEM Education: Innovations and Research*, **6**(3,4).

King, M. (2006) Course delivered to Teaching Sciences in English program. A collaborative project between the China Scholarship Council and The University of Sydney.