

Research and Application of Iterative Approach in Computer Language Teaching

Jiang Guangyuan

Software Engineering Department of Software School, Dalian University of Technology, Dalian, P.R. China

Abstract

Language points are explained to students one by one in traditional teaching, and their varieties make the first learning of a computer language rather difficult. All the language points in computer language courses are successive and interdependent, so any defect or misunderstanding would bring barriers to the subsequent learning. Based on the natural way of language acquisition of human beings and iterative models in software engineering, an iterative approach is put forward in this paper in order to maintain the consecutiveness of the language points and help students in their software research and development.

Index Terms: component; iterative approach; computer language; language points

© 2011 Published by MECS Publisher. Selection and/or peer review under responsibility of the International Conference on E-Business System and Education Technology

1. Background

Computer language courses are the basis of the computer specialty and other related specialties. After learning these courses, students can solve related problems with a computer, grasp common algorithms, and comprehend the central idea of program design^[1]. Our practice has proved that the teaching of computer languages plays a very important role in the teaching of the follow-up related courses.

In the common teaching approach, language points are explained in sequence one after another and a great number of grammar details need to be memorized. The use of certain points is sometimes not quite clear to students and makes them frustrated in learning^[2]. Moreover, the difficulties and defects in the learning of one language point may bring barriers to the learning of the follow-up points. Therefore, the students would likely lose interests and even lose heart in learning^[3].

Human being's language acquisition begins from imitation in which the learners don't understand the grammatical rules of the language. Usually the structure and grammar of language are learned in people's life and improved through speaking, reading, and writing.

Iterative approach is often applied in software development because it is almost impossible to know all the problems to be solved at the very beginning. Therefore, a basic structure can be set up first and then be improved by repeated iterations to correspond to the target system^[4].

* Corresponding author.
E-mail address: jgy@dlut.edu.cn

Computer languages are obscure for beginners because of their various and numerous language points, so it is rather difficult for most of them to grasp these languages. Computer language learning and the understanding of problem domain in software development are both similar to human being's natural language acquisition. Therefore, the human being's natural language learning process and the iterative approach in software development can be combined and applied in computer language teaching, which can be called "iterative teaching approach".

2. Model

Based on iterative teaching idea, this paper discusses how to teach through repeated iterations without changing the length of teaching session. The following three factors are considered: 1. times of iteration and teaching effect; 2. language points (the language points are closely interrelated and successive, so it is necessary to discuss the teaching effect of each language point in each iteration.); 3. mastery rate (the proportion of the students who master the language point in the iteration in those who did not master it before the iteration).

The consecutive dependence of language points is obvious in science and engineering disciplines. The mastery of language point No. j is divided by that of No. $j-1$, the IF (impact factor) is β .

The traditional teaching carried out in the sequence of language points is called linear teaching approach, which can be described with the following model:

Suppose the total number of students is s , the number of language points is m , the mastery rate of each point is α , and the number of the students who can master the language point No. j is P_j .

$$P_j = P_{j-1} \times \alpha$$

In which $1 \leq j \leq m$, $0 \leq \alpha \leq 1$, $P_0 = s$.

It can be seen that in linear teaching approach the mastery of a language point is greatly influenced by that of the previous one.

Impact factor β is:

$$\beta = P_j \div P_{j-1}$$

In which $1 \leq j \leq m$, $P_0 = s$.

It can be seen that the value of β is actually equal to that of α . The mastery of the latter attenuates quite fast and the mastery of the current point won't possibly be used once more. Therefore, iterative learning could provide another chance to digest the previous point and continue the learning process.

Iterative teaching approach can be described with the following model:

Suppose the total number of students is s , the number of language points is m , the times of iteration is n , the mastery rate of iteration No. n is α , and in the Iteration No. i , the number of the students who can master the language point No. j is $P_{i,j}$.

$$P_{i,j} = \left(\sum_{k=1}^i p_{k,j-1} - \sum_{k=1}^{i-1} p_{k,j} \right) \times \alpha_i$$

In which $1 \leq i \leq n$, $1 \leq j \leq m$, $0 \leq \alpha_i \leq 1$, $P_{i,0} = \begin{cases} s & i=1 \\ 0 & 2 \leq i \leq n \end{cases}$

In iterative teaching approach, when the language point j is studied for the time No. I , iteration No. i would offer a greatly increased mastery α_i because this language point has already been studied in the iteration No. $i-1$.

Impact factor in iterative teaching approach β is:

$$\beta = \frac{\sum_{i=1}^n P_{1,j}}{\sum_{i=1}^n P_{1,j-1}}$$

$$\text{In which } 1 \leq i \leq n, 1 \leq j \leq m, P_{i,0} = \begin{cases} s & i=1 \\ 0 & 2 \leq i \leq n \end{cases}$$

β is obviously much more gentle.

The iterative teaching approach aims at maximizing $\sum_{j=1}^m \sum_{i=1}^n P_{i,j}$, which is the number of the students who

master the language point. This needs a proper number of iterations so that the value of each α_i can be increased reasonably. The number of iterations is limited by the fixed teaching session, and 3 iterations is proved practical. The following part deals with a teaching case of C language based on 3 iterations.

3. Case Study

The following is an iterative teaching case, in which C language is divided into 3 primary language points: data type and operator, flow control and function, and data organization (array, pointer, structure, and file)^[5].

The teaching object in this case is 800 students. Table I is the mastery of the 3 language points in linear teaching approach. The mastery of the later points is obviously worse.

TABLE I practical mastery in linear teaching approach

Language points No. of students mastered	data type and operator	flow control and function	data organization
800	659	522	443

Table II shows the theoretical mastery of the linear teaching approach, and Table I is consistent with Table II.

TABLE II theoretical mastery in linear teaching approach

Language points No. of students mastered	data type and operator	flow control and function	data organization
800	640	512	410

The iterative idea is applied in C language which is divided into 3 primary language points: data type and operator, flow control and function, and data organization (array, pointer, structure, and file). Three iterations offer three learning periods to consolidate the mastery of each language point. The process is as the following. The first iteration is the perception part which enables the students to comprehend C language as a whole in a very short period so as to arouse their interest in the learning. The second iteration is a detailed explanation which in-depth introduces in sequence the reasons, grammatical rules, and application occasion and application order. The third iteration makes a comprehensive use of all the language points to analyze, design, and realize

the cases of certain difficulty. At the same time this part can also cultivate the students' ability in program designing.

In the 3 iterations, the first and the third one respectively occupies 20% of the total teaching session, and the second, 60%. Table III show the final teaching effects.

TABLE III. practical mastery in 3 iteration teaching approach

Languagepoints No. of students mastered	data type and operator	flow control and function	data organizatio n
800	779	740	722

What is shown in Table III accords with the effects of 3 iteration teaching approach. The mastery rate in the first iteration is 30%, and that in the second is 76%, and the third, 88%, just as Table IV shows.

TABLE IV. theoretical mastery in 3 iteration teaching approach

Language points Times of iteration	data type and operator	flow control and function	data organization
1st iteration	240	72	22
2nd iteration	426	451	381
3rd iteration	118	230	308
total	784	753	711

Compared with the linear teaching approach, the iterative approach can achieve much better results and offer a balanced mastery for each language point. A subject is assigned to each iteration, so an object is achieved in every iteration. Different iterations should be applied in different courses according to the learning conditions of students.

4. Conclusion

Iterative approach offers several periods to each language point, while the previous teaching material is consolidated in the current iteration and a certain increment is achievement at the same time. This approach makes all the periods easier for students because the next period would offer another chance to make it up if something is not well grasped. Similar to the transfer from waterfall model to iterative model, iterative teaching is also a revolution of the traditional teaching approach. It is rather effective for the beginners in computer language learning. Moreover, this approach also provides a new perspective to the research and learning of the readers.

Reasonable times of iteration and the time percentage of each iteration are the two primary factors in improving the teaching effects, and they are the further research object of the author as well.

References

- [1] Cheng, Henry H, "C for the Course", in *Mechanical Engineering*, pp50-52, Vol. 131, Iss 9, September 2009
- [2] Wang Yuanwei, "Study on Experiment Teaching of C Language", *Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering*, New York: IEEE Computer Society, 2009, pp. 3463–3465.
- [3] Lu Min, Jin Xiaoyu, "Analysis of a C Language Teaching Example Based on the ARCS Model: The Basic Concept and Use of Structure," in *Computer Sciences*, pp. 2692-2695, November 2008.
- [4] Pressman, Roger S. *Software Engineering: A Practitioner's Approach*, Sixth Edition, New York: McGraw-Hill, 2004
- [5] Jiang Guangyuan and Tian Limin, *A College Textbook on C Program Designing*, Beijing: Tsinghua University Publishing House, September 2010. (蒋光远, 田琳琳, *C 程序设计快速进阶大学教程*, 北京: 清华大学出版社, 2010 年)