

The Design and Implementation of a Kind of “Data Structure” Visualized CAI Software

Zhao Wang^{1,2,*} and Hui Ma¹

¹Institute of Software, School of EECS, Peking University, Beijing, 100871, China

²Key Laboratory of HCST, MoE, Peking University, Beijing, 100871, China

*Corresponding author

Abstract—Data structure and algorithm is one of the core courses in computer science. The textbook “Algorithm and Data Structure: C language description ” edited by Naixiao Zhang, Professor of Peking University, is one of the national higher education quality teaching materials in China. The software was built according to this textbook, the visualization of the algorithms in the textbook was implemented by the software. The dynamic demonstration of the classic algorithms, automatic code execution, code execution in single step, displaying of key memory data, highlighting the executing code, supporting for user defined data input and other functions were implemented. This software can be executed on Windows platform. It used C# language programming on .Net Framework, Windows Forms (Winforms) and GDI+ graphics library technology to achieve the dynamic demonstration and interactive features.

Keywords-data structure and algorithm; visualization; CAI; C#

I. INTRODUCTION

Data structure and algorithm is a fundamental course of computer science curriculum [1]. It has close contacts with other courses in the areas of computer science, plays an important role as a connecting link between the preceding and the following courses. It includes not only the theoretical knowledge but also the practice of application. This course is distinguished as A and B in Peking University, A is for students major in computer science, and B is for non-computer major science students. The main purpose is to help students understand the concept of algorithms and data structures, improve their ability of using computer technology to solve problems. However, in the teaching of this course, students who aren't major in computer science are relative poor at programming, and it is relative difficult for them to understand the algorithms. Therefore, suitable presentation software which visualize algorithms in the textbook is needed. Data structure and algorithm CAI software is a hot research field.[2] However, many of them involved in the design of courseware, or Flash based animation demonstration method. They can only be used to demonstrate and explain the fixed data, their user interaction are not very good.

In addition, most CAI software in China is adapted to the textbook "Data Structure (C language version)" edited by Prof. Weimin Yan. The textbook “Algorithm and Data Structure: C language description ” edited by Prof. Zhang which is used by many departments in PKU is lack of mature teaching aided software. For many science students majoring in non-computer

departments who have just begun to learn data structure and algorithms, the tiny difference in code can cause confusion, so a set of auxiliary software for Prof. Zhang 's textbook is needed.

II. SOFTWARE DESIGN GOAL

The software is adapted for the textbook edited by Prof. Zhang. It needs to achieve linear list, string, stack and queue, binary tree and tree, graph, dictionary and other data structures, as well as the dynamic presentation of the related algorithms [3].It will use pictures, animation, tables and other forms to demonstrate the process of the classic algorithm in the course, including animation effects and memory data changes, so that students can get intuitive and clear understanding of the course content. So as to achieve the purpose of CAI, improve the quality of teaching.

III. THINKING OF SOFTWARE DESIGN

A. Software Architecture

In order to achieve the purpose of simple and easy to use, all windows of the software are simplified to two types: navigation form and algorithm demonstration form.

Navigation form is the window the user can see after the software is open, it is responsible for the navigation of all the functions. Algorithm demonstration forms are used to display the dynamic running state of the algorithm, they are the cores of the software windows. Navigation form is designed as Figure I. Because the animation effects of different algorithms are not the same, the algorithm demonstration forms have more than one style, classical algorithm demonstration forms are designed as Figure II and Figure III.

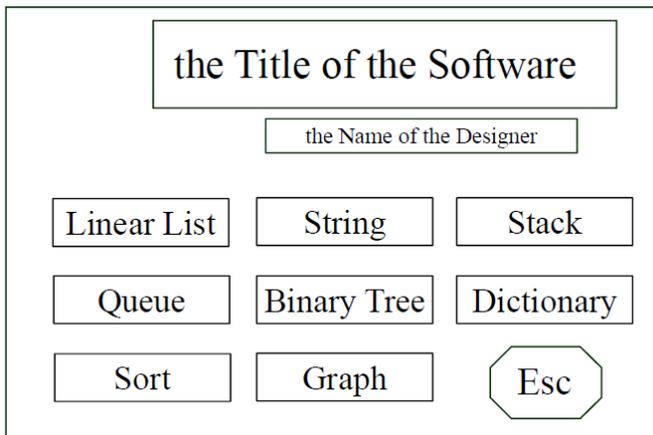


FIGURE I. NAVIGATION FORM DESIGN CHART

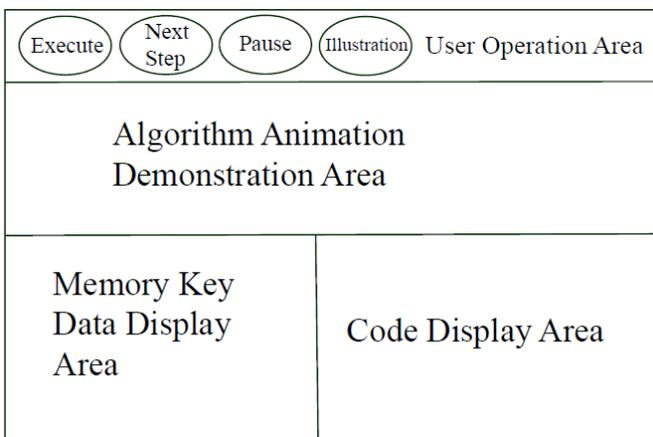


FIGURE II. ALGORITHM DEMONSTRATION FORMS DESIGN CHART 1

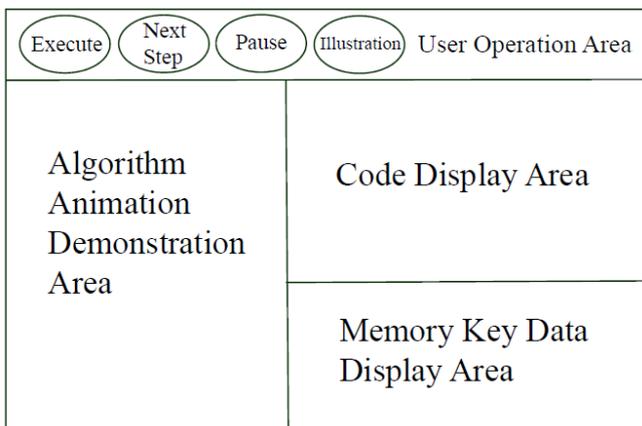


FIGURE III. ALGORITHM DEMONSTRATION FORMS DESIGN CHART 2

B. Thinking of the Software Design

The dynamic demonstration of the algorithm consists of two parts: the operation of the algorithm and the animated demonstration of the result of the algorithm. Because of the great difference between different algorithms, the animated

demonstration has a lot of work, including animation design, implementation, debugging, maintenance, expansion and so on. However, the algorithms belong to the same type have similarities, this point can be used to reduce the amount of code.

MVC software design framework was adopted in the development of software. MVC is an abbreviation for model, view and controller. MVC framework deals with logic, data, interface display independently. Each needs coding respectively. The benefits of doing so are many, for example, when the interface needs to be modified, the logic processing part does not need to rewrite the code [4]. View is the software interface that the user sees. The user interacts with the program through the view. The model is the main part of the application. Program data stored in the model. The view changes with the data in the model. A model does not only have a view, but can be applied to multiple views. Therefore, the MVC framework makes codes with a high degree of reuse. Controller is responsible for receiving the request or data provided by the user. Then, it sends messages to the model, and the model processes them. Finally, the view displays the data according to the model.

The software was written in C# language on .Net Framework platform. The interface of the software needs to use the WinForms technology, the dynamic demonstration of the algorithm needs to use GDI+ graphics technology. WinForms is the name given to a graphical (GUI) class library included as a part of Microsoft .NET Framework [5]. WinForms supports two types of controls, basic controls and autonomous development controls.

IV. KEY TECHNOLOGY OF SOFTWARE DESIGN

A. Establishment of MVC Framework

According to the design idea of MVC framework, in the design of this software, the function of the model module is to store the data structure of the algorithm, the function of the view module is drawing demo interface of the algorithms, and the function of controller module is processing the application logic of the dynamic demonstration of the algorithm. It can be seen that view and controller have a close relationship, because they are related to the dynamic demonstration of the algorithm. So in this software, the view module and controller module are encapsulated in the same class, it can not only reduce the workload of encoding, but also improve the operating efficiency of the software. Therefore, for each demonstration of the algorithm, two classes need to be coded respectively, one is responsible for the data structure, the other is responsible for the presentation interface.

Take the sort algorithm as an example, when the data to be sorted are set up by the user, the system creates an instance of the sorting data class, and then creates an instance of the sorting demonstration form class. When a user executes an instruction, such as "Next step", the interface will handle the received message, at the same time, the data class instance is invoked by the demonstration algorithm, and the result is drawn on the screen by the demonstration form class.

B. The Design and Implementation of the Demonstration Interface

In order to realize the dynamic demonstration of the algorithm, we need to inherit the WinForms class, use its subclasses to implement our custom form.

The algorithm demonstration form is divided into four regions, they are user operation area, code display area, algorithm animation demonstration area, memory key data display area. After creating a custom form class, add these four areas in the form.

The function of the user operation area is to receive the user's instruction, including the "Execute" (automatic execution algorithm), "Next step" (single step implementation algorithm), "Pause" (pause of automatic execution) and "Illustration" (explain the algorithm) four buttons. Buttons use the beautified WinForms Button component. When you add a button, you need to create their message processing function which is responsible for the operation of the button.

The code display area is implemented using WinForms RichTextBox component. RichTextBox can display rich text content, and there are many operating functions. Highlight the executing code is achieved through the Selection function in RichTextBox.

ListView components in WinForms are used to implement memory key data display area. Memory data of different algorithms is not the same, it is need to be set up when the demonstration form is created.

Algorithm animation demonstration area uses a custom my Buffered Panel component. This component inherits from the Win Forms Panel class. Because the Panel does not have very ideal response speed for multi control and multi graphics, animation demonstration can be realized with the custom control.

C. Design and Implementation of Algorithm Demonstration

During the process of the dynamic presentation of the algorithm, it is very important that the animation and code synchronization is needed. In this software, the execution of a sentence is decomposed to achieve the purpose that the execution of the animation and code are synchronous. In this software, the demonstration of the algorithm is divided into four steps: the current line of execution is highlighted, the animation of the current line of execution is displayed, data are updated after execution, jump to the next line that needs to be executed.

In the demonstration form class, there is a member variable of type int to record that the nth statement of the algorithm is executing. When the user clicks "Next step" button, the message response function is triggered, and the following four steps will be executed.

The function of High Light Running Line() is to highlight the current line of execution. The function of Choose Animation Method() is to draw the animation effect of the current line of execution. Udata Data Area () function is to update the data area in memory key data display area. The function of Running Code() is to calculate the sentence to be

executed. In the above four steps, in addition to the drawing of animation effect needs time to execution, the other three functions need very short execution time, so that the user will not notice the time difference between the animation and the code execution, you can achieve the effect of animation and code are synchronous.

```

{
//the current line of execution is highlighted
    High Light Running Line( );
//the animation effect of the current line of execution is draw.
    Choose Animation Method( );
//data are updated after execution
    Udata Data Area( );
// jump to the next sentence that needs to be executed.
    Running Code( );
}

```

This software supports algorithm is executed automatically. Bool-type variable AutoRun is used to mark if the code is automatically executed. When AutoRun is true, the loop calls the four functions mentioned above in order will jump out of the loop until the user suspend the algorithm or the algorithm is finished, AutoRun is then set to false. This is the automatic execution of the demonstration algorithm.

V. MODULE DESIGN AND IMPLEMENTATION

A. Design and Implementation of the Main Interface

The user opens the software to enter the main interface, the main interface needs to explain the function of the software, the design idea is simple, and the overall style of the software is determined. Achieved result is shown as Figure IV.

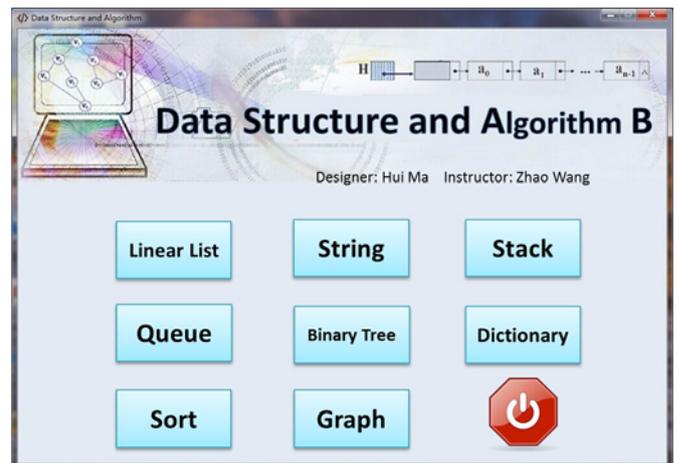


FIGURE IV. REAL NAVIGATION FORM CHART

B. The Design and Implementation of the Algorithm Demonstration Forms

The demonstration forms of different algorithms are not the same, for examples, Conversion between decimal

representation and N-scale notation demonstration form is different from element insert in single linked list in the interface layout, showed as Figure V and VI.

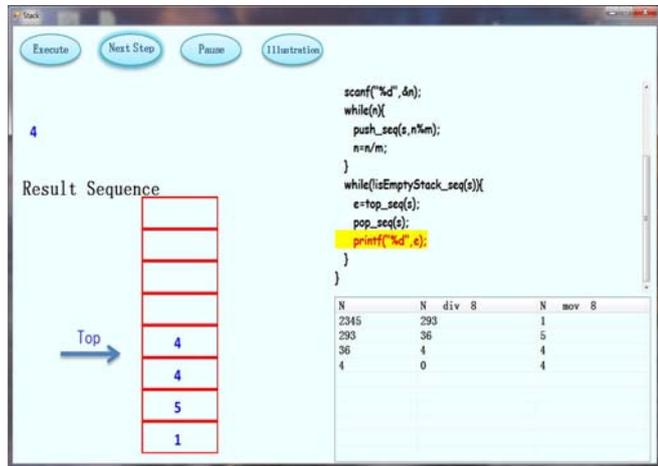


FIGURE V. REAL CHART OF CONVERSION BETWEEN DECIMAL REPRESENTATION AND N-SCALE NOTATION DEMONSTRATION FORM

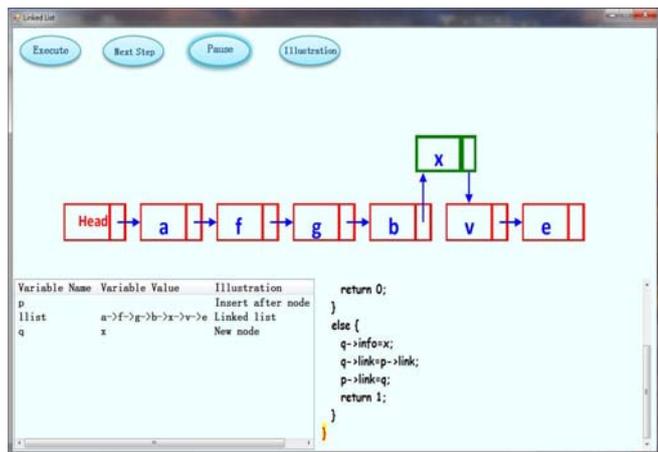


FIGURE VI. REAL CHART OF ELEMENT INSERT IN SINGLE LINKED LIST

C. The Design and Implementation of Functional Modules

At present, the software realized the dynamic demonstration of the following classical algorithms, including:

- Creation of the sequential list, element insertion and deletion in a sequential list; element insertion and deletion in a single linked list; user defined data input is supported;
- Non-backtracking pattern matching, user defined data input is supported;
- Preorder, infix order and post order traversal of the binary tree;
- Binary search algorithm for an ordered dictionary, user defined data input is supported;

- Push and pop operation of the sequential stack; Conversion between decimal representation and N-scale notation (N=2~16), user defined data input is supported;
- The operation of the circular queue , user defined data input is supported;
- Direct insertion sort, direct selection sort, bubble sort, user defined data input is supported;
- Breadth-first traversal, depth-first traversal of undirected graph.

VI. CONCLUSION

C# language, WinForms, GDI+ and other technologies were used to the completion of the “data structure and algorithm B” visualized CAI software. The core function of this software is the realization of the synchronous execution of the algorithm and the animation demonstration. At present, the dynamic demonstration of the 21 kinds of classical algorithms were realized. This software is currently only available for Windows system, but due to the .Net framework supports multiple platforms, it should be convenient to transfer to other platforms in the future.

ACKNOWLEDGMENT

Zhao Wang is the corresponding author of this paper. We would like to thank the anonymous reviewers for their helpful suggestions. This work was sponsored by the NSFC under grant No. 61371131.

REFERENCES

- [1] Yan Yumin, Wu Weimin. Data structure (C language version). Beijing: Tsinghua University press, 2011(in Chinese)
- [2] Sun Haimin. “The analysis of the status of design and development of “data structure” teaching software”. Computer Education, Oct 2005, pp.22~24(in Chinese)
- [3] Zhang Naixiao. Algorithm and data structure: C language description. Beijing: Higher Education Press, 3rd ed, 2014(in Chinese)
- [4] Ren Zhongfang, Zhang Hua, et al. “Overview of the research of MVC model”. Computer Application Research, Oct 2004, pp.4-8(in Chinese)
- [5] Eric Gunnerson, Nick Wienholt. A Programmer's Introduction to C# 5.0. 4th ed, Apress, 2012