Optimization into UI Application Development Using Control Array, Inheritance and Polymorphism in OOP’S

Devendra Gahlot¹, Prof. (Dr.) S.S.Sarangdevot², Asst. Prof. (Dr.) Manish Tanwar³

#Research Scholar, Dept. of Computer Science, Suresh Gyan Vihar University, Jaagtpura, Jaipur, Rajasthan, India. 
#HOD, Institute of Management Studies B.J.S Rampuria Jain College, Bikaner, Rajasthan, India. 
¹devendragahlot@gmail.com 
²imanishtanwar@gmail.com
³sarangdevot@gmail.com

Abstract- A interactive UI can help to solve the great deal of problem. As far as my research is concern is to develop a UI which is academically helpful for related respondents. Like trainer, professionals and students. The purpose of this research is to develop an interactive UI to understand different sorting process and their algorithm. The Copy of a components (Object) have being avoided by the developers in Object Oriented Programming since beginning. Because copying of components create array of related components. But using my concern research, we can develop an interactive UI to understand different sorting algorithm being some Optimization into the same. My Aim is optimization into UI Application using array of component, inheritance and polymorphism which have generally not been used during the development of UI application. Academically, the UI will be so effective & useful to learn sorting algorithm.

Keywords-visual programming, control array, inheritance, polymorphism, performance, sorting algorithms, Interactive UI development.

1. Introduction

As far as research is concern, we are trying to build a user interface. Which will be an integrated environment to show automatic tracing and process of different sorting and their algorithms? This UI will be useful in academics for a number of purposes. First, the UI can be used by students to easily understand the algorithms of related sorting. Second the trainers can make easily understand to respondents about different sorting algorithms and their process using this UI. Third this UI will make easy to understand different sorting algorithms for any one. This UI will help to sustain the tracing and process speed of sorting by the respondents.

To develop this UI we are using that kind of aspects which is not being used earlier for sorting algorithm using control array(by copy of components) . The form inheritance and polymorphic feature of oop’s are also used. As far as research review is concern, Graphical user interfaces (GUIs) simplify use of computers by presenting information in a manner that allows rapid assimilation and manipulation. The use of visual constructs (widgets) that mimic physical objects such as ‘switches’ and ‘buttons’ can speed learning, by providing an intuitive method to provide input to the computer. A GUI is not always an improvement. As demonstrated by some widely used commercial programs, a poor GUI implementation can obscure functionality. If the GUI is organized in a counterintuitive manner, or if the menu contents are arranged haphazardly, or if commonly performed operations require several unexpected steps to be performed, then a user must typically invest a significant amount of time in learning how to use the program before the program can be used effectively. A good GUI design does not require users to memorize the steps needed to perform an action. This is particularly important for scientific applications, where the goal of the user should be to understand the theory behind a program rather than master the arcane steps needed to perform an action. There is one case where memorization may be unavoidable. It is appropriate that a GUI incorporate shortcuts that simplify multi-step tasks; use of these shortcuts may not always be intuitive and expert users may choose to commit them to memory to speed their work. There should also be an obvious way to perform the same tasks, albeit less efficiently, without use of the shortcut. [1]

A. What is Object Oriented Programming

Not all programming languages can be ‘object oriented’. Yet claims have been made to the effect that APL, Ada, Clu, C++, LOOPS, and Smalltalk are object-oriented programming languages. I have heard discussions of object-oriented design in C, Pascal, Modula2, and CHILL. Could there somewhere be proponents of object-oriented Fortran and Cobol programming? I think there must be. “Object-oriented” has in many circles become a high-tech
synonym for “good”, and when you examine discussions in the trade press, you can find arguments that appear to boil down to syllogisms like:

<table>
<thead>
<tr>
<th>Ada is good</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Oriented is good</td>
<td></td>
</tr>
<tr>
<td>Ada is object oriented</td>
<td></td>
</tr>
</tbody>
</table>

“object oriented” ought to mean in the context of a general purpose programming language.[2]

B. Window Programming(GUI)

An application displays windows on the screen that must be managed and coordinated. A window object corresponds to at most one on-screen window. The two principal functions of windows are to provide an area in which views can be placed and to accept and distribute events the user sends through actions with the mouse and keyboard. The term window sometimes refers to the Application Kit object and sometimes to the window server’s window device; which meaning is intended is made clear in context. Panels are a special kind of window, typically serving an auxiliary function in an application, such as utility windows. [16]

The GUIs built are of the simple form-based type where all input and output are shown in a single window. These GUIs are similar to the calculator program that comes with the Windows operating system. All GUIs can run as stand-alone (executable) applications easily installed on other Windows machines. As far as our research is concern, we are using visual basic.

Visual Basic

Visual Basic has previously been used by engineers to create customized software [3_5]. In these studies, the authors explain the increased visual aspect of GUIs and their advantages in engineering. [6]

C. Array

Array programming has two key characteristics:

- Operations can be directly applied to entire arrays of values.
- A set of special functions and operators provides powerful means of data manipulation and allows complex data manipulation processes to be expressed concisely.[8]

D. Control Array

A control array can be created only at design time, and at the very minimum at least one control must belong to it. You create a control array following one of these three methods:

You create a control and then assign a numeric, non-negative value to its Index property; you have thus created a control array with just one element.

You create two controls of the same class and assign them an identical Name property. Visual Basic shows a dialog box warning you that there’s already a control with that name and asks whether you want to create a control array. Click on the Yes button

You select a control on the form, press Ctrl+C to copy it to the clipboard, and then press Ctrl+V to paste a new instance of the control, which has the same Name property as the original one. Visual Basic shows the warning mentioned in the previous bullet.[15]

Control arrays are one of the most interesting features of the Visual Basic environment, and they add a lot of flexibility to your programs:

- Controls that belong to the same control array share the same set of event procedures; this often dramatically reduces the amount of code you have to write to respond to a user's actions.
- You can dynamically add new elements to a control array at run time; in other words, you can effectively create new controls that didn't exist at design time.
- Elements of control arrays consume fewer resources than regular controls and tend to produce smaller executables. Besides, Visual Basic forms can host up to 256 different control names, but a control array counts as one against this number. In other words, control arrays let you effectively overcome this limit.

The importance of using control arrays as a means of dynamically creating new controls at run time is somewhat reduced in Visual Basic 6, which has introduced a new and more powerful capability.[9]

E. Array Programming

Array programming was introduced in the early sixties to ease the description of mathematical processes manipulating arrays [10]. It was initially thought as a simplified way for manipulating array data structures in the language and many implementations are sequential. However, it was also considered as a mean to take advantage of mainframes vectorial processors and exploit data parallelism. The principle is simple: arrays are considered as first-class entities within the language and traditional arithmetic operators (such as addition, etc) are defined natively to operate on arrays or combination of scalar values and arrays (e.g. if X and Y denote arrays of numerical values, X + Y and 2 _X are valid expressions). Array operations are seen as a convenience to avoid writing explicit loops for simple repetitive operations. They reduce the need for control structures use inside the language.[12]

F. Inheritance

Inheritance is a valuable mechanism which enhances reusability and maintainability of software.[18]
Visual Form Inheritance

On some occasions, you may decide that a project calls for a form similar to one that you have created in a previous project. Or you may want to create a basic form with settings such as a watermark or certain control layout that you will then use again within a project, with each iteration containing modifications to the original form template. Form inheritance enables you to create a base form and then inherit from it and make modifications while preserving whatever original settings you need.

How can possible to inherit a Windows Forms Application

The sample application is going to mock up part of the interface of an application. The master form will be frmMasterLayout. Let’s see what the prototype application I am going to build looks like Figure 1 shows three of the forms in the application Which are inherited into frmMasterLayout.

![Figure 1 - Interface](image)

**Bubble Sort Algorithm**

Bubble sort algorithm used in the experiments below was described by C++ language as:

```cpp
template<class Type>
void BubbleSort(Type *R,int n)
{
    int i,j; Type temp;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-i-1;j++)
        {
            if(R[j]>R[j+1])
            {
                temp=R[j];
                R[j]=R[j+1];
                R[j+1]=temp;
            }
        }
    }
}
```

In the best case, the input sequence is positive, the algorithm needs n-1 comparisons, and its time complexity is O(n). In the worst case, the input sequence is negative, the algorithm needs

$$2 \sum_{i=1}^{n} (i-1) = n(n-1)/2$$

comparisons, and its time complexity is O(n^2).

Whatever, the space complexity of bubble sort is O(1).

**Selection Sort Algorithm**

Select sort algorithm [17] used in the experiments below was described by C++ language as:

```cpp
template<class Type>
void SelectSort(Type R[], int n)
{
    int i,j,k;
    for(i=1;i<n;i++)
    {
        k=i;
        for(j=i+1;j<=n;j++)
        {
            if(R[j] < R[k])
            {
                k=j;
                if(k!=i)
                {
                    R[0]=R[i];
                    R[i]=R[k];
                    R[k]=R[0];
                }
            }
        }
    }
}
```

It’s easy to prove: the algorithm needs n (n-1) / 2 comparison operations whatever the input sequence is. That is the algorithm’s time complexity is O(n^2).

Additionally, its space complexity of select sort is O(1).[13]
Insertion Sort Algorithm

Insertion sort algorithm[17] used in the experiments below was described by C language as:

```c
template<class Type>
void InsertSort(Type R[],int left,int right)
{
    Type temp;int i,j;
    for(i=left+1;i<right;i++)
        if(R[i]<R[i-1])
        {
            temp=R[i];j=i-1;
            do{R[j+1]=R[j];j--;}while(j>=left&&temp<R[j]);
            R[j+1]=temp;
        }
}
```

The time complexity of insert sort algorithm described above is O(n^2 ) and its space complexity is O(1) . When the input sequence is positive, the algorithm needs n -1 comparison operations without any move operation. Otherwise, when the input sequence is negative, the algorithm needs (n + 2)(n -1) / 2 comparison operations while (n + 4)(n -1) / 2 move operations needed.[13]

G. Polymorphism

Polymorphism is one of the important concepts of object oriented programming. [19]

Traditional inheritance, polymorphism, and late binding interact nicely to provide both flexibility and safety when a method is invoked on an object via a polymorphic reference; late binding ensures that we get the appropriate implementation of that method for the actual object. [20]

As far as our research is concern, following code used which are polymorphically implemented among all sorting algorithm.

```c
Public i As Byte
Public j As Byte
Public temp As Single
Public flag As Boolean
Public tindex As Byte
Public Sub Delay(s As Integer)
    Dim l As Integer, i As Integer
    For l = 0 To s
        DoEvents
        Next
    End Sub
Public Sub BigDelay(s As Integer)
    Dim l As Integer, i As Integer
    For l = 0 To s
        DoEvents
        For i = 0 To s
            DoEvents
        Next
    Next
    End Sub
```

H. Optimization

Optimization is the art and science of making your program faster, smaller, simpler, less resource hungry, etc. Of course faster often conflicts with simpler and smaller so optimization is a balancing act.[14]

2. Aims and Objectives of Research

WORK

The Object Oriented Programming is very powerful and reliable language to develop a UI Application. If the tools of Object Oriented Programming are optimally used then an interactive and efficient UI Application can be built.

The minimum amount of communication in a sorting algorithm involves moving elements from the locations they start out to where they eventually belong (in the sorted order). An optimal sorting algorithm will communicate every element from its adjacent location to a location in remote memory at most once. Our algorithm is optimal in this sense. For instance, if the input is already sorted, no data movement occurs. However, if the input is in reverse sorted order, almost all elements may need to be communicated to their destinations.

Aim

- As far as my research is concern, my Aim is optimization into UI Application using array of component. Which have generally not been used during the development of UI application?
- With some optimization into UI application using array of object. I want to purpose to develop a UI. Which will show the method of sorting with moving components (Textbox) towards their real position according to their consisting data? And show also related algorithm for related sorting (Bubble sort, Selection sort, Insertion Sort, etc.) with tracing view of problem.
- The user of UI application can control all these things regarding sorting. Like speed of sorting, order of sorting, data to sort etc.
- Nuts and shell, my UI application will be an integrated group of different view of problem solving at same running interface. Like process view of problem solving, tracing view of algorithm of related problem.

Objective

- My first objective is, to optimum use of array of object, which is generally not used by the developer.
- My second objective is optimization into UI application, as Graphical representation to solving the problem of sorting. Because, Graphical User interface has been very popular, interactive and efficient to make easy to interact and understanding the problems.
• To provide a tool, which can play big role to make easy to explain related problem by the mentor and understand by viewer.

3. Methodology/Laboratory Work
• My research belongs to analytical research.
• As far as, our research is concern, it may only be academic oriented so that for the same, we will get information about algorithm of different types of sorting.
• We will adopt a particular Object Oriented Programming language among C++, VB, and Java. Or VB.net, PERL etc.
• Drawing the flow diagram to solve related problem. (Diagrammatical Method)
• Create the array of component (Like Textbox or label) using copy & paste method at same container (Form or frame). Which does the developer generally not use?

4. Conclusion
• My interactive and effective User Interface will simplify complicated task into easy task.
• Effective way to explain sorting and their algorithm for mentor or trainers.
• Effective way to understand sorting and their algorithm for student.
• The real implementation of array of Object in Object Oriented Programming.
• The problems can be solved using other problems being their proper optimization (Like Copy of Textbox1 at same form).
• Control array, inheritance and polymorphism had been play great role during my research work to develop an interactive UI.

References


[19] PENG Jian-sheng, WAN Zhen-wu, PENG Qi-shen (Department of Physics, Hechi University, Yizhou, Guangxi 546300, China; College of Electronic Information Engineering, South-central University of Nationalities, Hubei wuhan 430074, China), Realization of Polymorophism in the Object Oriented Programming on C++ Language, pp1to2, DOI: CNKI:SUN:HCSF.0.20 04-04-018.

[20] ECOOP 2001-Object-Oriented Programming