

ИНФОРМАТИКА ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.023

Разработка технологических мнемосхем на языках высокого уровня

Н. В. Бильфельд¹, М. Н. Фелькер²

Березниковский филиал Пермского национального исследовательского политехнического университета; Россия, 618417, г. Березники, ул. Тельмана, 7

¹bilfeld@mail.ru; +79194910349

²erymarya@yandex.ru; +79638613725

Рассмотрены проблемы при разработке мнемосхем на языках программирования высокого уровня и методы их решения. Приведены примеры процедур масштабирования мнемосхем, а также поиска и отображения требуемого компонента мнемосхемы.

Ключевые слова: мнемосхема; масштаб, Image; технологический процесс; скроллинг и т.д.

DOI: 10.17072/1993-0550-2020-1-65-68

При разработке информационных систем управления технологическими процессами приходится работать с изображениями, размер которых превышает размер стандартного экрана, зачастую приходится сталкиваться с проблемами перемещения по изображению и выбором элементов изображения при переменном масштабе.

К таким изображениям можно отнести изображения мнемосхем технологических процессов [2], пример приведен на рис. 1.

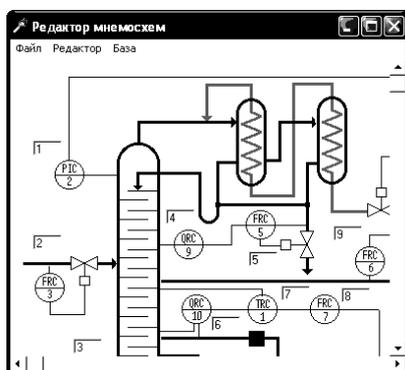


Рис. 1. Пример изображения мнемосхемы

Как правило, к таким мнемосхемам предъявляются следующие требования:

- Возможность вывода изображения в различных масштабах, начиная с минимального масштаба (все изображение помещается на экране монитора), до максимального масштаба (все элементы мнемосхемы выводятся без искажений).
- Возможность скроллинга по изображению, когда его масштаб больше минимального.
- При выборе оборудования, в частности из таблицы базы данных (БД), оно должно выделяться на мнемосхеме (контуром, другим цветом и т.д.) с поддержкой автоматического скроллинга по мнемосхеме [5].
- Возможность выбора конкретного аппарата прибора или исполнительного механизма на мнемосхеме. В частности при выборе регулятора PIC 2 должно открыться окно с данным регулятором, представленным на рис. 2.

При этом мнемосхема должна одинаково реагировать на нажатие кнопок мыши пользователем независимо от масштаба. Такие мнемосхемы, как правило, отображаются на компонентах Image.



Рис. 2. Диалоговое окно регулятора

Необходимо выбирать размер компонентов *Image* таким образом, чтобы он был в кратное число раз меньше размера картинка мнемосхемы [1]. Поэтому, при создании самой мнемосхемы, желательно ориентироваться на размер окна программы, учитывая, что необходимо оставить место для интерфейса. Как правило, ширина окна составляет 950 пикселей, высота 650 пикселей, размер рисунка определяется в зависимости от масштаба по таблице.

Определение размера изображения в зависимости от масштаба

Масштаб	Ширина, пиксели	Высота, пиксели
1:1	950	650
1:2	1900	1300
1:4	3800	2600
1:8	7600	5200

Для удобства и исключения искажений при масштабировании мнемосхемы масштаб по горизонтали и вертикали рекомендуется выбирать одинаковый.

Для управления скроллингом и масштабom мнемосхемы, приведенной на рис. 1, достаточно двух переменных. Одна переменная – *M* – отвечает непосредственно за масштаб,

при создании формы она приравнивается к нулю. Вторая переменная – *MM* – отвечает за выделение на мнемосхеме различных объектов при различных установленных масштабах, при создании формы она приравнивается к отношению размера картинка к размеру компонента *Image*, например:

```
M:=Image1.Picture.Width div
Image1.Width;
```

При этом исходим из того, что установить масштаб больший натурального размера изображения недопустимо.

Ниже приведен код процедуры увеличения масштаба мнемосхемы:

```
Procedure TForm1.Mashtab_Plus;
Begin
  If Image1.Width<Image1.Picture.Width
  Then Begin
    M:=M*2+1;
    MM:=MM div 2;
    Image1.Width:=Image1.Width*2;
    Image1.Height:=Image1.Height*2;
    Image2.Width:=Image2.Width*2;
    Image2.Height:=Image2.Height*2;
    ScrollBar1.Max:=IW*M;
    ScrollBar2.Max:=IH*M;
  End;
End;
```

Процедура уменьшения масштаба мнемосхемы будет выглядеть аналогично:

```
procedure TForm1.Mashtab_Minus;
begin
  If M>0 Then Begin
    M:=(M-1) Div 2;
    MM:=MM * 2;
    Image1.Width:=Image1.Width
  div 2;
    Image1.Height:=Image1.Height
  div 2;
    Image2.Width:=Image2.Width
  div 2;
    Image2.Height:=Image2.Height
  div 2;
    ScrollBar1.Max:=IW*M;
    ScrollBar2.Max:=IH*M;
  End;
End;
```

Для полос скроллинга устанавливаются начальные максимальные значения, равные нулю, так как при загрузке мнемосхемы она будет уменьшена в максимальное количество раз. Как видно из приведенных выше процедур *Mashtab_Plus* и *Mashtab_Minus*, максимальные значения полос скроллинга меняются в зависимости от выбранного масштаба.

Рассмотрим листинг кода для обработки полос скроллинга:

```

Procedure Ustan;
Begin
  Image1.Left := -ScrollBar1.Position;
  Image2.Left := -ScrollBar1.Position;
  Image1.Top := -ScrollBar2.Position;
  Image2.Top := -ScrollBar2.Position;
End;

```

Скроллинг с помощью кнопки мыши происходит по следующему алгоритму:

- Фиксируется нажатие кнопки мыши (свойство *OnMouseDown*).
- Фиксируется перемещение мыши (свойство *OnMouseMove*) и сохраняются ее координаты (*X* и *Y*).
- Фиксируется отпускание кнопки мыши (свойство *OnMouseUp*) и в момент отпускания – положение полос скроллинга (*ScrollBar1.Position*, *ScrollBar2.Position*) приравнивается к значению положения мыши (*X* или *Y*) соответственно. На данном этапе может возникнуть проблема, связанная с тем, что событие *OnMouseDown* уже используется, в частности для выбора компонента на мнемосхеме. Для решения проблемы можно разделить функции, выполняемые при нажатии кнопки мыши, например, для выбора компонента использовать левую кнопку, а для скроллинга – правую или среднюю [3].

Определить, какая кнопка мыши была нажата, можно с помощью процедуры, приведенной ниже:

```

Procedure TForm1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
  Var S:String;
Begin
  Case Button Of
    mbLeft:S:='Левая кнопка';
    mbRight:S:='Правая кнопка';
    mbMiddle:S:='Средняя кнопка';
  End;
End;

```

Для выделения на мнемосхеме каких-либо компонентов на форму необходимо поместить два компонента *Image*, причем саму мнемосхему рекомендуется загрузить в нижний компонент. При этом верхний компонент нужно просто сделать прозрачным, установив свойство *Stretch* в истину. Если сделать наоборот, придется загружать в нижний компонент белый фон, но это не самая главная проблема. Все зависит от того, в каком формате и качестве нарисована мнемосхема, например, форматы *.jpg* не поддерживают прозрачность. Кроме того, многие некачественно нарисованные цветные картинки при установке им прозрачности теряют часть цветов и выглядят при этом искаженными.

Все процедуры, связанные с рисованием и выделением, осуществляются в верхнем компоненте *Image*. Ниже приведена процедура отмены всех выделений:

```

Procedure Clear;
Begin
  Image2.Picture:=Nil;
  Image2.Canvas.Pen.Width:=1;
  Image2.Canvas.Pen.Color:=clWindow;
  Image2.Canvas.Brush.Color:=clWindow;
  Image2.Canvas.Rectangle(0,0,Image2.Width,Image2.height);
End;

```

Чтобы выделить на мнемосхеме какой-либо компонент, координаты прямоугольной области этого компонента должны храниться в таблице БД, при этом координаты следует хранить для натурального размера мнемосхемы.

Выделение компонента должно осуществляться независимо от установленного масштаба, для этого используется масштаб *MM*:

```

Procedure Select;
  Var X1,Y1,X2,Y2:Integer;
  XP1,YP1,XP2,YP2:Integer;
Begin
  {Берем значения координат из таблицы}
  X1:=AdoTable1.Fields[3].Value;
  Y1:=AdoTable1.Fields[4].Value;
  X2:=AdoTable1.Fields[5].Value;
  Y2:=AdoTable1.Fields[6].Value;
  Clear; {Очистка предыдущего
  выделения}
End;

```

```

XP1:=Round(X1 div
MM);YP1:=Round(Y1 div MM);
XP2:=Round(X2 div MM);
YP2:=Round(Y2 div MM);
Im-
age2.Canvas.Rectangle(XP1,YP1,XP2,YP2);
{Перемещение к найденному
объекту}
ScrollBar1.Position:=Round(X1/MM)-
Panel1.Width div 2;
ScrollBar2.Position:=Round(Y1/MM)-
Panel1.Height div 2;
Ustan;
End;

```

С другой стороны, при нажатии кнопки мыши на каком-либо компоненте мнемосхемы его необходимо отыскать в БД, например, чтобы отобразить его полное наименование или открыть дополнительную форму [4]. Рассмотрим процедуру поиска и отображения компонента мнемосхемы:

```

Procedure Find;
Begin
{Формируем запрос}
S:='Where P3<'+IntToStr(X*MM)+'
and P5>'+IntToStr(X*MM)+' and P4<'+
IntTo-
Str(Y*MM)+' and P6>'+IntToStr(Y*MM);
AdoQuery1.SQL[2]:=S;
AdoQuery1.Active:=True;
{Находим в таблице и выводим ин-
формацию}
If AdoQuery1.RecordCount=1 Then
Begin
S:=AdoQuery1.Fields[1].Value;
AdoTa-
ble1.Locate('P1',S,[]);
Ed-
it5.Text:=AdoQuery1.Fields[2].Value+'
'+AdoQuery1.Fields[1].Value;

```

```

End
Else Edit5.Text:='Объект не выбран';
End;

```

При этом опять же используется масштаб *MM*, так как поиск должен работать независимо от установленного масштаба мнемосхемы. Рассмотренные подходы успешно использовались при разработке модели управления процессом грануляции на ПАО "Уралкалий" [7].

Список литературы

1. Бобровский С. Учебный курс Delphi 7 // М.: Питер, 2006. 712 с.
2. Бильфельд Н.В. Моделирование систем с использованием блока чистого запаздывания // Молодой ученый. 2013. № 6. С. 25–30.
3. Бильфельд Н.В., Затонский А.В. Применение самоорганизующихся систем при управлении сложными процессами // Проблемы теории и практики управления. 2007. № 12. С. 70–74.
4. Беккер В.Ф., Плехов П.В., Затонский А.В. Управление средствами производства в системе менеджмента качества химической продукции // Горный информационно-аналитический бюллетень (науч.-техн. журн.). 2010. № 9. С. 66–72.
5. Бильфельд Н.В. Использование команд MATLAB – от версии к версии // Молодой ученый. 2014. № 2. С. 83–84.
6. Затонский А.В. Программные средства глобальной оптимизации систем автоматического регулирования. М.: Инфра-М: ИЦ РИОР, 2013. 136 с.
7. Затонский А.В. Моделирование технологического участка обогатительной фабрики в пакете Matlab // Обогащение руд. 2014. № 4 (352). С. 49–54.

Development of technological mnemonic schemes in high-level programming languages

N. V. Bilfeld¹, M. N. Felker²

¹Berezniki branch of Perm National Research Polytechnic University
7, Telmana st., Berezniki, 618417, Russia; bilfeld@mail.ru; +79194910349

²Berezniki branch of Perm National Research Polytechnic University
7, Telmana st., Berezniki, 618417, Russia; erymarya@yandex.ru; +79638613725

The paper deals with the problems in the development of mnemonic schemes in high-level programming languages and discusses methods of their solution. There are provided examples of the procedures for mnemonic schemes scaling, search and display of the desired component.

Keywords: *mnemonic scheme; scale; Image; technological process; scrolling.*