Original Research Article:
full paper

(2020), «EUREKA: Physics and Engineering»
Number 1

DEVELOPMENT OF INFORMATION VISUALIZATION METHODS FOR USE IN MULTIMEDIA APPLICATIONS

Yevhen Hrabovskyi
Department of Computer Systems and Technologies[1]
maxmin903@gmail.com

Natalia Brynza
Department of Informatics and Computer Engineering[1]
natalia.brynza@hneu.net

Olga Vilkhivska
Department of Informatics and Computer Engineering[1]
grom_o@i.ua

[1]*Simon Kuznets Kharkiv National University of Economics*
9-a Nauky ave., Kharkiv, Ukraine, 61166

Abstract

The aim of the article is development of a technique for visualizing information for use in multimedia applications. In this study, to visualize information, it is proposed first to compile a list of key terms of the subject area and create data tables. Based on the structuring of fragments of the subject area, a visual display of key terms in the form of pictograms, a visual display of key terms in the form of images, and a visual display of data tables are performed. The types of visual structures that should be used to visualize information for further use in multimedia applications are considered. The analysis of existing visual structures in desktop publishing systems and word processors is performed.

To build a mechanism for visualizing information about the task as a presentation, a multimedia application is developed using Microsoft Visual Studio software, the C# programming language by using the Windows Forms application programming interface. An algorithm is proposed for separating pieces of information text that have key terms. Tabular data was visualized using the “parametric ruler” metaphorical visualization method, based on the metaphor of a slide rule.

The use of the parametric ruler method on the example of data visualization for the font design of children's publications is proposed. Interaction of using the method is ensured due to the fact that the user will enter the size of the size that interests for it and will see the ratio of the values of other parameters. The practical result of the work is the creation of a multimedia application “Visualization of Publishing Standards” for the visualization of information for the font design of publications for children. The result of the software implementation is the finished multimedia applications, which, according to the standardization visualization technique in terms of prepress preparation of publications, is the final product of the third stage of the presentation of the visual form.

Keywords: information visualization, multimedia application, visual display, key terms.

DOI: 10.21303/2461-4262.2020.001103

1. Introduction

In the process of development of modern society, the computer era has begun – the era of information and informatization. Hundreds of terabytes of information appear annually in local and global networks. Mechanisms are being introduced to search for the necessary information, however, these tools are effective when users have a specific goal and understand what information is stored. In other cases, methods for visualizing information can help the user.

The main objective of visualization is the use of human visual perception of information to enhance its cognitive abilities. The human visual system is capable of quickly processing visual signals, and advanced information technology has turned the computer into a powerful tool for managing digital information. Visualization is a bridge connecting the human visual system and the information system, helping to identify images, hypothesize and extract ideas from huge data sets, and contributes to scientific research and forecasting. The main purpose of visualization is the use of human visual perception of information to enhance its cognitive abilities.

To date, there are a large number of techniques that are devoted to the visualization of various information [1–5]. In a variety of information it is possible to understand objects that have a physical nature (scientific visualization), program codes, software (software visualization), etc.

According to various methods of visualizing information in specialized literature [6–10], mechanisms for controlling the quality of content and optimizing the interface of multimedia applications are proposed.

The most effective application of information visualization in various multimedia applications will facilitate the perception of a large amount of text information and facilitate its visual representation in multimedia systems.

2. Materials and Methods

The research methods are as follows. Analysis and synthesis are used at all stages of the study. Induction and deduction methods are used to study the theoretical foundations of the design of multimedia applications and determine the subject area of information visualization. Logical analysis is applied to study the principles of information visualization existing in theory and practice in the process of designing multimedia applications.

3. Experimental procedures

The first stage of the information visualization technique for use in multimedia applications should consist of two steps:

1) formation of a list of key terms of the subject area;

2) formation of data tables.

Step 1. 1. Formation of a list of key terms.

Raw data can be presented in any form, starting with spreadsheets and ending with unstructured texts. Key terms will be used as raw data to be used in the process of visualization of publishing standards.

Step 1. 2. Formation of data tables.

The second stage of the standardization visualization technique consists of the following steps:

1) visual display of key terms in the form of pictograms;

2) visual display of key terms in the form of images;

3) visual display of data tables.

Visual structures are images that fully convey the content of the keyword. To visualize information with a view to further use in multimedia applications, it is necessary to apply three types of visual structures:

1) the first type is pictograms that schematically depict the content of the key term;

2) the second type is an image, which is an example of the use of key terms in practice;

3) the third type is the interactive infographics of parameter ruler, with the help of which tabular data is presented.

It is important to choose the most expressive visual structure that reflects all the data without loss, which can easily and quickly be interpreted by a person with the least number of errors and to the maximum transfers all the differences in the data. The visual structures of the first type are developed in accordance with existing examples of the use of key terms of a multimedia publishing house. The second type of visual structures is also developed independently, but some structures have already been developed and are used in desktop multimedia publishing systems. The third type of visual structures is the parameter ruler interactive infographic, which refers to one of the visualization methods, namely metaphor visualization, with which the data will be visualized.

Programs of the electronic publishing house QuarkXPress, Adobe InDesign, Adobe PageMaker have already developed visual structures regarding key terms in the field of prepress preparation of publications, in particular in terms of general concepts, design and execution of the publication page and printing performance of the publication. Visual structures refer to pictograms that are used as interface elements of desktop publishing systems – these are buttons that convey the contents of a key term in the field of multimedia publishing. Visual structures that convey the content of key terms in the field of prepress publications, in particular, are used not only in desktop publishing systems, but also in a regular text editor such as Microsoft Word. A set of pictograms is formed on the basis of pictograms distributed in desktop publishing systems and word processors (**Table 1**).

Table 1

Analysis of existing visual structures in desktop publishing and word processors

Key term	Desktop publishing & word processor			
	Adobe PageMaker	**Adobe InDesign**	**QuarkXPress**	**Microsoft Word**
Edition page	🗋	🗋	🗋	🗋
Font group				
Indent		☰	☰	Absent
Initial	Absent	A≡	A≡	A≡
Illustration	☒	☒	🖼	
Table	▦	▦	▦	
Text	T	T	T	T
Mark	N**B***I*	Absent	**B***I*	**Ж***К*Ч
Fields	Absent	Absent	Absent	▯
Publication format	Absent	Absent	Absent	🗋

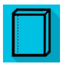
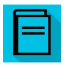
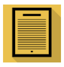
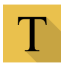
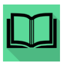
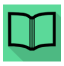
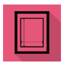
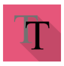
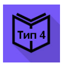
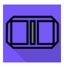
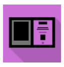
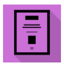
The analysis of existing visual structures will help to develop all other structures that have no analogues in desktop publishing. The visual structures that were discovered during the analysis, some will be used without changes, some will be amended. The selected and developed pictograms have a different display style, some flat and some voluminous, in order to present them in a single style, a "flat design" was chosen.

"Flat" design – a minimalistic approach to the design of objects, which emphasizes ease of use, it is more focused on the end user.

As a result of applying a "flat" design with respect to the pictograms, the following results were obtained (**Table 2**).

Table 2

Pictograms in flat design style

Edition block	**Edition**	**Main page**	**Text**
			
Complex notebook	**Notebook**	**Borderless edition page format**	**Font group**
			
Compiled cover	**Base**	**Frontispiece**	**Title**
 			

The third stage of the methodology consists of 3 steps:

1) development of a general concept of presentation;

2) development of the constituent elements of the presentation;

3) implementation of navigational interaction of presentation elements.

Step 3. 1. Determine in what form the data tables and visual structures will be reflected.

Relative to the task, a multimedia application based on Microsoft Visual Studio software, the C# programming language, using the Windows Forms application programming interface is being developed as a presentation.

Step 3. 2. Implementation of navigational interaction of presentation elements.

Navigation through the multimedia application for the visualization of information should be performed according to the following scheme (**Fig. 1**).

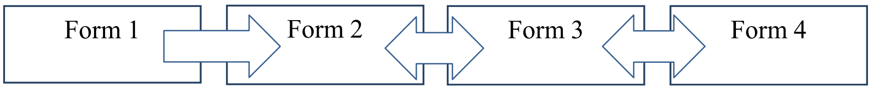


Fig. 1. Navigation through the multimedia application for the visualization of information

The navigation of elements of a visual form is implemented directly by writing code for each form and each of its elements.

Presentation of a visual form interactively changes and complements visual structures in order to get visualization from a static presentation by setting graphic parameters – to get a representation of the visual structure. The multimedia application developed using the C # programming language using XML (the standard for constructing markup languages for hierarchically structured data for exchanging between different applications) acts as a visual form.

4. Results

The formation of data tables takes place in two stages.

The first stage is the formation of the data Table "terms – determination".

The second stage is the formation of the data Table "term – a piece of information" based on the use of a piece of information, which is visualized.

The formation of the data Table "terms – fragments of information" consists in isolated fragments of the text of the source information containing the key term (**Fig. 2**).

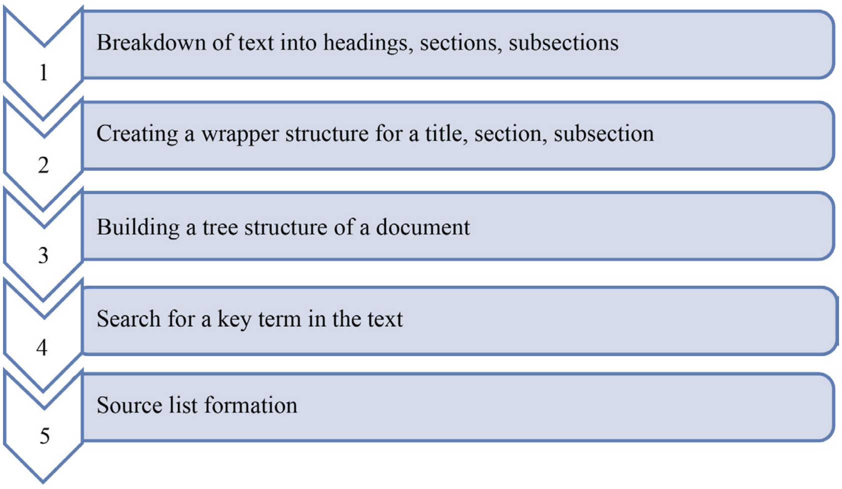

1
Breakdown of text into headings, sections, subsections
2
Creating a wrapper structure for a title, section, subsection
3
Building a tree structure of a document
4
Search for a key term in the text
5
Source list formation

Fig. 2. The algorithm for separating text fragments of information that have key terms

Relative to each key term, visual structures are developed, and data tables are also supplemented – these are pictograms and images.

It is proposed to use interactive infographics to create visualized tabular data.

To visualize tabular data in publishing standards, it is proposed to select a method from the "metaphorical visualization" group. Visual metaphors have a dual function: firstly, they present information in graphical form in order to organize and structure it, and secondly, they convey an idea of the information presented through the key characteristics of the metaphor that is used. Metaphors are useful because they allow to explain an abstract concept in specific conditions, which makes it easier to understand. They can also cause an emotional reaction.

Tabular data can be visualized using the metaphorical visualization method "parametric ruler". This interactive method is based on a metaphor of a slide rule that allows users to move the scales to analyze various combinations of parameter values. "Parametric ruler" refers to interactive infographics, because the user has the opportunity to interact with the information display system and observe its reaction.

Alternative methods for solving the tasks are indexing and using classification and clustering algorithms.

Indexing is necessary to extract some attributes of visualization objects; they convey the meaning and content of the objects themselves. It is necessary the different indexing algorithms depending on the nature of the data: for example, natural language processing algorithms (dictionaries of keywords, key phrases, group of names, parts of speech) for collections of texts; image processing algorithms (segmentation by color, brightness, structure) for image collections; audio processing algorithms (by sound and pitch) for collections of audio files; video processing algorithms (scene segmentation) for video collections.

Classification algorithms distribute visualization objects into certain categories (groups) using machine learning algorithms, for example, Bayes algorithm, k-nearest neighbor algorithm, neural network algorithms, etc.

Clustering algorithms dynamically divide visualization objects into groups by calculating some degree of similarity between them, for example, the k-average algorithm, Kohonen self-organizing maps, hierarchical algorithms, etc. As a result, matrices are obtained that describe objects and the relationships between them and their groups.

However, the methods of indexing and using classification and clustering algorithms do not provide opportunities to facilitate the perception of a large amount of textual information and to ensure its presentation in multimedia systems.

Let's consider the use of the metaphorical visualization method "parametric ruler" on the example of data visualization for the font design of publications for children.

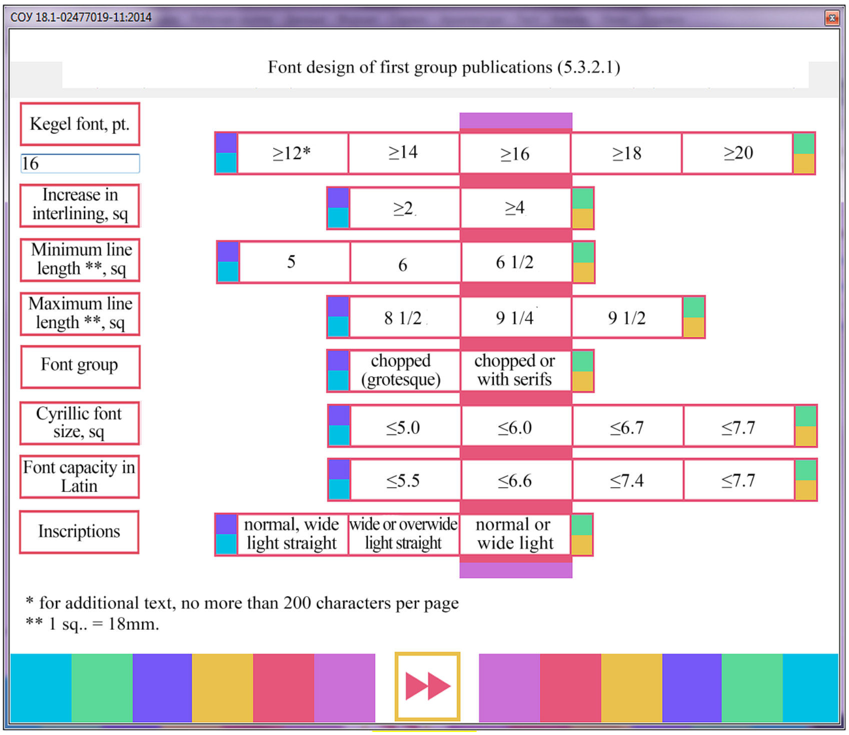
The parameters will include: font size, leading increase, minimum line length, maximum line length, font group, font capacity on the Cyrillic graphic basis, font capacity on the Latin graphic basis, styles. Horizontal scales with possible parameter values, sorted from small to large, should move relative to each other. Interactivity is realized due to the fact that the user will enter the size of the size that interests for it and will see the ratio of the values of other parameters (**Table 3, Fig. 3**).

Table 3

Font design of publications for children

Font size, at least, points	Leading increase, not less, points	Line length				Font group	Font capacity, no more sign/sq.	Mark
		minimal		maximal				
		squares	mm	squares	mm			
20 and more	2	6 1/2	117	9 1/2	171	Sans-serif (grotesque) or with serif	5,0 (5,5)*	Normal or wide bright direct
16–18	4	6 1/2	117	9 1/4	167	Sans-serif (grotesque) or with serif	6,0 (6,6)*	Normal or wide bright direct
14	4	6	108	8 1/2	153	Sans-serif (grotesque)	6,7 (7,4)*	Normal or extra wide bright direct
12**	2	5	90	8 1/2	153	Sans-serif (grotesque)	7,7	Normal, wide or extra wide bright direct

*Note: * – for a font on a Latin graphic basis; ** – for additional text that is over 200 characters per page*



СОУ 18.1-02477019-11:2014

Font design of first group publications (5.3.2.1)

Kegel font, pt. 16	≥12*	≥14	≥16	≥18	≥20
Increase in interlining, sq		≥2	≥4		
Minimum line length **, sq	5	6	6 1/2		
Maximum line length **, sq		8 1/2	9 1/4	9 1/2	
Font group		chopped (grotesque)	chopped or with serifs		
Cyrillic font size, sq		≤5.0	≤6.0	≤6.7	≤7.7
Font capacity in Latin		≤5.5	≤6.6	≤7.4	≤7.7
Inscriptions	normal, wide light straight	wide or overwide light straight	normal or wide light		

* for additional text, no more than 200 characters per page
** 1 sq.. = 18mm.

Fig. 3. The Table and the result of its visualization

Tables visualized in this way will help simplify the process of covering a large amount of information, as well as reduce the cost of finding the necessary data and consolidate them at the visual level.

The existing data Table "term – definition – standard fragment – visual structures" has the following data set: key term, definition, standard fragments, and visual structures. The grouped data of the table, namely key terms, their definitions and visual structures will be presented in the form of a kind of database, namely in the form of an xml-file (keywords.xml) (**Fig. 4**).

```
<?xml version="1.0" encoding="utf-8" ?>
<keywords>

 <chapter name="...">
  <keyword name="..." image="..." icon="...">...</keyword>
 </chapter>

</keywords>
```

Fig. 4. The structure of the xml file that contains the elements of the data Table "term – definition – visual structures"

The structure of the xml file consists of tags. In principle, xml is a database with a tree structure (**Fig. 5**).

The root tag <keywords> ... </ keywords>, it lists the entire list of grouped key terms. The <chapter> ... </ chapter> tag represents each group of key terms (General concepts, Design and implementation of a publication's page, Design and implementation of a publication's block, Design and implementation of a publication's cover, Target design of a publication, and Polygraphic execution of a publication). This tag has the attribute name="...", in which it is necessary to declare the name of the group of key terms in quotation marks. The <keyword> ... </ keyword> tag is responsible for the safety of elements from the data Table "term - definition – visual structures", namely the key term, definition, visual structure of the first and second type. This tag has three attributes name="...", image="...", icon="...". The attribute name="..." is responsible for the preservation of the key term, the name of which must be placed in quotation marks after the declaration of the attribute. The attribute image="..." is responsible for the preservation of the visual structure of the first type, in the blots of this attribute, it is necessary to specify the path of the corresponding image, the attribute icon="..." is responsible for the preservation of the visual structure of the second type, in the blots of this attribute it is necessary to register the path of the corresponding icon. The term definition is placed between the <keyword> ... </ keyword> tags.

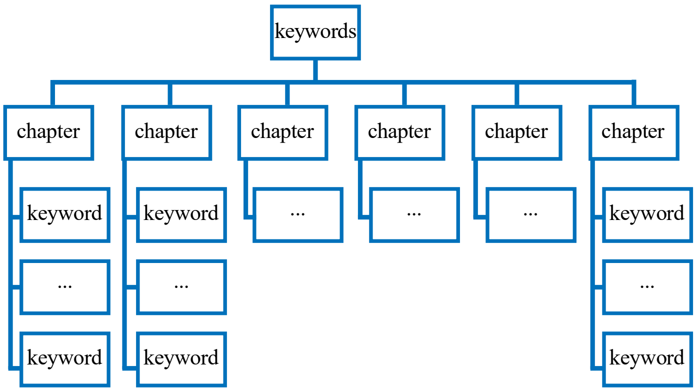


Fig. 5. The tree structure of the xml file that contains the elements of the data Table "term – definition – information fragment – visual structures"

The existing structure of the xml-file is filled with elements from the data Table "term – definition – visual structures" (keywords.xml) (**Fig. 6**).

```
<keywords>
        <chapter name=" General concepts ">
        <keyword     name="Edition     block"     image="images\zagal1.jpg"
icon="images\z2.png">
        a set of printed sheets or notebooks, enclosed at the root, containing all
pages of a future edition.
        </keyword>
        <keyword          name="Edition"          image="images\zagal.jpg"
icon="images\z.png">
        a printed product intended for the dissemination of information contained
therein, has undergone editorial and publishing work, is self-printed and has
original information.
        </keyword>
        <keyword          name="Cover"          image="images\zagal2.jpg"
icon="images\z3.png">
        the outer cover of the edition that connects to the unit without the
flywheels.
        </keyword>
        <keyword      name="Bookbinding"      image="images\zagal3.jpg"
icon="images\z4.png">
        the outer covering of the product that connects to the block with or without
the two flywheels and the core material.
        </keyword>
     </chapter>
 </keywords>
```

Fig. 6. The xml file filled with elements of the data Table "term – definition – visual structures"

The xml-file filled with elements of the data Table is responsible for the safety of the data that will be used in the future developed by the multi-dairy edition.

5. Discussion

Microsoft Visual Studio, the C# programming language using Windows Forms, was chosen as a tool for developing an electronic publication.

The practical result of this work is the creation of a multimedia application “Visualization of Publishing Standards” for the visualization of information for the font design of publications for children. The design of this multimedia application was developed according to new trends in the field of design – the “flat” design (flat design), as evidenced by the colors and navigation elements used in the multimedia application.

Basic principles of Flat design:

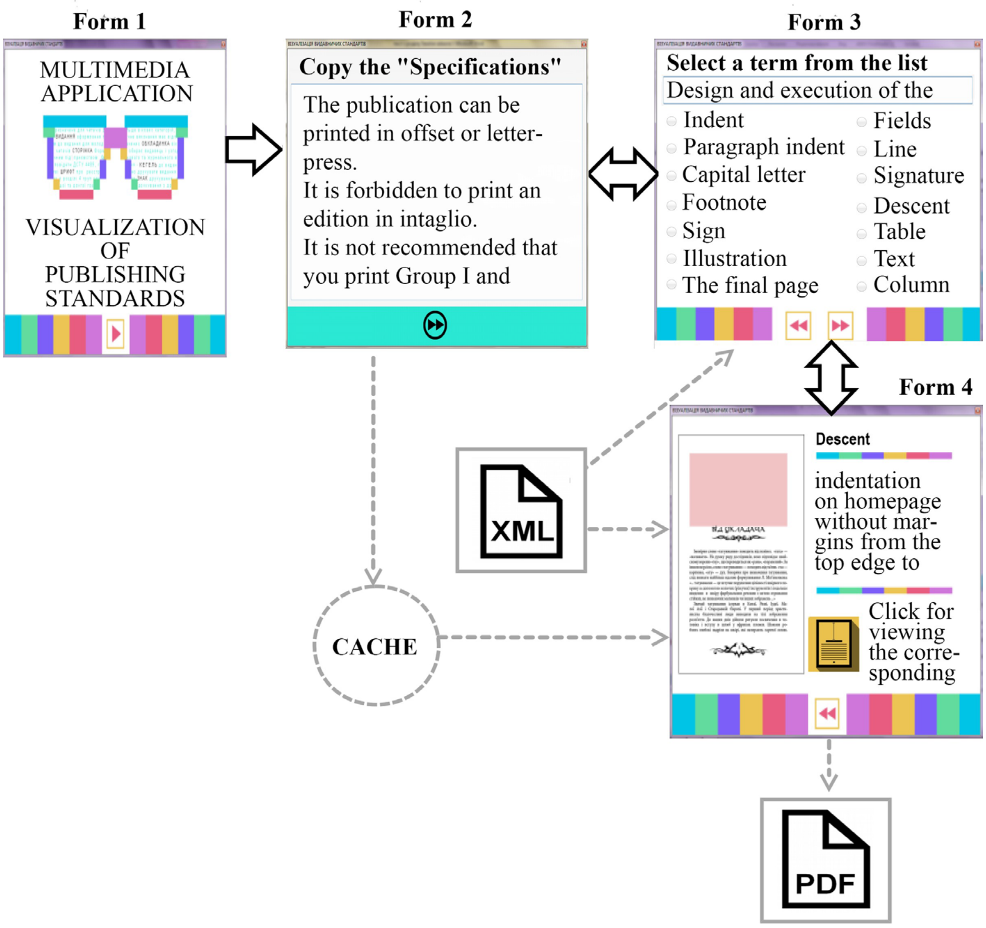
1) lack of unnecessary effects;

2) simplicity of the elements;

3) accurate work with fonts. The use of simple elements increases the importance of typography in design. Working with fonts should be done very carefully. The nature of the font should complement, and not contradict, the general design scheme. A font in a “flat” design is an important navigation element. The simplicity of the elements does not mean that complex fonts cannot be used. Simply, everything should be sustained in the style of minimalism;

4) minimalism. In a “flat” design, it is possible to avoid unnecessary “bells and whistles”, complex approaches to the visualization of elements.

The structure of the application “Visualization of Publishing Standards” has four component forms, each of which has corresponding elements (which perform the corresponding navigation and interactive tasks) and performs functional tasks. The general concept of the multimedia application is presented in **Fig. 7**.


Form 1
MULTIMEDIA APPLICATION
VISUALIZATION OF PUBLISHING STANDARDS
Form 2
Copy the "Specifications"
Form 3
Select a term from the list
Form 4
Descent
XML
CACHE
PDF

The publication can be printed in offset or letter-press.
It is forbidden to print an edition in intaglio.
It is not recommended that you print Group I and

Design and execution of the

○ Indent ○ Fields
○ Paragraph indent ○ Line
○ Capital letter ○ Signature
○ Footnote ○ Descent
○ Sign ○ Table
○ Illustration ○ Text
○ The final page ○ Column

indentation on homepage without margins from the top edge to

Click for viewing the corresponding

Fig. 7. The work scheme of the “Visualization of publishing standards” multimedia application

The software implementation of a multimedia application begins with the creation of a new project in the Microsoft Visual Studio environment, namely the WindowsFormsApplication project, which will allow to begin creating windows (form) of the future multimedia application. At the beginning of the creation of forms, let's determine the size of the future publication and the ability to change the size of the form. In order to set the sizes of future forms, the application needs to specify the sizes in pixels in the properties of the forms in the Size section: 900; 754. The future multimedia application has fixed form sizes, therefore, in the form properties in the FormBorderStyle section, specify the value FixedToolWindow. It is also necessary to enter a name for the future publication and fix the icon, in the Text section let's mark "Visualization of information", and in the Icon section let's define an icon for the future multimedia application. For each form, a developed background image is loaded using the section in the properties of the BackgroundImage forms.

Form 1 of the application has only one element – this is the navigation element – the button (button1), which is responsible for switching to relatively (**Fig. 8**).

This code is in the class Form1.cs.

```
private void button1_Click(object sender, EventArgs e)
    {
      // At the click of a button go to form 2
      var form2 = new Form2();
      form2.Show();
      this.Hide();
    }
```

Fig. 8. The program code of the button, which makes the transition to Form 2

Form 2 has three elements: the first label1 – does not have any navigation and interactive functions, it is just an inscription, informs the user about what needs to be done in this form, the second element is textBox1, the field into which it is necessary to copy a fragment of the standard text (section "Technical requirements") and the third element is button1, which is responsible for saving the text that was entered into textBox1 by activating the corresponding class, as well as for switching to the next form (**Fig. 9**).

This code is in the class Form2.cs.

```
private void button1_Click(object sender, EventArgs e)
    {
     // Memorizes the entered text
     Program.State.Text = textBox1.Text;

     // Press the button to go to form 3
     var form3 = new Form3();
     form3.Show();
     this.Hide();
    }
```

Fig. 9. The program code of the button, performs the operation of saving text and transition to Form 3

The code that declares the State class, which is responsible for saving the entered text, is in the Program.cs class (**Fig. 10**).

```
// A global object for storing information
public static State State { get; private set; }
```

Fig. 10. State class declaration

The description of the State class is found in State.cs (**Fig. 11**).

```
using System.Collections.Generic;
using System.Linq;
using System.Xml.Linq;

namespace PublicationStandartsVisualization
{
   ///<summary>
   /// A class that stores the state of an application at runtime
   /// It also has a database of key terms
  ///<summary>
     public class State
     {
       public string Text { get; set; }
       public Chapter SelectedChapter { get; set; }
       public Keyword SelectedKeyword { get; set; }
       public IEnumerable<Chapter> Chapters { get; set; }
       public static State Load()
    {
      // Creating an empty status object
        var result = new State();
        try
        {
          // Open keywords file
          var keywordsDocument = XDocument.Load("keywords.xml");

          // Handling groups of key terms for the drop-down list
                result.Chapters = keywordsDocument
          .Descendants("chapter")
          .Select(c => new Chapter
                         {
                           Name = c.Attribute("name").Value,
                         Keywords = c.Elements("keyword")
                                .Select(ParseKeyword).ToArray()
                          }).ToArray();
        }
       catch
       {
        }
return result;
}
// Processing each keyword from a configuration file
private static Keyword ParseKeyword(XElement el)
{
          var imageAttr = el.Attribute("image");
          var iconAttr = el.Attribute("icon");
          return new Keyword
                   {
                   Name = el.Attribute("name").Value,
                   Image = imageAttr != null ? imageAttr.Value : null,
                     Icon = iconAttr != null ? iconAttr.Value : null,
                     Description = el.Value
                   };
          }
     }
}
```

Fig. 11. State class program code

Form 3 has five elements: the first label1 does not have any navigation and interactive functions, it is just an inscription that informs the user about what needs to be done in this form, the second comboBox1 element is a drop-down list of terms group names, the third flowLayoutPanel1 element reflects the list key terms using the RadioButton elements, the fourth element of button1 is the transition to Form 2, the fifth element of button2 is the transition to Form 3.

The second and third elements are related to each other, because when to select one of the group names options presented in comboBox1, the set of key terms that are part of the flowLayoutPanel1 element will change (**Fig. 12**).

This code is in the class Form3.cs.

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
      {
         // Check whether events can be processed from the drop-down list
         if (!isInitialized) return;

         // Cleaning the panel with RadioButton
         flowLayoutPanel1.Controls.Clear();

         // Memorizes the selected item from the list
         var chapter = comboBox1.SelectedItem as Chapter;
         if (chapter == null) return;
         Program.State.SelectedChapter = chapter;

         // Fill in the RadioButton panel from the key terms associated with the drop
down box
         bool hasSelection = false;
         foreach (var k in chapter.Keywords)
         {
            // Creating a RadioButton with a key term name
var rb = new RadioButton {Text = k.Name, Margin = new Padding(5), Tag = k,
Width = 300, Height=28};
            // If this is the last selected key term (the user has returned to form 4) mark
it as selected
            if (Program.State.SelectedKeyword == k)
            {
               hasSelection = true;
               rb.Checked = true;
            }
            flowLayoutPanel1.Controls.Add(rb);
          }
          // If no key terms are selected, select the first term in the list
          if (!hasSelection)
          {
          (flowLayoutPanel1.Controls[0] as RadioButton).Checked = true;
          }
        }
```

Fig. 12. The program code of the comboBox1 (flowLayoutPanel1) elements

Buttons button1 and button2 navigate – go to Form 2 and Form 4 (the code is also in the file Form3.cs) (**Fig. 13**).

```
private void button1_click( object senter, EventArgs e)
          {
                // click buttons to go to form 2
                var form2 = new Form2();
                form2.Show();
                this.Hide();
          }
          private void button1_click( object senter, EventArgs e)
          }
                // searching for RadioButton favorites and memorizing the associated
                key term
                foreach (var rb in
flowLayotPanel1.Controls. 0fType< RadioButton>())
                      if (rb.Checked)
                      {
                           Program.State.SelectedKeyword = rb. Tag as Keyword;
                      }
                 }
                 // click buttons to go to form 4
                var form4 = new Form4();
                form4.Show();
                this.Hide();
```

Fig. 13. Program code for button1 and button2

Form 4 has six elements: label 1 – display of the key term, piturebox – reflection of the first type of visual structures, richtextbox – reflection of the definition of the key term, button1 – transition to Form 3, button 2 – opening a pdf file with a visual structure of the second type, label 2 – an inscription corresponding to the task of Button 2 (**Fig. 14**).

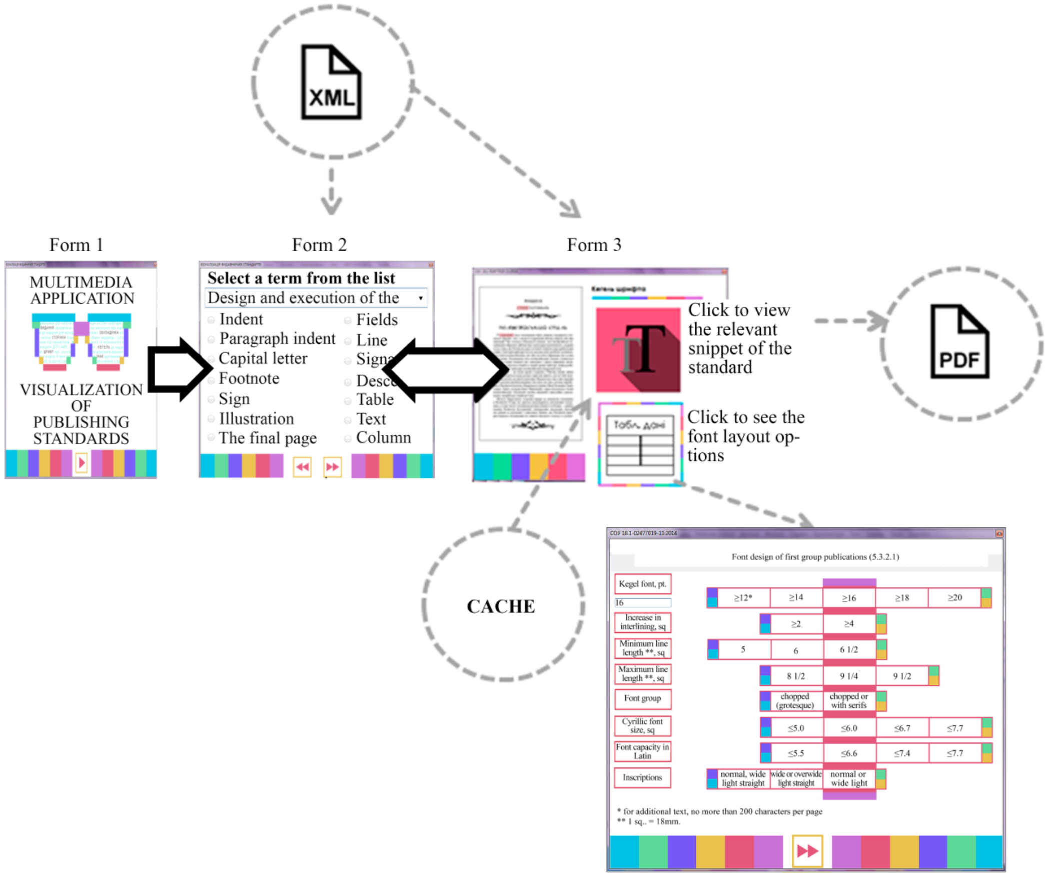
This code is in the class Form 4.cs.

```
          public Form4 ()
{
          InitializeComponent();
          // The key term chosen on Form3
          var keyword = Program.State.SelectedKeyword;
          if (keyword !=null)
          {
                    // Displays an image related to a key term
                    // If there is no image, show the default image
                    if(string.IsNullOrEmpty(keyword.Image))
                    {
                              pictureBox1.Image = Properties.Resources.no_image;
                    }
                    else
                    {
                              pictureBox1.Image = Bitmap(keyword.Image);
                    }
                    // Display the key term related icon on the PDF output button
                    // If there is no icon, show the default icon
                    if (string.IsNullOrEmpty(keyword.Icon))
                    {
                              button2.Image = Properties.Resources.no_icon;
                    }
                    else
                    {
                    button2.Image = new Bitmap(keyword.Icon);
                    }
                    // Output the name of the key term
                    label1.Text = keyword.Name;
               // Output of key term description
                    richTextBox1.Clear();
                    richTextBox1.SelectAll();
                    var i = richTextBox1.SelectionLength;
                    richTextBox1.AppendText((keyword.Description ?? "").Trim(new[] {
          ' ', '\r', '\n', '\t' }));
                              }
                    }
private void button2_click( object senter, EventArgs e)
{
        // search in the text typed in Form 2 for the key term selected in Form 3
        var chapters = TextSearch.FindString(Program.State.Text,
Program.State.SelectedKeyword.Name).ToArray();

        // if not found, show the message
        If (chapters.Length == 0)
        {
              messageBox.Show("No matching snippets were found!");
              return;
        }
        // create a temporary file to burn a PDF document
        var file = Path.GetTempFileName();
        file = Pats.ChangeExtension(file, "pdf");

        // Generate a PDF document in a temporary file
        PDFHelper.CreatePDF(file, chapters);
        // wait for the operation to complete
        System.Threading.Thread.Sleep(1800);
        // open pdf document
        Process.Start(file);
}
private void button1_click( object senter, EventArgs e)
{
        // click buttons to go to form 3
        var form3 = new Form3();
        form3.Show();
        this.Hide();
}
```

Fig. 14. Program code for Form 4

In the process of writing code for Form 4, two auxiliary classes were added – TextSearch.cs, which searches for fragments of text that was entered in Form 2, which contains the key term that is selected on Form 3 and PDFHelper.cs, which generates PDF-document.

The structure of the application "SOU 22.2-02477019-11: 2014 Printing. Edition for children. General Technical Requirements" has eight component forms, each of which has corresponding elements (which perform the corresponding navigation and interactive tasks) and performs functional tasks. The general concept of the multimedia application is presented in **Fig. 15**.


XML
Form 1
MULTIMEDIA APPLICATION
VISUALIZATION OF PUBLISHING STANDARDS
Form 2
Select a term from the list
Design and execution of the
Indent
Paragraph indent
Capital letter
Footnote
Sign
Illustration
The final page
Fields
Line
Sign
Desc
Table
Text
Column
Form 3
Click to view the relevant snippet of the standard
Click to see the font layout options
PDF
CACHE
СОУ 18.1-02477019-11:2014

Font design of first group publications (5.3.2.1)

Kegel font, pt. 16	≥12*	≥14	≥16	≥18	≥20
Increase in interlining, sq		≥2	≥4		
Minimum line length **, sq	5	6	6 1/2		
Maximum line length **, sq		8 1/2	9 1/4	9 1/2	
Font group		chopped (grotesque)	chopped or with serifs		
Cyrillic font size, sq		≤5.0	≤6.0	≤6.7	≤7.7
Font capacity in Latin		≤5.5	≤6.6	≤7.4	≤7.7
Inscriptions	normal, wide light straight	wide or overwide light straight	normal or wide light		

* for additional text, no more than 200 characters per page
** 1 sq.. = 18mm.

Fig. 15. The work scheme of the "Visualization of publishing standards" multimedia application

In general, the functionality remained from the previous application, but the new one is supplemented with a button that opens a window with visualized data tables. This button appears at the moment when the user has selected at least one key date, which is part of the tabular data. Regarding functionality, a textBox element was added, which allows the user to enter the value of the font size parameter and depending on what value is entered, the ratio of other parameters relative to the entered value changes. The element also implements the interactivity of the multimedia application, depending on the parameter entered, the font size of the font changes the appearance of the rendered data tables.

The textbox element code is shown in **Fig. 16**.

The result of the software implementation is the finished multimedia applications, which, according to the standardization visualization technique in terms of prepress preparation of publications, is the final product of the third stage of the presentation of the visual form.

Another example of the implementation of the metaphorical visualization method "parametric ruler" is a multimedia didactic complex in the discipline "Technological processes of publishing and printing" (Air Force TP), developed by the authors as part of the e-learning system (**Fig. 17**).

```
private void textBox_TextChanged(object sender, EventArgs e)
    { try
     {
        if (double.Parse(textBox1.Text) < 14 &&
double.Parse(textBox1.Text) >=12)
          {
            pictureBox1.Image =
PublicationStandartsVisualization.Properties.Resources._3_1;
          }
          if (double.Parse(textBox1.Text) < 16 &&
double.Parse(textBox1.Text) >= 14)
          {
            Clear();
            pictureBox1.Image =
PublicationStandartsVisualization.Properties.Resources._3_2;
          }
          if (double.Parse(textBox1.Text) < 18 &&
double.Parse(textBox1.Text) >= 16)
          {
           Clear();
           pictureBox1.Image =
PublicationStandartsVisualization.Properties.Resources._3_3;
          }
          if (double.Parse(textBox1.Text) < 20 &&
double.Parse(textBox1.Text) >= 18)
          {
           Clear();
            pictureBox1.Image =
PublicationStandartsVisualization.Properties.Resources._4_2;
          }
          else
          {
            if (double.Parse(textBox1.Text) >= 20)
            {
               Clear();
               pictureBox1.Image =
PublicationStandartsVisualization.Properties.Resources._4_3;
```

Fig. 16. Program code for the textbox element

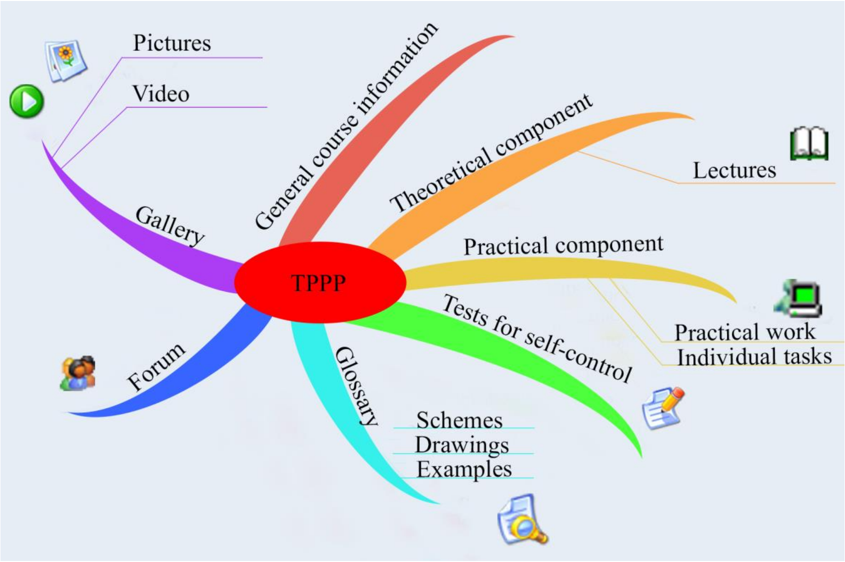


Fig. 17. General structure of the multimedia didactic complex in the discipline "Technological processes of publishing and printing" (TPPP)

For the created multimedia didactic complex, the following types of interactive infographic objects were used:

– infographic stories for building interesting and visual practical exercises;

– interactive test systems in the form of infographic tables;

– interactive maps for the organization and administration of the educational process, scheduling, student accounting and their performance;

– infographic tables that allow students to perform common tasks or consult on their implementation.

As possible limitations of using the proposed methodology for visualizing information for use in multimedia applications, there are individual characteristics of the user's mental activity formed under the influence of personal and situational factors. In particular, the proposed visualization technique can be effectively used by users of visual-effective and visual-figurative types of thinking. But objective difficulties with the use of visualization techniques may arise in users with a verbal-logical type of thinking.

6. Conclusions

Thus, as a result of research, a technique is proposed for visualizing information for use in multimedia applications, which facilitates the perception of a large amount of text information and ensures its presentation in multimedia systems. Using this technique allows to more fully cover a large amount of information for presentation in multimedia applications, reduce the cost of finding the necessary information and fix it at the visual level.

This study allows the user to explore the space of objects with their parameters, resulting in the possibility of optimizing the functionality of multimedia applications. The results of this research can be used in the design of multimedia applications both at the design stage to support decision-making on the choice of information presentation method, and to analyze the degree of optimization of the functionality of an already developed publication.

References

[1] Cerdas, F., Kaluza, A., Erkisi-Arici, S., Böhme, S., Herrmann, C. (2017). Improved Visualization in LCA Through the Application of Cluster Heat Maps. Procedia CIRP, 61, 732–737. doi: https://doi.org/10.1016/j.procir.2016.11.160

[2] Ahn, J., Brusilovsky, P. (2013). Adaptive visualization for exploratory information retrieval. Information Processing & Management, 49 (5), 1139–1164. doi: https://doi.org/10.1016/j.ipm.2013.01.007

[3] Corchado, E., Herrero, Á. (2011). Neural visualization of network traffic data for intrusion detection. Applied Soft Computing, 11 (2), 2042–2056. doi: https://doi.org/10.1016/j.asoc.2010.07.002

[4] Ben, X., Beijun, S., Weicheng, Y. (2013). Mining Developer Contribution in Open Source Software Using Visualization Techniques. 2013 Third International Conference on Intelligent System Design and Engineering Applications. doi: https://doi.org/10.1109/isdea.2012.223

[5] Chaolong, J., Hanning, W., Lili, W. (2016). Research on Visualization of Multi-Dimensional Real-Time Traffic Data Stream Based on Cloud Computing. Procedia Engineering, 137, 709–718. doi: https://doi.org/10.1016/j.proeng.2016.01.308

[6] Zhang, F., Tourre, V., Moreau, G. (2014). Using metrics to evaluate and improve text-based information visualization in 3D urban environment. WSCG2014 Conference on Computer Graphics, Visualization and Computer Vision, 375–382.

[7] Frey, G., Jurkschat, A., Korkut, S., Lutz, J., Dornberger, R. (2019). Intuitive Hand Gestures for the Interaction with Information Visualizations in Virtual Reality. Progress in IS, 261–273. doi: https://doi.org/10.1007/978-3-030-06246-0_19

[8] Vultur, O.-M., Pentiuc, S.-G., Lupu, V. (2016). Real-time gestural interface for navigation in virtual environment. 2016 International Conference on Development and Application Systems (DAS). doi: https://doi.org/10.1109/daas.2016.7492592

[9] Teras, M., Raghunathan, S. (2015). Big data visualisation in immersive virtual reality environments: embodied phenomenological perspectives to interaction. ICTACT Journal on Soft Computing, 05 (04), 1009–1015. doi: https://doi.org/10.21917/ijsc.2015.0141

[10] Hrabovskyi, Y., Fedorchenko, V. (2019). Development of the optimization model of the interface of multimedia edition. EUREKA: Physics and Engineering, 3, 3–12. doi: https://doi.org/10.21303/2461-4262.2019.00902

Received date 20.08.2019
Accepted date 07.12.2019
Published date 20.12.2019