Proceedings of the Third International Students Conference on Informatics

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Romania, Sibiu, May 16-18, 2013

ICDD

Actes de la Troisième Conférence Internationale d'Informatique pour les Eétudiants

IMAGINATION, CRÉATIVITÉ, DESIGN, DÉVELOPPEMENT

Roumanie, Sibiu, 16 -18 mai

Editor Dana Simian

Proceedings of the Third International Students Conference on Informatics

ICDD

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Romania, Sibiu, May 16-18, 2013

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of International Students
Conference on Informatics
ICDD

May 16th – 18th, 2013
Sibiu, Romania

“Lucian Blaga” University of Sibiu

Éditeur **DANA SIMIAN**

IMAGINATION, CRÉATIVITÉ, DESIGN, DÉVELOPPEMENT

Actes de la Troisième
Conférence Internationale d'Informatique
pour les Eétudiants

16 - 18 mai, 2013
Sibiu, Roumanie

Université "Lucian Blaga" de Sibiu

“Lucian Blaga” University, Sibiu, 2013

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN 2069-964X

Associated Editor Laura Stoica

Proceedings of International Students Conference on Informatics ICDD
May 16th – 18th, 2013, Sibiu, Romania

Université “Lucian Blaga” de Sibiu, 2013

Éditeur Dana Simian

Tous les articles de ce volume étaient examen par les pairs par deux examinateurs indépendants

ISSN: 2069-964X

Éditeur associé Laura Stoica

Actes de la Troisième Conférence Internationale d’Informatique pour les Eétudiants
16 – 18 mai, 2013, Sibiu, Roumanie

Motto:

“There are no limits, only your imagination”

Devise:

“Il n'y a pas de limites, seule votre imagination”

Proceedings of International Students Conference on Informatics, ICDD – 2013
IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT
Sibiu, Romania

Editor *Prof. PhD. Dana Simian*

Scientific committee

Actes de la Troisième Conférence Internationale d’Informatique pour les Eétudiants,
Imagination, Créativité, Design, Développement
Roumanie, Sibiu

Éditeur *Prof doctorat. Dana Simian*

Liste des membres du Comité scientifique

Table des mati¯res:

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Photo organiser
An application for a better organization of your photos

Cristian-Ionel Aldulea, Bogdan-Mihai Fieraru
Teacher Coordinator: Anca Vasilescu

Abstract

This application has been developed using C# programming language. The main sections of the application aim to "*Add Photo/Folder*" and to "*View Photos*". The first one enable us to add one or many photos in application database, by filling the specific dialog windows, and the second one, to view a certain list of photo accordingly with an user query. The application also provides the user the option to save the result of his query into an album for easily further specific photo visualization.

1 Why we made our choice

Since a regular computer has more than 500 GB on its hardware, users don't worry about the space when they download their pictures from camera. It yelds that users use to copy their picturs on Desktop or in folders spreaded on the Hard Disk. That produce a messy computer memory. Besides when the user needs to find a picture, he would never find it, or he would find it after hard searching.

That is why we thought we colud develop an application wich helps the user to access a specific photo in few seconds, just filling some fields. Similar application like Picasa are larger and slowly. Another problem from our point of view with Picasa is that when the user run for the first time the application it automatically scans all files from computer to find all photos.

2 About our application

2.1 Technology that we used

The application was developed using C# programming language and Microsoft SQL Server Compact 4.0 on .NET platform. C# is an OOP language which appeared in 2000, used by software developers for building .NET applications, and SQL Server, is a free embedded database.

2.2 System requirements

The system requirements for this software are low cost:

- Compatible Operating System (Windows XP/Vista/7/8)
- 1.8 GHz Processor
- 256MB RAM
- 5 MB free on HDD

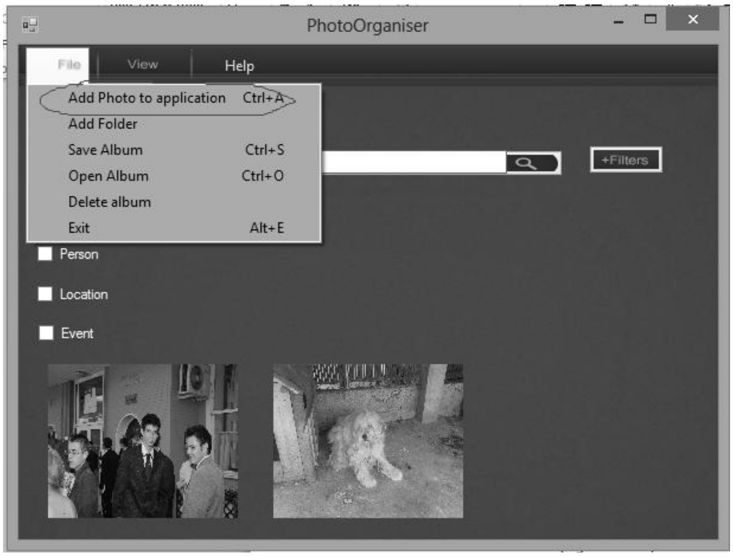
2.3 How the application works

The software consists in two parts, as follows. The first part refers the operation of uploading photos in the database. The uploading is efficient because the application don't change the disk location of any photo, don't create additional folders, only keep the address of each photo and load it when is necessary. We have designed four filters supporting the user desire to view a certain list of photo filtered by multiple criteria, or only search a photo by its name.

The second part of the application refers the organization of photos in albums, by allowing the user to save a certain search intro an album, for an easily access later. The user could also add specific photos to a specific album or he could delete a specific photo from an album. The albums could be deleted too, if they are not for interest anymore.

2.3.1 The "*Add photo*" function (photo nr1)

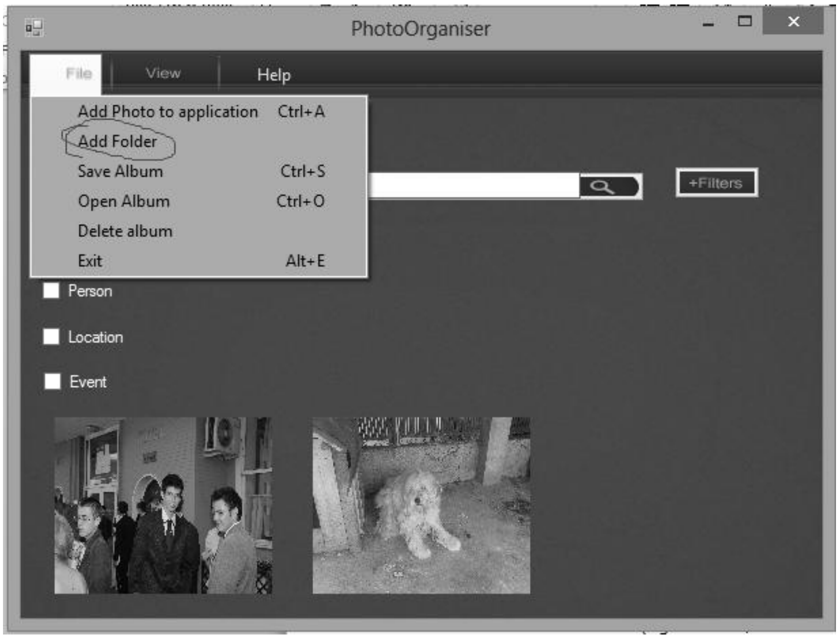
Using this function requires the user to open the "*Add photo to application*" window by clicking on its respective button or selecting the option from File Menu. This option enable adding one photo to the database.



1

2.3.2 "*Add folder*" function (photo nr2)

Add folder function enable the user to add multiple photos in the database at once, without selecting the photos one-by-one. The shortcoming could appear when the user wants to insert too many photos, he might get bored filling the fields for each photo individually.



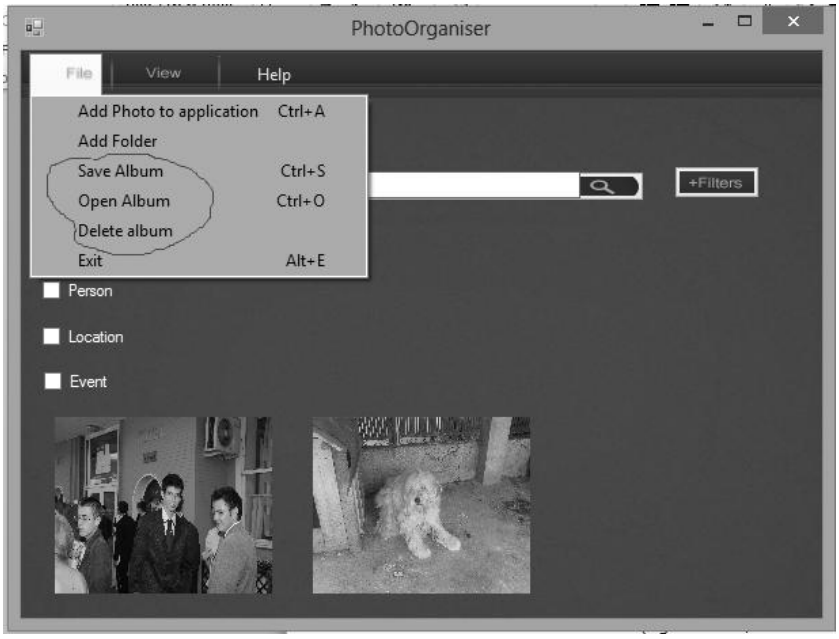
2

The following code is used to save a folder to the application.

```
private void button1_Click(object sender, EventArgs e)
{
   String date = textBox1.Text;
   String person = textBox2.Text;
   String location = textBox3.Text;
   String eveniment = textBox4.Text;
   String descriere = textBox5.Text;
   String nume = Path.GetFileNameWithoutExtension(files[poz]);
   string path =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
   SqlCeConnection sqlc = new SqlCeConnection(@"Data Source = " + path + "/Photos.sdf");
   sqlc.Open();
   SqlCeCommand sqlcmd = new SqlCeCommand("INSERT INTO Poza (Adresa , Data  , Persoane , Locatie ,
Eveniment , Descriere , Nume) VALUES (@adresa ,@data , @persoane , @locatie , @eveniment , @descriere ,
@nume)", sqlc);
      sqlcmd.Parameters.AddWithValue("@adresa", files[poz]);
      sqlcmd.Parameters.AddWithValue("@data", date);
      sqlcmd.Parameters.AddWithValue("@persoane", person);
      sqlcmd.Parameters.AddWithValue("@locatie", location);
      sqlcmd.Parameters.AddWithValue("@eveniment", eveniment);
      sqlcmd.Parameters.AddWithValue("@descriere", descriere);
      sqlcmd.Parameters.AddWithValue("@nume",nume);
      sqlcmd.ExecuteNonQuery();
      sqlc.Close();

}
```

2.3.3 Functions “*Save Album*”, “*Open Album*” and “*Delete Album*” (photo nr3)

After the user creates his album, the application allows to save it for later easily access using “*Save Album*” function from the *File Menu*. Also from the *File Menu*, the user is able to open a saved album or to delete one, if consider he won’t need it.



3

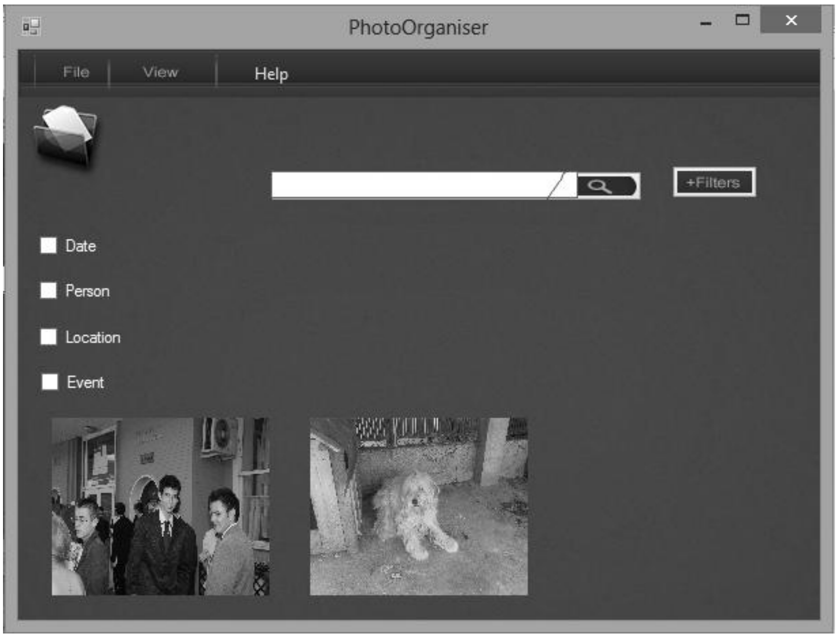
The following code is used to open an album.

```
private void openAlbumToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Form3 f3 = new Form3();
        f3.ShowDialog();
        string path = album;
        StreamReader sr = new StreamReader(path);

        for (int i = 0; i < pb.Length; i++)
          this.Controls.Remove(pb[i]);
        k = 0;
        bool ok = false;
        while ((s[k] = sr.ReadLine()) != null)
        {
          if (!File.Exists(s[k]))
          {
            MessageBox.Show("Photo " + s[k] + " was eliminated from album because it does not exist.");
            ok = true;
            k--;
          }

          if (k == s.Length - 1)
          {
            string[] new_s = new string[2 * s.Length];
            Array.Copy(s, new_s, k);
            s = new_s;
            k++;
          }
          else
            k++;
        }

        sr.Close();
        displayImage(s, k);
        if (ok == true)
        {
          File.Delete(path);
```

```
            StreamWriter sw = File.CreateText(path);
            for (int i = 0; i < k; i++)
                sw.WriteLine(s[i]);
            sw.Close();
        }
    }
```

2.3.4 The “*Search*” function (photo nr4)

This function is one of the most complex functions implemented by our software. It has a class in the back which knows how to handle with SQL queries. Following the fields values filled by the user, the function decide which class functions have to be called. The filter fields only appears after *Filters* button is pressed. The filters could also be disabled from the *View Menu*. In the *View Menu*, the application has another function, namely “*View All*”. This function allows the user to have an overview of all the pictures uploaded in applications databases.



4

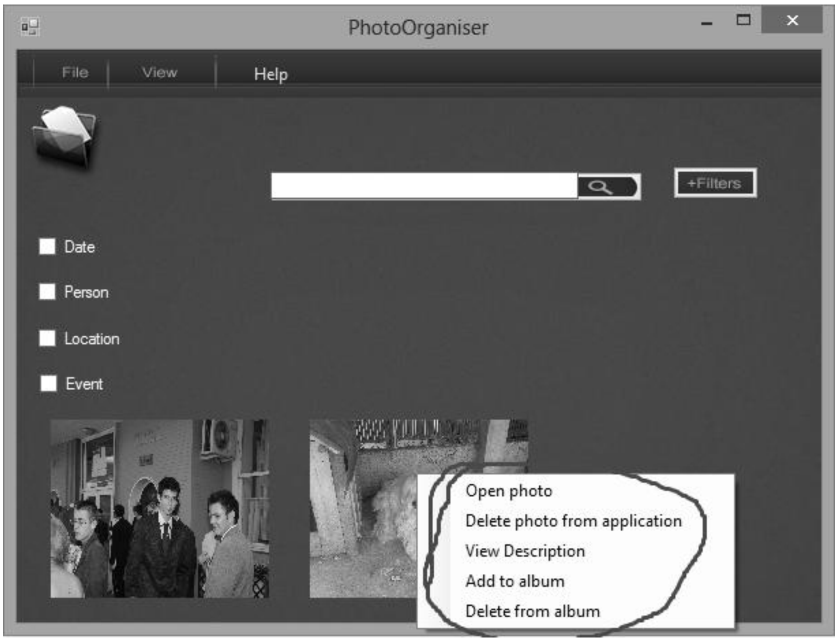
The following code is used to search an display photos.

```
private void button3_Click(object sender, EventArgs e)
{
    String nume = textBox6.Text;
    String dt1 = textBox1.Text;
    String dt2 = textBox5.Text;
    String person = textBox2.Text;
    String location = textBox3.Text;
    String eveniment = textBox4.Text;
    if ((checkBox1.Visible == true && dt1 != "") || (checkBox1.Visible == true && dt2 != ""))
    {
        if ((textBox1.Text != "" && textBox5.Text == "") || (textBox1.Text == "" && textBox5.Text != ""))
            MessageBox.Show("Fill both labels!");
        else
        {
            try
            {
                DateTime dt11 = DateTime.Parse(dt1);
                DateTime dt12 = DateTime.Parse(dt2);
            }
            catch (Exception err)
```

```
                {
                    MessageBox.Show("Date format failure! Fill date fields again!");
                    textBox1.Text = "";
                    textBox5.Text = "";
                }
            }
        }
        for (int i = 0; i < pb.Length; i++)
            this.Controls.Remove(pb[i]);
        //Name
        if (nume != "" && dt1=="" && dt2=="" && eveniment=="" && person =="" && location =="")
        {
            s = Baza.sNume(nume);
            k = getK(s);
            verif(s, k);
            displayImage(s, k);
        }
        //Date
        if (nume == "" && dt1 != "" && dt2 != "" && eveniment == "" && person == "" && location == "")
        {
            s = Baza.sData(dt1,dt2);
            k = getK(s);
            verif(s, k);
            displayImage(s, k);
        }
…..
//Name, Event, Location, Person, Date
        if (nume != "" && dt1 != "" && dt2 != "" && eveniment != "" && person != "" && location != "")
        {
            s = Baza.sNELPD(nume, eveniment, location, person, dt1, dt2);
            k = getK(s);
            verif(s, k);
            displayImage(s, k);
        }

      }
```

2.3.5 *Right Click* on photo functions (photo nr5)

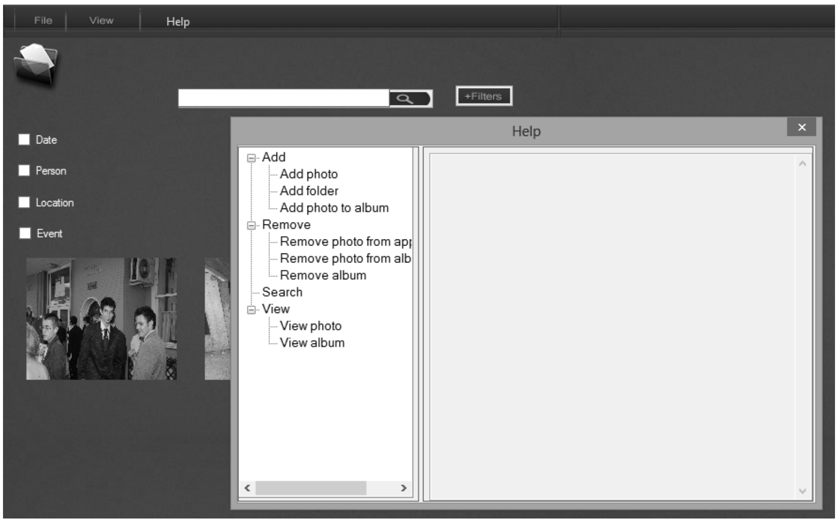
Right clicking a certain photo, a useful category of functions is accessed. The first one in the list is the "*Open Photo*" function, which enables the user to admire each picture, zoomed. Left clicking on this function name opens a form, where the user is able to see the photos following the *Next* and *Previous* buttons. Another function in the context list is ”*Delete photo from application*". The details about each photo can be seen by selecting the "*View Description*" function. If it is necessary, a photo could be assigned to an album, or it could be deleted from an album using the "*Add to album*" or "*Delete from album*" function, also by right clicking on that photo.



5

2.3.6 User Help (photo nr6)

Each of the functions listed above has a help option which explains in details its functionality and how it should be used.



6

3 Technical aspects

The application was developed using C# programming language and Microsoft SQL Server Compact 4.0 on .NET platform.

Here is a breakdown of the most important functions:

The following is the function which realise the searching by name:

```
public static String[] sNume(String nume)
{
   SqlCeConnection con = getconnection();
   con.Open();
   int k = 0;
   string[] s = new string[2];
   SqlCeCommand cmd = new SqlCeCommand("SELECT adresa from Poza where nume = (@nume)", con);
   cmd.Parameters.AddWithValue("@nume", nume);
   SqlCeDataReader reader = cmd.ExecuteReader();

   while (reader.Read())
   {
      if (k == s.Length - 1)
      {
         string[] new_s = new string[2 * s.Length];
         Array.Copy(s, new_s, k);
         s = new_s;
         s[k++] = reader.GetString(0);
      }
      else
      {
         s[k++] = reader.GetString(0);
      }
   }
   return s;
}
```

The following function deletes a photo from the databases:

```
private void elim(string s)
      {
         SqlCeConnection con = getconnection();
         con.Open();
         SqlCeCommand cmd = new SqlCeCommand("DELETE FROM Poza WHERE adresa = (@adresa)", con);
         cmd.Parameters.AddWithValue("@adresa", s);
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("A fost eliminata din baza de date.");
      }
```

The next function display a photo on the screen:

```
private void displayImage(String[] s, int n)
      {
         pb = new PictureBox[s.Length];
         int i = 0;
         int x1 = x, y1 = y;

         while (i < n)
```

```
  {
    if (i < n && File.Exists(s[i]))
    {
      Bitmap bmp = new Bitmap(s[i]);
      Bitmap result = new Bitmap(160,130);
      Graphics g = Graphics.FromImage(result);
      g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;
      g.DrawImage(bmp, 0, 0, 160, 130);

      pb[i] = new PictureBox();
      pb[i].Name = s[i];
      pb[i].Image = result;
      pb[i].Size = new Size(160, 130);
      pb[i].Location = new Point(x1, y1);
      pb[i].SizeMode = PictureBoxSizeMode.Zoom;
      this.Controls.Add(pb[i]);
      pb[i].ContextMenuStrip = contextMenuStrip1;
      i++;
    }

    if (i < n && File.Exists(s[i]))
    {
      Bitmap bmp = new Bitmap(s[i]);
      Bitmap result = new Bitmap(160, 130);
      Graphics g = Graphics.FromImage(result);
      g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;
      g.DrawImage(bmp, 0, 0, 160, 130);

      pb[i] = new PictureBox();
      pb[i].Name = s[i];
      pb[i].Image = result;
      pb[i].Size = new Size(160, 130);
      x1 += distx;
      pb[i].Location = new Point(x1, y1);
      pb[i].SizeMode = PictureBoxSizeMode.Zoom;
       this.Controls.Add(pb[i]);
      pb[i].ContextMenuStrip = contextMenuStrip1;
      i++;
    }

    if (i < n && File.Exists(s[i]))
    {
      Bitmap bmp = new Bitmap(s[i]);
      Bitmap result = new Bitmap(160, 130);
      Graphics g = Graphics.FromImage(result);
      g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.NearestNeighbor;
      g.DrawImage(bmp, 0, 0, 160, 130);

      pb[i] = new PictureBox();
      pb[i].Name = s[i];
      pb[i].Image = result;
      pb[i].Size = new Size(160, 130);
      x1 += distx;
      pb[i].Location = new Point(x1, y1);
      pb[i].SizeMode = PictureBoxSizeMode.Zoom;
       this.Controls.Add(pb[i]);
      pb[i].ContextMenuStrip = contextMenuStrip1;
      i++;
      y1 += disty;
      x1 = x;
    }
  }
}
```

4.Conclusions and future developments

This application was built for viewing and organizing the photos after desired criteria. As an upgrade, we thought that it would be a further step if we add a multiple-person search option. We think that our application is productive, as it takes up little space on the hard drive, it’s fast, efficient and it does its job well.

5.References

[1] Limbajul C#, Lucian Sasu, 8 iunie 2012, Brasov
[2] Microsoft Developer Network , http://msdn.microsoft.com/library/vstudio

Aldulea Cristian Ionel
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
Romania
alduleacristi@yahoo.com

Fieraru Bogdan Mihai
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
Romania
fierarubogdan2000@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Determine the number of twitter's user clusters

Alexandru Petrut Becheru
Teacher Coordinator: Catalin Stoean

Abstract

Now days social media has become very important although it might look like nonsense or a waste of time, there is more to it that you can imagine. Data obtained from this SMS like social network can be used in many fields such as marketing, or less known as earthquake alert service [5]. In my work we used a dataset provided at UMAP 11[6] conference containing over 10 million tweets and 20000 users. In particular the database used was provided within this dataset, where for every tweet was associated a measure of similarity with one of the 18 topics. By using K-means algorithm we tried to clusterise the users, and determine the number of clusters. Besides determing the number of clusters we aslo proposed a better method of creating a user model which is computational faster.

1 Introduction

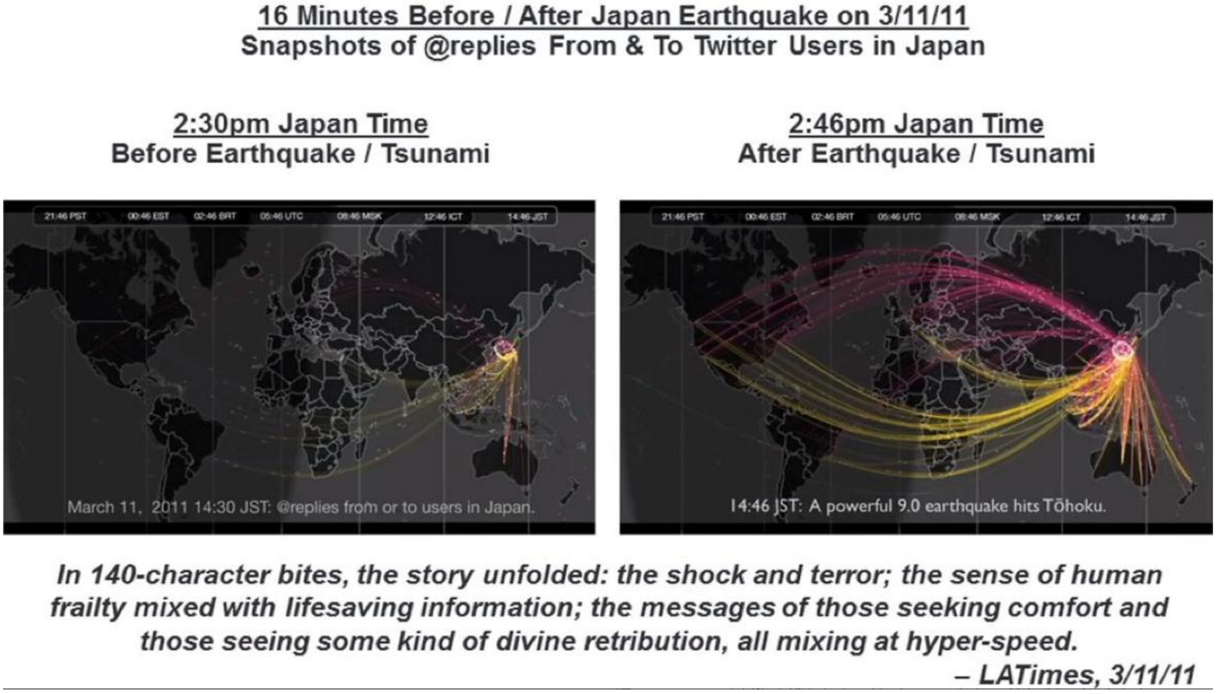
Created in 2006 Twitter is the second best known social network on the planet, with over 500 million users it is only surpassed by Facebook. Also it is one of the most 10 visited web sites. This service can be used to post messages, but no longer than 140 characters, although the length of the message may be considered a drawback, this was actually a win. Having so little space to write users have to just write the essential, making in perfect for changing information fast and reliable. One of the most important uses of twitter is newsgathering; this is done almost in real time, and with persons that are at the place of the event. As shown in Figure 1 almost immediately after the earthquake that shook Japan in 2011 news were changed all over the world.

Clustering is a very important technique that groups a set of objects in a way that the elements of a cluster are more similar together than those in other groups. This can be used in many fields of work, like analysis. On the internet the main objective for clustering is to aid in recommending content or users. This way a user receives for example only advertisements from products that are in his filed of interest, thus increasing the potential of advertised products to be bought.

This paper is mainly based on the work done by Fabian Abel et al.[2]. We used their dataset together with the findings of another paper called "Clustering Users in Twitter Based on Interests" [7] as a starting point.

In this paper we are trying to find the cluster types of users based on the topics that they tweeted. We chose to use the well known K-means[3] algorithm that is kwon to give good and fast results. At the beginning the cosine method was used to create user models, but as it used a lot of floating point calculations we proposed another method, which is computational faster and also gives better results in clustering.

This paper is structured the following way: the next chapter speaks about creating the user model and K-means algorithm, then we use these to determine the number of clusters and compare the two modes of creating user models, finally the conclusions and further work are stipulated.



16 Minutes Before / After Japan Earthquake on 3/11/11
Snapshots of @replies From & To Twitter Users in Japan

2:30pm Japan Time
Before Earthquake / Tsunami

2:46pm Japan Time
After Earthquake / Tsunami

21:46 PST
00:46 EST
02:46 BRT
05:46 UTC
08:46 MSK
12:46 ICT
14:46 JST
March 11, 2011 14:30 JST: @replies from or to users in Japan.
14:46 JST: A powerful 9.0 earthquake hits Tōhoku.

In 140-character bites, the story unfolded: the shock and terror; the sense of human frailty mixed with lifesaving information; the messages of those seeking comfort and those seeing some kind of divine retribution, all mixing at hyper-speed.
– LATimes, 3/11/11

Figure 1 Map of tweets sent and received before and after the 2011 Japan earthquake.

2 Creating the user model and K-means

2.1 User model

First of all we had to create a user model for each model. This will be used to further compare users using different metrics.

As said above each user has a number of tweets each tweet being associated with a topic based on a metric of similarity. Basically a number between 0 and 1, where 0 means that the tweet is not at all similar with that topic, and 1 says that it is surely on the domain of that topic.

For a certain user we took one specific topic and made the average between the sum of the measures of similarity and the number of tweets for that topic. So the average of user relevance to that topic, this being a number between 0 and 1. We repeated this operation for every topic of that user and got a vector with topic average topic relevance for all the topic.

Example 1 *User generated vector***:**

[0.582, 0, 0, 0.16325, 0, 0, 0.32473333333333334, 1.0, 0, 0, 0, 0, 0.754, 0, 0.9135, 0.7881, 0, 0]

We did all this for all the users present in the database. At a closer look we observered that may of the users posted just to one or two topics. This is very easy to classify, so we decided to take in consideration only the users that posted in at least 2/3 of the topics, or in our case 12 topics. This being the case, we obtained 2160 users out of a total of more than 20000. Considering the fact that the makers of this dataset used about 1600 users for their paper work, the current number of users is sufficient to work on.

This vector can be a good user model, but we decided to create another one based on this vector, just to have a better overview of the results. So for a random user the vector above and ordered it. This being done it resulted a ordered list of user relevance to topics.

Example 2 *List creation:*

Having a vector of topic relevance like this: [0.3; 0.2; 0.5] relevant for the topics [Other, Sports, Law]; We obtained a list: Sports, Other, Law; In other words this means that this particular user is most relevant in the topic sport followed by the others.

Example 3 *User list:*

['Disaster_Accident', 'Education', 'Environment', 'Health_Medical_Pharma', 'Labor', 'aw_Crime', 'Other', 'Politics', 'Social Issues', 'War_Conflict', 'Weather', 'Entertainment_Culture', 'Hospitality_Recreation', 'Business_Finance', 'Religion_Belief', 'Technology_Internet', 'Sports', 'Human Interest']

This should be a also a very good user model. For simplicity we will further refer to this as the topic list, and the first one with cosine list.

2.2 Metric of similarity for K-means

For the cosine list we decided to use the cosine similarity, which is very well known for it's good results. This is adapted for calculating the similarity between two vectors, and it is not compute intensive. This is represented using a dot product (1), and a magnitude (2). Where vectorA and vectorB represent the two vectors to be compared, and (i) being one of the 18 topics.

$$\sum_{i=1}^{18} vectorA_i * vectorB_i \qquad (1)$$

$$0.5 * \sum_{i=1}^{18} vectorA_i{}^2 \qquad (2)$$

$$\frac{\sum_{i=1}^{18} vectorA_i * vectorB_i}{\left(0.5 * \sum_{i=1}^{18} vectorA_i{}^2\right) * \left(0.5 * \sum_{i=1}^{18} vectorB_i{}^2\right)} \qquad (3)$$

We calculated the cosine similarity (3) between the first user and the rest of users. Then this results were put in a separate file, so all this calculation have to be done only once, reducing the computing time.

For the list topic we developed a measure that calculates the sum of distances between topics in the list. Equation (4) depicts this measure of similarity, where $posA_i$ represents the position of the topic (i) in the list A, same with $posB_i$ but for list B.

$$\sum_{i=1}^{18} |posA_i - posB_i| \qquad (4)$$

Example ***4*** *Calculating the distance between two topics:*

Having two lists: [Law, Labor, Other] and [Labor, Other, Law]

The distance of the Law topic is 2, on the first list it is positioned as first and in the second list it is positioned as last. So the difference between position 3 and position 1 equals 2. Doing the same with Labor and Other we get: Labor distance = 1, Other distance =1; Summing this distances and we get that the total distance between this two lists to be equal to 4.

Again we calculated the distances between the first user and the rest of users and put them in a separate file.

2.3 K-means

For the K-means algorithm [1] we decided that the stopping condition should be either if 51 generations passed or if the number of changes is equal or lower to 20. We determined the number of generations by experimenting multiple trials, and concluded that this number should suffice, if it was lower then a better results may be omitted, but if it were larger no significant improvement was seen. The number of maximal changes permitted was chosen 20, as it represent less then 1% of the total number of users.

To determine how good a clustering was we used the sum of squared error or SSE (5). At the end of the K-means algorithm it would take each user in the cluster, and square it's distance to its particular centroid. Then summing these squared distances and get the SSE value for that particular clusterisation. You can find the formula in Equation 5, where $posCentroid_i$ represents the position of the centroid for the user (i), and pos_i the position of the user (i).

$$\sum_{i=1}^{2159} (posCentroid_i - pos_i)^2 \quad (5)$$

We also implemented some visual drawing just to make a better sense of the shape of the users' clusters. In the first picture we have a representation of the users with 5 clusters for the cosine implementation. As you can see it is very hard to differentiate the different clusters.

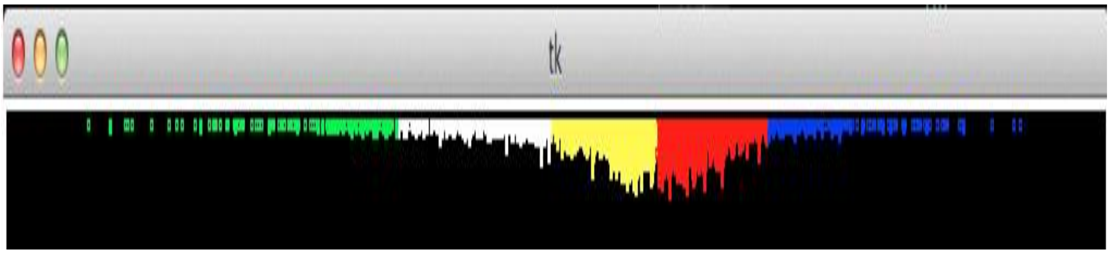


Figure 2 Cosine plot

In the second picture the topic list implementation was used, also for 5 clusters. You can notice more clearly the possible different clusters, one cluster may be represented by a bar. We counted them and got 60 different bars.

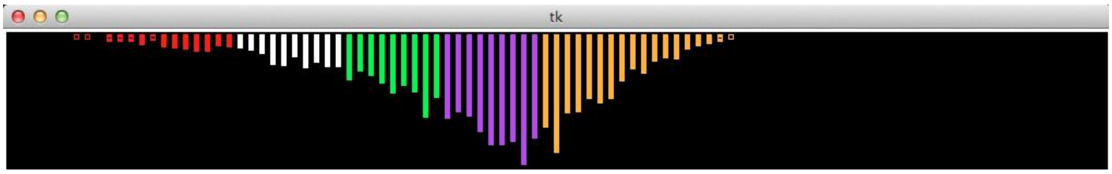


Figure 3 Topic list plot

3 Determining the number of clusters

To determine the number of clusters we calculated the SSE for every k between 2 and 99, where k is the number of clusters. Repeated this several times for both methods of user representation. Then calculated and average of SSE value for each k value. We observed that somewhere after the 50 generation the gain of SSE is less, as seen in Figure 4 below for the cosine method.

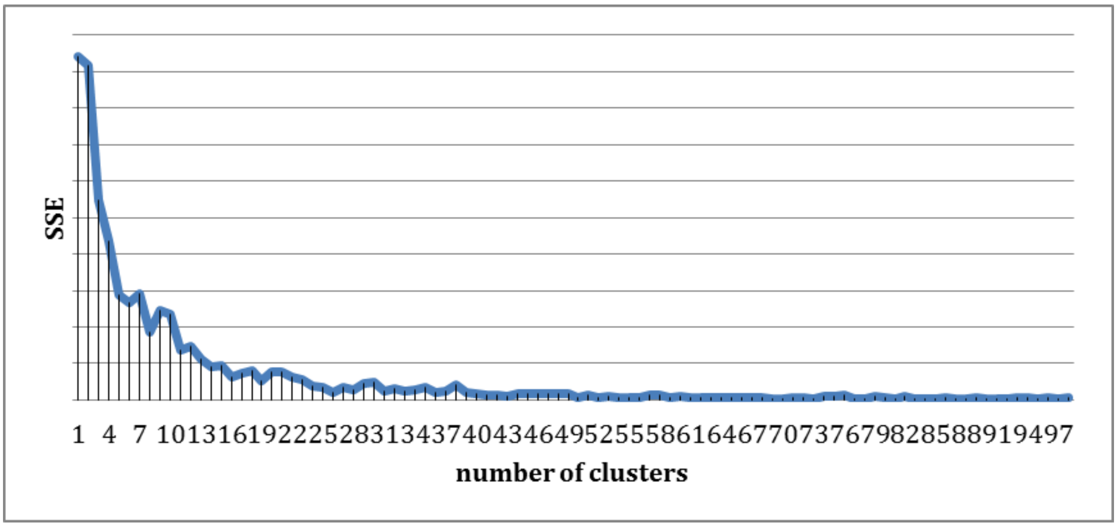


Figure 4 Average for cosine method

Knowing that the SSE values goes down for every increase in the cluster number, we thought of normalizing this. So we multiplied the average value of the SSE k number of clusters with k. For example if SSE was 10 for k=5 then it will become 5*10.

Plotting this was a great surprise as it could better seen that for k=55 the values is very low, actually just after k reaches 90 or more the values goes down. This can be seen on Figure 5.

Then repeated the same thing with the topic list method of user modeling and not surprisingly we got results showing that 55 could be the number of clusters. See Figure 6 and 7.

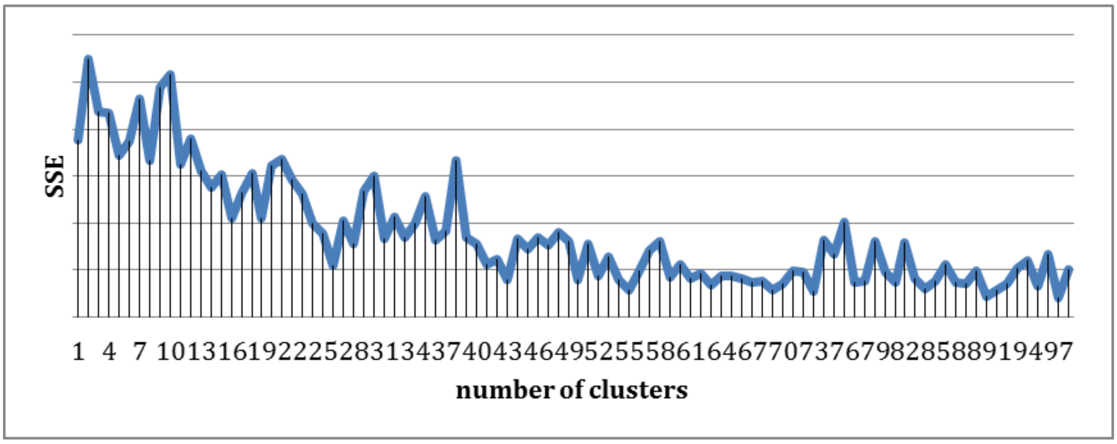


Figure 5 Normalized average for cosine method

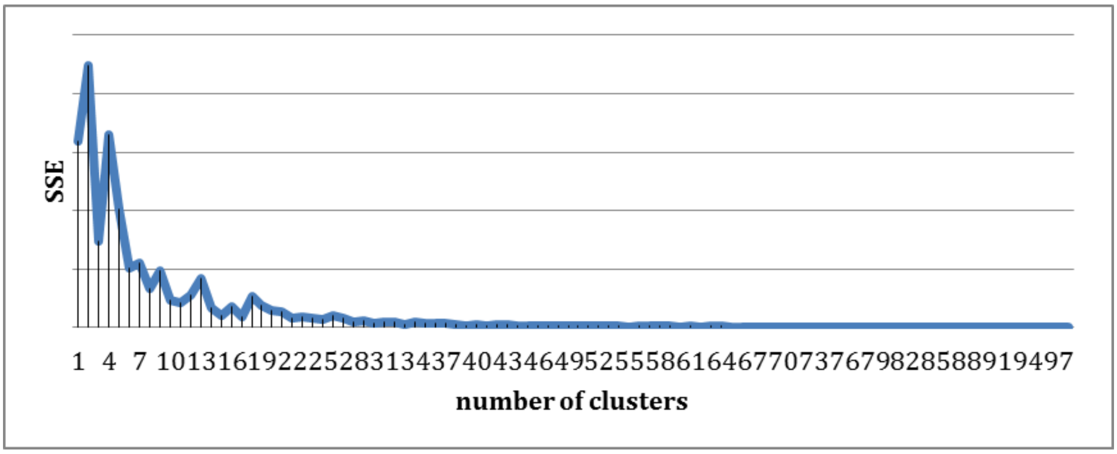


Figure 6 Average for topic list method

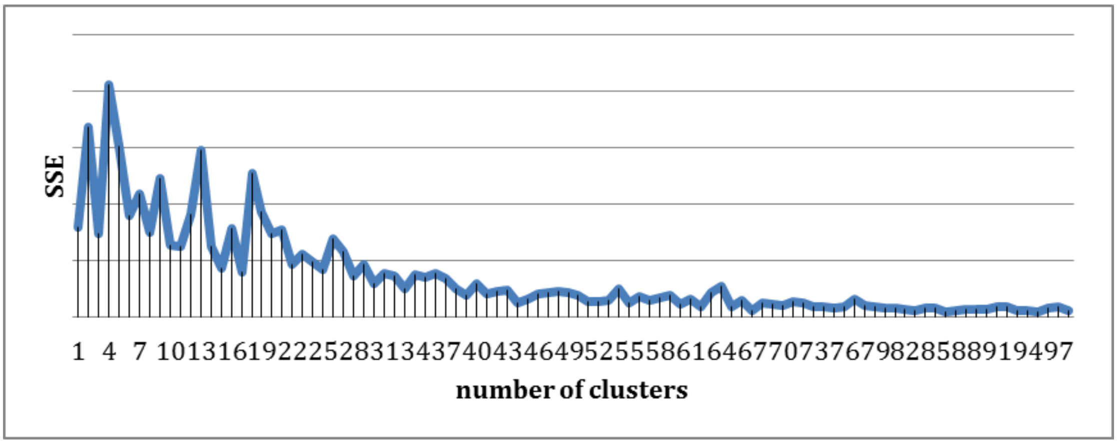


Figure 7 Normalized average for cosine method

Next we decided to plot the normalized averages for both methods together, so I could make a better sense. Although 55 seemed to be a good number, 43 or even 66 is good.

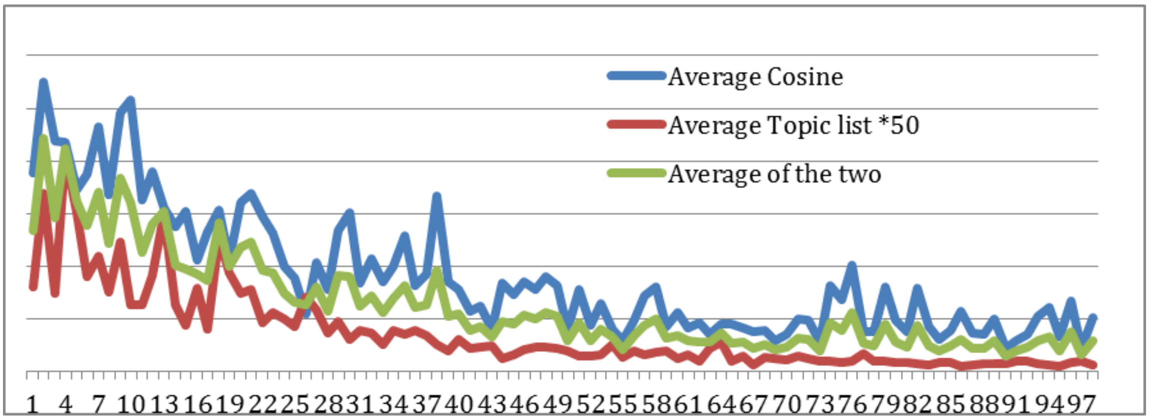


Figure 8 Combined normalized averages

To plot Figure 8 in a comprehensive manner we had to multiply the value of the topic list average by 50. This being said we can see here that 55 is the best value for k. This also consistent with the number of bars found in Figure 3, counted to be 60 but considering that there some outliers this should further strengthen out opinion.

4 Conclusions and further work

We concluded that k-means algorithm gives good results regarding clusterisation of twitter users, the speed of it depending on the type of user model implemented. Altough both methods give the same reuslt for the future we recommend using the topic list approach as it is easier to compute, and just looking at the user plot could easier identify the number of clusters. Altough this proposed method is very simple it is better in every aspect.

For the future many more methonds of user modeling should be implemented and also other clustering algorithms. Also the algorithms should be tweeked and modified to better suite the scenarios.

References

[1] Brusilovsky, P., Kobsa, A. e Nejdl, W. *The Adaptive Web: Methods and Strategies for Web Personalization*, Springer, LNCS 4321, 2007.
[2] Fabian Abel, Qi Gao and Geert-Jan Houben, Ke Tao, *Analyzing User Modeling on Twitter for Personalized News Recommendations*. Procedings of UMAP 2011, Girona, Spain.
[3] Florin Gorunescu, *Data Mining Concepts and Techniques*, Springer, 2011.
[4] A. Micarelli, *Course of Inteligent Systems for Internet 2012-2013*, University of Roma 3, Roma, Italy, https://sites.google.com/site/sistemiintelligentiperinternet/
[5] *Twitter faster than earthquake*, Mashable 26-8-2011, http://mashable.com/2011/08/26/twitter-earthquake-video/
[6] *UMAP 11*, http://www.umap2011.org/
[7] *Clustering Users in Twitter Based on Interests*, http://www.nlpr.ia.ac.cn/2011papers/gnhy/nh4.pdf.

ALEXANDRU PETRUT BECHERU
University of Craiova
Faculty of Exact Sciences
Department of Computer Science
Str. A.I.Cuza, Nr. 13, Craiova
ROMANIA
E-mail: alex.becheru@inf.ucv.ro

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Finding regulatory motifs in the Solanum Lycopersicum genome

Michelle Blackwood
Teacher Coordinator: Sofia Visa, Esther Van der Knaap

Abstract

Finding regulatory motifs (also known as promoters) is very similar to searching for a needle in a haystack. Therefore the computational step of identifying potential promoters is valuable as it narrows down dramatically the number of, more precise but very laborious and time-consuming, lab experiments. Here we use promoter predictor software as well as our own specific approach to computationally identify such potential motifs in tomato fruit genome. Nine such potential promoters are found, which can be further validated by in-lab analysis.

1 Introduction

Finding genetic regulatory networks in an organism is very important as most of the time genes do not act in isolation; rather many genes interact with each other triggering other genes' expression or leading to a particular behaviour or trait. Discovering such genetic interactions is difficult because diverse genetic information must be aggregated in order to find dependencies. For instance, to find genetic regulatory networks in the tomato fruit (Solanum Lycopersicum) information about differentially expressed genes at various developmental time points together with regulatory motifs (also known as promoters or transcription factor biding sites) is needed. In this paper we address the promoter identification part. Once we identify promoters in the tomato fruit, we will investigate co-regulated genes. Further, these two pieces of information can be used to discover regulatory networks.

The results presented here are part of a larger NSF funded study that investigates genes that regulate size and shape in tomato fruits [2]. In [1] three genes that influence size and shape of tomato fruits are identified, namely, ovate, which gives the fruit a pear shape-like; fas, which results in a larger and flat tomato; and sun which significantly enlarges and elongates a tomato fruit. The interaction between these three morphological genes where further studied, but the outcomes (i.e. the degree of change in shape and size of fruits) cannot be fully explained by these interactions alone. This leads to the hypothesis that the size and shape traits are a result of the interaction between more genes.

In this bioinformatics research we seek to identify potential promoters in the Solanum Lycopersicum. As pointed above, knowing promoters in an organism is important because it further helps researching co-regulated genes and genetic networks. Promoters are short DNA sequences (4-10 nucleotides), which initiate transcription of genes. As Fig. 1 shows, they are located upstream of a gene anywhere within 1,000-3,000 bases. Here we search for promoters within the first 1,000 upstream bases because they occur more frequently in this region.

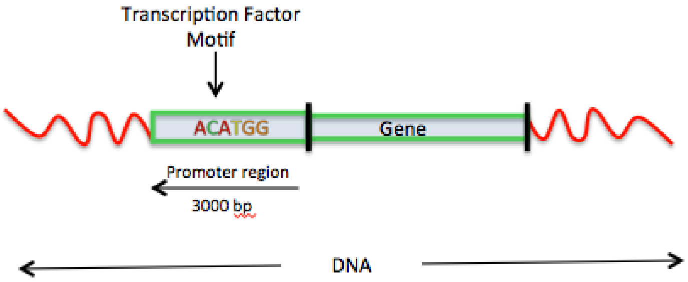


Fig. 1 Promoters are found anywhere within the 3,000 bases upstream of a gene.

Identifying such short sequences is similar to searching a needle in a haystack. Thus performing this search experimentally using laboratory techniques is unfeasible: such experiments are very laborious, take a long time (years), and are not easy to set up. Therefore, it is very important to computationally identify such potential motifs and, eventually, to further narrow down the set of candidates. Once a pool of likely promoters is identified, future more precise in-lab testing will validate the true ones.

2 The data set

Two different tomato backgrounds are used here to identify tomato fruit promoters: the wild type tomato and its mutation carrying the ovate gene. Fig. 2 illustrates the morphology of these two tomato varieties. To build our dataset for promoter analysis first we identify differentially expressed genes, then we isolate 1,000 upstream bases for each of the differentially expressed genes.

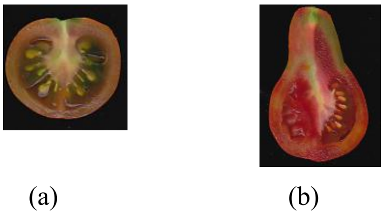


Fig. 2 Size and shape of wild type tomato fruit (a) and its mutation carrying ovate gene (b).

2.1 Finding differentially expressed genes

To identify differentially expressed genes in the two tomato varieties, we developed a computational pipeline (written in Python and R), which first, using Tophat software [10], aligns the tomato Illumina data (referred to as reads) to the tomato genome; then computes the read counts per gene; and finally selects the genes that are differentially expressed (for this latest step we used DEseq package [11]). There are three replicas for each of the two genotypes, with each replica consisting of approximately 2 million reads.

Table 1 shows the break down of the 396 differentially expressed genes (out of 35,000 known tomato genes) identified by our analysis. A fold increase in Table 1 signifies an up-regulated gene in the ovate mutation variety. Similarly, a fold decrease indicates a down-regulated gene. For instance, third row of the table shows that 24 genes are expressed twice as much in the pear-shape-like tomato fruit (ovate mutation) as in the round variety (wild type tomato).

Group number	Number of genes in the group	Fold increase/decrease in gene expression level
1	11	>= 3
2	24	2
3	94	1
4	75	Between 0 and 1
5	82	Between -1 and 0
6	77	-1
7	25	-2
8	8	<= -3

Table 1. The 396 differentially expressed genes grouped by fold increase/decrease of their expression level.

2.2 Extracting upstream (promoter) region

For each of these genes, we extract 1,000 bases located upstream of these genes and save them in a fasta file. This file contains the working data (referred here as the dataset) that will be used for the rest of the research presented here. To access the tomato genome and extract the desired region, we wrote Python code that used the Biopython package [12].

3 Identifying regulatory motifs

In this section we discuss two different approaches in identifying regulatory motifs in tomato fruits. First one involves the use of promoter predictor software, which did not give conclusive results. Therefore, we designed a second approach more specific to our data, which can be adapted to other plants or organisms as well.

3.1 Using regulatory motif predictors

We used three different bioinformatics tools to identify unusually common short sequences in our promoter region data, namely Cosmo [3], MEME [4], and MosDi [5]. For a detailed explanation of these predictors we refer our readers to their reference. These promoter predictors report E-values for each candidate; the closer the E-value to 0, the more likely the corresponding motif is a true one and not overly represented due to chance. To validate that the found motifs are statistically significant and are not present due to chance alone, we shuffle the search regions 1,000 times and repeated the prediction analysis. The hypothesis is that if the same motif is found in the shuffled sequence, then it is not a true motif.

In general, promoter prediction software uses some probabilistic or statistical model to produce likely motifs for some given sequences. Such software looks for unusually frequent motifs relative to the background sequence. For instance, it might use a hidden Markov model that additionally takes in consideration the background sequence to increase the likelihood of a predicted motif to be a true motif. However, such models work best if the pool of sequences analyzed together share a common motif. But in our 396 differentially expressed genes it is very likely that there are several motifs, each regulating a different subset of these genes. Yet, a predictor looks for a unique motif for all 396 genes. Therefore we suspect that a predictor either finds an unlikely unique motif (maybe) encompassing parts of individual true motifs, or it finds one of the several true motifs but it assigns it a biased and less meaningful E-value.

Therefore we try some intuitive grouping of the data to increase the chances of finding good promoter candidates:

Dataset 1: consists of upstream of all 396 genes;

Dataset 2: comprises upstreams of ten known up-regulated genes and 10 down-regulated genes;

Dataset 3: consists of the eight groups shown in Table 1. Each group reflects the fold increase or decrease in its member genes' expression level.

For the reasons discussed above, the predictors are run on each individual group of a dataset. The three predictors give similar results by predicting promoters with high E-values that indicate poor likelihood of being true promoters. For illustration we include below several such results obtained with MEME (ZOOPS parameters).

Dataset 1 For the pool of 396 genes the motif GTTTTTG is identified as a possible motif, although with an infinite E-value. This indicates that several different motifs might regulate different genes in this pool and a more focused (per subgroups) search will help identifying more significant motifs.

Dataset 2 Fig. 3 shows the most likely motif found in this dataset. The size of a nucleotide in the figure indicates its likelihood of occupying that position. One instance of this motif, CCCACC, is well known in plants as the morning motif and is shared by eight out of 10 genes. This motif has a very low E-value which indicates that its frequency relative to the background sequence is significant and not due to chance. Fig. 4 shows the motif found in the shuffled 10 sequences, which is very different than the motif shown in Fig. 3. This result strengthens our belief that the morning motif is a true promoter in the tomato genome. In addition, the motif found in the shuffled sequences has a very high E-value, which indicates that its relatively frequent occurrence is very likely due to chance.

In the subset of 10 up-regulated genes MEME identifies the motif shown in Fig. 5. However, its corresponding E-value indicates that it is not a unique promoter for all 10 genes. Fig. 6 illustrates an unlikely motif found in the corresponding shuffled dataset.

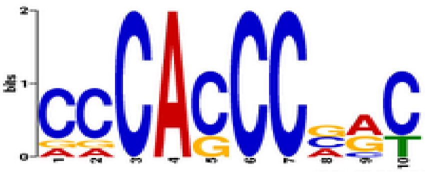


Fig. 3 Morning motif found in the 10 down-regulated genes (its low E-value of 2.9e-002 indicates that it is likely a true motif).

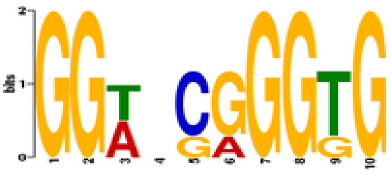


Fig. 4 Motif found in the shuffled 10 down-regulated genes (its high E-value of 6.8e+001 indicates that it is likely due to chance).

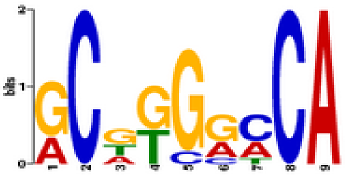


Fig. 5 Top motif found in the 10 up-regulated genes (E-value = 6.6e+000). Based on the e-value, it is not likely to be a true motif.

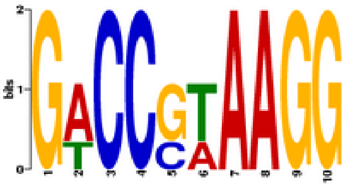


Fig. 6 Motif found in the shuffled 10 up-regulated genes (its high E-value of 7.5+002 indicates that it is likely due to chance).

Dataset 3 Only group 8 (the 3-fold down-regulated genes) of dataset 3 produces a candidate promoter (the morning motif found in dataset 2) with an acceptable E-value (see Fig. 7). The top predictions of the other groups (1-7) have high E-values indicating that the motifs found are unlikely, or that the genes within each group might be regulated by several different motifs.

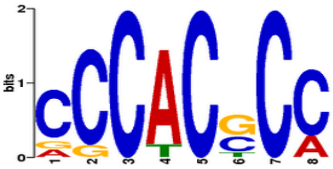


Fig. 7 Motif found in group 8 of dataset 3 with an E-value of 3.8e+000

This experiment suggests that a different gene grouping might lead to better (i.e. low E-value and therefore more likely) predictions, but it is difficult to know beforehand such better grouping. The outcome of this first experiment, namely (1) only two promoter candidates found for our genes of interest, and (2) a better but unknown grouping of genes might lead to the identification of other promoters, led us to approach this problem from a different direction, as detailed next.

3.2 Using Arabidopsis known regulatory motifs to identify promoters in tomato fruits

In this second approach to identify promoters in tomato fruits, we use known promoters from Arabidopsis [6]. The Arabidopsis genome is the most studied genome within plants and therefore much is known about it: its genome, genes, promoters, regulatory networks, etc. Therefore, many plant researchers use the Arabidopsis genome information as reference.

Fig. 8 outlines the algorithmic steps of our approach. In a nutshell, we search our dataset for 213 known Arabidopsis promoters obtained from PLACE database [6]. The top 20 most frequent promoters of lengths 4 to 10 are saved for the next steps. In this way a set of 120 candidate promoters is obtained. To quantify the statistical significance of such sheer frequencies, we randomly shuffle the nucleotides in our dataset and compute the occurrence frequency of each of the 120 candidates identified in the previous step. We repeat this randomization step for 10 times and we average the frequencies of an individual promoter over the 10 runs. In the final step we keep the most likely promoter candidates by applying a filter of 3 fold difference between the frequency and the shuffled frequency of a given motif. For instance if a motif is found 24 times in the original dataset and on average 22 times in the shuffled dataset, it is not likely a true motif. But if it is found only 8 times (or less) in the shuffled case, than it is more likely to be a true motif worthy of a further more thorough analysis.

```
1. For motif length l = 4 to 10 do
    a. Compute frequency f of each Arabidopsis promoter of
       length l found in our dataset
    b. Keep top 20 most frequent promoters of length l found
    c. For i = 1 to 10
           i. Shuffle the dataset
          ii. Compute frequency for each of the top 20 most
              frequent promoters from step 1.b.
    d. End for
    e. $f_{shuffled}$ = average of the shuffled frequencies computed
       at step 1.c.ii
    f. Select promoter candidates satisfying $f/f_{shuffled}$ >=3
2. End for
```

Fig. 8 Pseudocode for the algorithm used to find known Arabidopsis promoters in tomato fruit genome.

Using the algorithm outlined in Fig. 8, nine potential motifs are identified. They are listed in Table 2 together with their true occurrence frequencies and the shuffled frequencies. The fifth column of the table shows how many genes share the corresponding promoter. This column does not add up to 396, the number of genes in the dataset, because one or more of the 9 promoter candidates can be present in a given upstream gene. In addition, a promoter candidate may occur once or more within a gene's upstream region. The morning motif CCCACC identified by MEME software is not present in Table 2 because it is not part of the Arabidopsis pool.

Promoter candidate	Frequency f	Frequency $f_{shuffled}$	$f/ f_{shuffled}$	No. of genes having this motif
CACGTG	107	21	5.1	85
CAAACAC	64	20	3.2	56
ACGTGTC	31	7	4.5	28
ACGTGGC	26	3	8.1	26
GAGTGAG	20	7	3.0	19
TTTTTTCC	59	20	3.0	55
AAAATATCT	59	16	3.6	48
TCTCTCTCT	21	1	21.0	14
CTATAAATAC	11	3	3.8	11

Table 2. Nine likely promoters identified in our upstream regions of the 396 differentially expressed genes.

Analysis of the evening motif AAAATATCT

Next we analyze in more depth one of the nine potential motifs identified above. The AAAATATCT motif (see row 8 of Table 2) is also known in the literature as the evening motif. As Table 2 shows, the evening motif is found in 48 upstream genes: in one upstream is found three times; in nine upstreams is found twice; and in the rest of 38 upstreams is found only once. Fig. 9 shows the upstream position where this motif is found in the 48 genes having this motif. Fig. 10 shows the distribution of the 48 genes having the evening motif within the 8 fold-expression groups defined in Table 1. An interesting result is that this motif upregulates 44 genes and downregulates only 5.

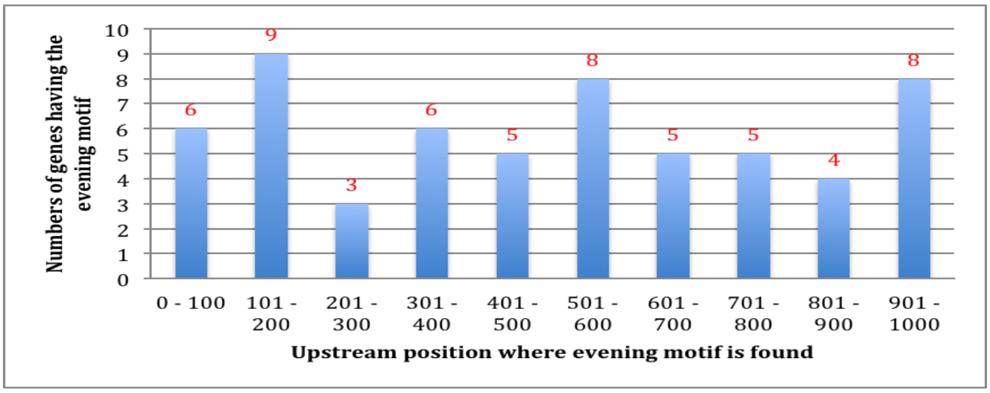


Fig. 9 Upstream position where the evening motif AAAATATCT is found in the 48 genes having this motif.

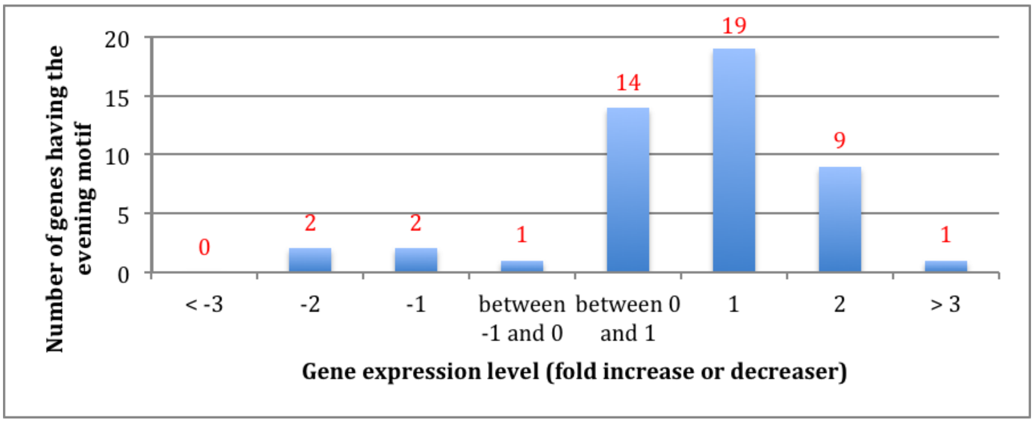


Fig. 10 Distribution of the 48 genes having the evening motif AAAATATCT within the 8 fold-expression groups defined in Table 1.

3 Conclusions and future work

We present an approach to identifying Arabidopsis known promoters in another genome (here the tomato fruit genome). Nine potential promoters in tomato fruit are identified with our method. They can be further validated by more sophisticated, but also more expensive and time-consuming, in-lab methods. Given that these nine promoters are true promoters in Arabidopsis and that they are found in the tomato genome by our algorithm, they are very likely to be true motifs, which means that they potentially regulate the expression of some of the 396 differentially expressed genes in the wild and ovate mutation tomato varieties.

One disadvantage of our method described in section 3.2 is that it will not identify promoters unknown yet (in our case motifs that exist in the tomato genome but do not exist in Arabidopsis). However, our approach can be used in combination with promoter predictors to discover both known and unknown yet promoters. Namely, we propose the following two-step approach: (1) using first the algorithm in section 3.2 to identify the Arabidopsis promoters present in the tomato genome and remove from the pool those genes having these promoters; (2) using promoter predictors in the remaining pool of genes. In this way a predictor will find a better promoter candidate with an E-value unbiased by too many genes not sharing the new motif. Of course, this practical approach of finding promoters can be generalized to any set of genes (not only tomato genes) using other known promoters (not only Arabidopsis promoters).

The research presented here shows that promoter prediction software does not work well if one searches for motifs in large sets of genes that, more likely, are regulated by several different motifs. Therefore, we hypothesize that clustering genes that might share a common motif will improve the prediction ability of the software. Here we clustered genes accordingly to their fold expression level, but the predictors found a likely promoter in only one of the eight groups. In future we plan to investigate another clustering: cluster the upstream regions according to similarly expressed genes. For example, we will cluster genes with similar expression patterns over a time period (also known as co-regulated genes), and focus on finding potential motifs in each such cluster.

Acknowledgement: This paper is founded from the research Grant NSF DBI-0922661(60020128).

References

[1] G.R. Rodriguez, S. Munos, C. Anderson, S.C. Sim, A. Michel, M. Causse, B.B. McSpadden Gardener, D. Francis, and E. van der Knaap. *Distribution of SUN, OVATE, LC, and FAS in the Tomato Germplasm and the Relationship to Fruit Shape Diversity*. Plant Physiol 156: 275-285. 2011.

[2] NSF Grant (3.7 mil.) for *"Discovery of Genes and Networks Regulating Tomato Fruit Morphology"* (PI: E. van der Knaap, Co PI: S. Visa, B. McSpadden Gardener, C. Catala), 2009-2013.

[3] O. Bembom, S. Keles, M.J. van der Laan (2007). *Supervised Detection of Conserved Motifs in DNA Sequences with cosmo.* Statistical Applications in Genetics and Molecular Biology: Vol. 6: Iss. 1, Article 8. http://www.bepress.com/sagmb/vol6/iss1/art8, [Accessed on July 2012].

[4] *MEME*, http://meme.nbcr.net/meme/, [Accessed on July 2012].

[5] Tobias Marschall and Sven Rahmann. *Efficient exact motif discovery*. Bioinformatics (Proceedings of ISMB), 25(12):i356–364, 2009. DOI: 10.1093/bioinformatics/btp188, [MoSDi accessed on July 2012].

[6] *A Database of Plant Cis-acting Regulatory DNA Elements*, http://www.dna.affrc.go.jp/PLACE/ [Accessed on July 2012].

[7] K. Higo, Y. Ugawa, M. Iwamoto and T. Korenaga (1999) *Plant cis-acting regulatory DNA elements (PLACE) database:1999*. Nucleic Acids Research Vol.27 No.1 pp. 297-300.

[8] O. Bembom, S. Keles, M.J. van der Laan. *Supervised Detection of Conserved Motifs in DNA Sequences with cosmo.* Statistical Applications in Genetics and Molecular Biology: Vol. 6: Iss. 1, Article 8, 2007. http://www.bepress.com/sagmb/vol6/iss1/art8.

[9] Timothy L. Bailey, Mikael Bodén, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, William S. Noble. *MEME SUITE: tools for motif discovery and searching*. Nucleic Acids Research, 37:W202-W208, 2009.

[10] B. Langmead, C. Trapnell, M. Pop and S. L. Salzberg. *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome*. Genome Biology. Vol. 10 (3), 2009. (tophat.cbcb.umd.edu).

[11] S. Anders, W. Huber. *Differential expression of RNA-Seq data at the gene level the DESeq package.* European Molecular Biology Laboratory (EMBL), Heidelberg, Germany, 2012. (bioconductor.org/packages/2.11/bioc/vignettes/DESeq/inst/doc/DESeq.pdf).

[12] P.J. Cock, T. Antao, J.T. Chan, B.A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M.J. de Hoon. *Biopython: freely available Python tools for computational*

molecular biology and bioinformatics. Bioinformatics 2009 Jun 1; 25(11) 1422-3. doi:10.1093/ bioinformatics/btp163 pmid:19304878. PubMed HubMed (biopython.org).

Michelle BLACKWOOD
The College Of Wooster
Department of Mathematics and Computer Science
1189 Beall Avenue
Wooster, OH, 44691
U.S.A.
E-mail: mblackwood14@wooster.edu

Sofia VISA
The College Of Wooster
Department of Mathematics and Computer Science
1189 Beall Avenue
Wooster, OH, 44691
U.S.A.
E-mail: svisa@wooster.edu

Esther VAN der KNAAP
The Ohio State University
Department of Horticulture and Crop Science
204 A Williams Hall
Ohio Agricultural Research and Development Center (OARDC)
1680 Madison Avenue
Wooster, Ohio, 44691
U.S.A.
E-mail: vanderknaap.1@osu.edu

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Weekly TV Show Helper 3: A utility application for tracking TV show broadcasts

Bogdan-Andrei Blăniță, Ana-Maria Barti
Teacher Coordinator: Anca Vasilescu

Abstract

Weekly TV Show Helper 3, or WTvSH3 for short, is a desktop application designed to help the television series-watching user to keep track of information regarding the last-watched episodes of each series. People that enjoy viewing several television programmes will find it hard at some point to remember what they saw last from each show. This is when our application becomes useful: its efficient algorithms and data structuring, pleasant aspect and user-friendly interface, along with its easily-accessible data management and statistical features allow even computer novices to never miss an episode from that new sitcom, and to never forget when the last episode of their favorite soap opera was broadcast.

1 The motivation for our choice

In time, there has been increasing popularity and diversification of television series, and an equal increase of their audience. Nowadays, there are many people that enjoy watching several series weekly, and even daily. It then becomes difficult, when watching for instance 2-3 shows each day, to keep track what the last episode of each show is. Unlike some websites that allow users to browse long listings of programmes which may or may not be of interest, our application provides a simple offline interface that will aid greatly in tracking what was last watched only from certain shows. We, the developers of WTvSH3, are ourselves part of this public, having already developed two less-functional and more limited applications of this type for own personal needs. That is why, we also successfully use it and we can vouch for its usefulness.

2 About WTvSH3

2.1 Technology used

The application was developed using Embarcadero Delphi XE2, which is an IDE for console, desktop graphical, web and mobile applications. Its compilers generate 32- and 64-bit (for Windows) and 32-bit (for Mac OSX and iOS) native code from a dialect of Object Pascal. Delphi was originally developed by Borland as a rapid application development tool for Windows, and as the successor of Borland Pascal. Delphi and its C++ counterpart, the C++Builder, shared many core components, notably the IDE and VCL, but remained separated until the release of RAD Studio 2007 (Rapid Application Development; a shared host for Delphi, C++Builder and others). In 2006, Borland's developer tools section was transferred to a subsidiary known as CodeGear, which in turn was sold to Embarcadero Technologies in 2008[1].

2.2 System requirements

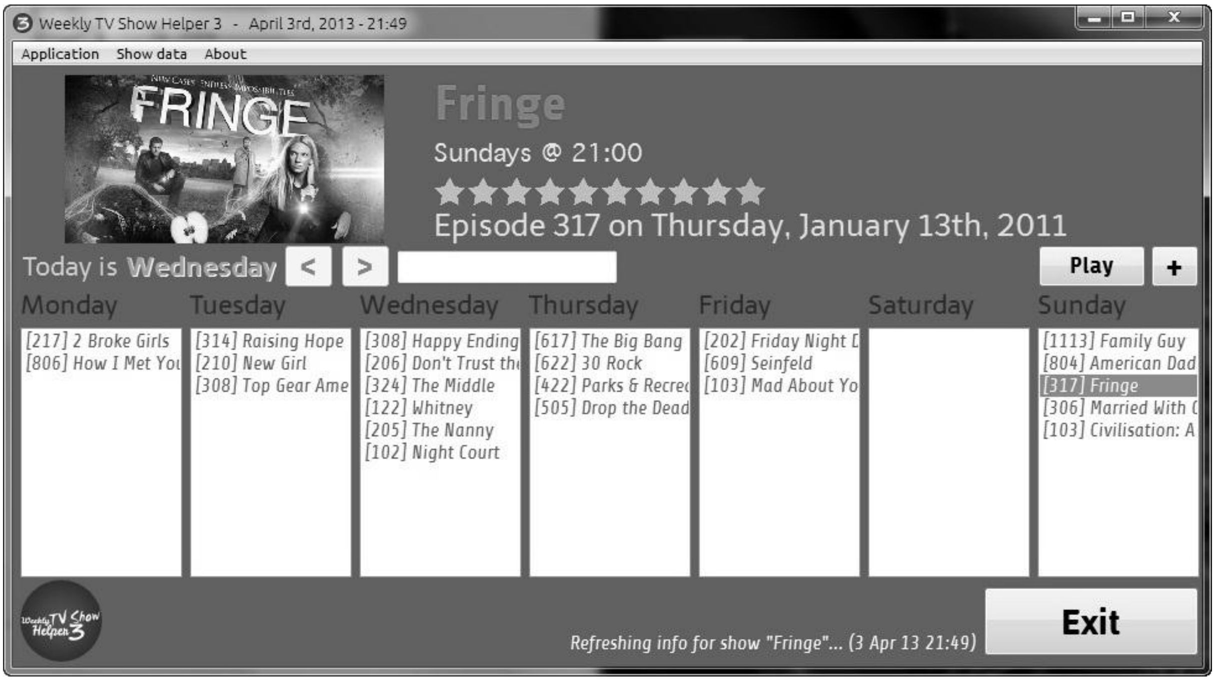
The system requirements for this software are relatively low-cost:

- Compatible Operating System (Windows XP/Vista/7)
- 1.2 GHz Processor
- 256 MB RAM
- 50 MB free on HDD for base application

3 Application execution/run

3.3 The main window

The interface was designed to be as intuitive and user-friendly as possible. It has many sections, as follows. It displays in seven boxes, in the middle of the form, the shows that air in the selected days. There is a section at the top which is used to show some details about the specified show. We also added several buttons that act as shortcuts to the main menu functions, by which, in turn, all features available in the program can be accessed.



Fringe

Sundays @ 21:00

★★★★★★★★★★

Episode 317 on Thursday, January 13th, 2011

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
[217] 2 Broke Girls	[314] Raising Hope	[308] Happy Ending	[617] The Big Bang	[202] Friday Night L		[1113] Family Guy
[806] How I Met You	[210] New Girl	[206] Don't Trust th	[622] 30 Rock	[609] Seinfeld		[804] American Dad
	[308] Top Gear Ame	[324] The Middle	[422] Parks & Recrea	[103] Mad About Yo		[317] Fringe
		[122] Whitney	[505] Drop the Dead			[306] Married With C
		[205] The Nanny				[103] Civilisation: A
		[102] Night Court				

Refreshing info for show "Fringe"... (3 Apr 13 21:49)

Image 1. Main window

3.1 Data organization

Information regarding the shows is stored in an XML file, in the “\data” folder within the program directory, as well as other files that will be used during execution. Photos, theme music files and themes are stored in sub-folders named accordingly.

```
type TShow=record
  name: string;
  dow, broadcast_hour, rating: byte;
  last_episode_season, last_episode: word;
  last_episode_date: TDate;
  last_episode_modified: TDateTime;
  active: boolean;
end;
```

3.2 Launching the application

When the application is started, several global variables are initialized, visual components are dynamically created, the data is read from de XML files, then the main form is shown and the graphical theme is applied. The code below is used to proceed in doing exactly that.

```
//creating the main form
procedure TF1.FormCreate(Sender: TObject);
var i: word;
begin
  with FormatSettings do
        begin DecimalSeparator:='.'; ThousandSeparator:=','; DateSeparator:='/'; TimeSeparator:=':' end;
  assignfile(f, 'data\log.txt'); rewrite(f); closefile(f);
  bgImg.SendToBack; bgImg.AutoSize:=true;
  if fileexists('data\bgImg.png') then
        bgImg.Picture.LoadFromFile('data\bgImg.png');
  mpl.TimeFormat:=tfMilliseconds;
  //
  for i:=1 to 7 do
    begin
      lday[i]:=TLabel.Create(Self); lday[i].Parent:=Self; lday[i].Caption:='lday'+inttostr(i);
    end;
  //
  for i:=0 to 7 do
    begin
      lb[i]:=TListBox.Create(Self); lb[i].Parent:=Self;
      lb[i].OnClick:=lb_click; lb[i].OnDblClick:=lb_dbl_click; lb[i].OnMouseDown:=ListBox1MouseDown
    end;
  lb[0].Visible:=false;
  //
  for i:=1 to 10 do
    begin
      img[i]:=TImage.Create(Self);        img[i].Parent:=Self;        img[i].Center:=true;        img[i].Proportional:=true;
img[i].Stretch:=true;
      img[i].Picture.LoadFromFile('data\starG.png'); img[i].Cursor:=crHandPoint;
      img[i].OnClick:=starClick;
      img[i].Picture.RegisterFileFormat('.PNG', 'Portable Network Graphics', TPNGImage)
    end;
  imag.Picture.LoadFromFile('data\logo2.png');
  init_read_data;
end;
```

3.3 Interface application content

3.3.1 The list boxes

The boxes contain the name of the shows that are broadcast each day and other information about the series. When an item from these boxes is clicked, detailed information is shown on the top of the form including a picture with the chosen series which can be added by the user. As well, the play button is enabled above the boxes, and can play the soundtrack of the show (also added by the user), but only if the file exists.

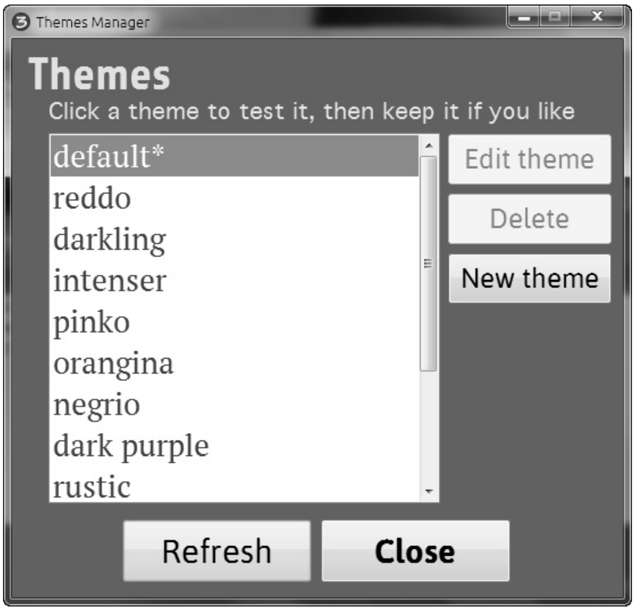
3.3.2 The menu

The menu bar contains three drop-down menus called "*Application*", "*Show data*" and "*About*". Each one provides access to functions designed to help users in the shows management process.

The "*Application*" item contains tasks like "*Open*" which can offer users easier acces to multiple folders within the program directory; the "*Appearance*" feature can be used to make the interface more attractive by allowing the user to change the graphical theme (colors and fonts) of the forms in a user-friendly manner; it is also possible to maximize the main window on the entire screen.

Features within the „*Show data*" menu item include: refreshing the information in the lists, filtering shows by name, selecting the sorting criteria, selecting a show randomly, managing theme music options, exporting information about the data into a file, links to website search queries, backing up the data, and a few general display options.

The "*About*" item gives access to statistics and the help section, which should aid users in understanding how this program works.



Click a theme to test it, then keep it if you like

Image 2. Graphical theme manager window

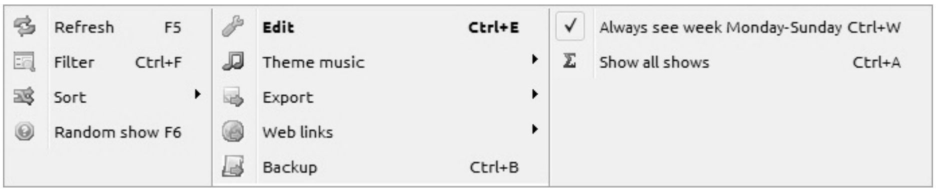


Image 3. The „*Show data*” menu

The procedure below is used to open the „*\photos*” program folder:

```
procedure TF1.Photosfolder1Click(Sender: TObject);
begin
      ShellExecute(Handle, 'open', PChar('data\photos'), nil, nil, SW_SHOW)
end;
```

The following code snippets open links to various webpages:

```
procedure TF1.GoogleSearch1Click(Sender: TObject);
begin
      ShellExecute(Handle, 'open',
      PChar('http://www.google.ro/search?sourceid=chrome&ie=UTF-8&q='+web_str(s[sel.show].name)),
      nil, nil, SW_SHOW)
end;

procedure TF1.Wikipedia1Click(Sender: TObject);
begin
      Shellexecute(Handle, 'open',
      PChar('http://en.wikipedia.org/w/index.php?title=Special:Search&search='+web_str(s[sel.show].name)),
      nil, nil, SW_SHOW)
end;

procedure TF1.IMDb1Click(Sender: TObject);
begin
```

```
        Shellexecute(Handle, 'open', PChar('http://www.imdb.com/find?q='+web_str(s[sel.show].name)+'&s=all'),
        nil, nil, SW_SHOW)
end;
```

This procedure is called to display information about a randomly selected show from the database:

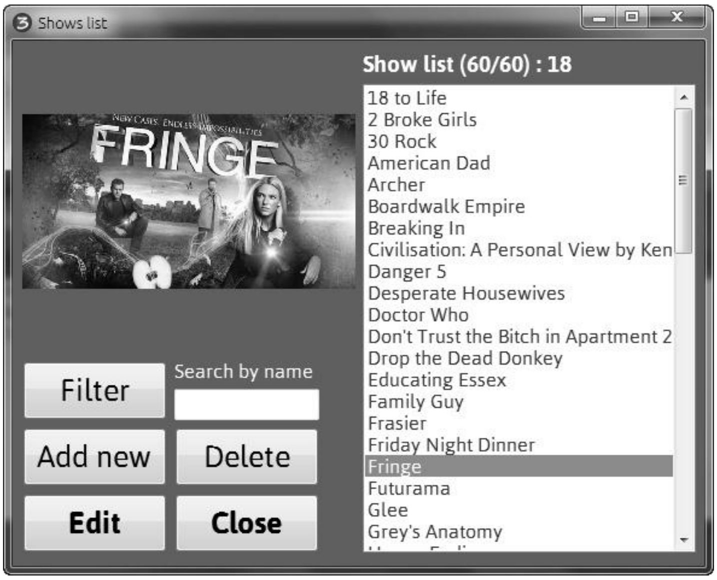
```
procedure TF1.Showinfoforrandomshow1Click(Sender: TObject);
var i: byte;
begin
        sel.show:=random(ns)+1; sel.day:=s[sel.show].dow;
        ref_info_bClick(FEd)
end;
```

When toggling the full-screen option, the next code is executed:

```
procedure TF1.Togglefullscreen1Click(Sender: TObject);
begin
        if F1.BorderStyle=bsSizeable then
                begin F1.BorderStyle:=bsNone; F1.WindowState:=wsMaximized end
        else
                begin F1.BorderStyle:=bsSizeable; F1.WindowState:=wsNormal end;
        F1.OnResize(Togglefullscreen1)
end;
```

3.3.2 Editing / adding data

The editing/adding data function is accessed from the "*Show data*" menu item, by choosing the "*Edit*" button. On click it displays a form which contains a list of shows on the right, a picture with the selected series on the left, and five options below. The filter is used to only show the series that you want (depending on whether they are active or having missing information, in addition to searching by name). From this form the user can easily edit show data (this opens a new window) or images (view details, add/change, or even search online information based on the name of the show).



18 to Life
2 Broke Girls
30 Rock
American Dad
Archer
Boardwalk Empire
Breaking In
Civilisation: A Personal View by Ken
Danger 5
Desperate Housewives
Doctor Who
Don't Trust the Bitch in Apartment 2
Drop the Dead Donkey
Educating Essex
Family Guy
Frasier
Friday Night Dinner
Fringe
Futurama
Glee
Grey's Anatomy

Image 4. The show list window

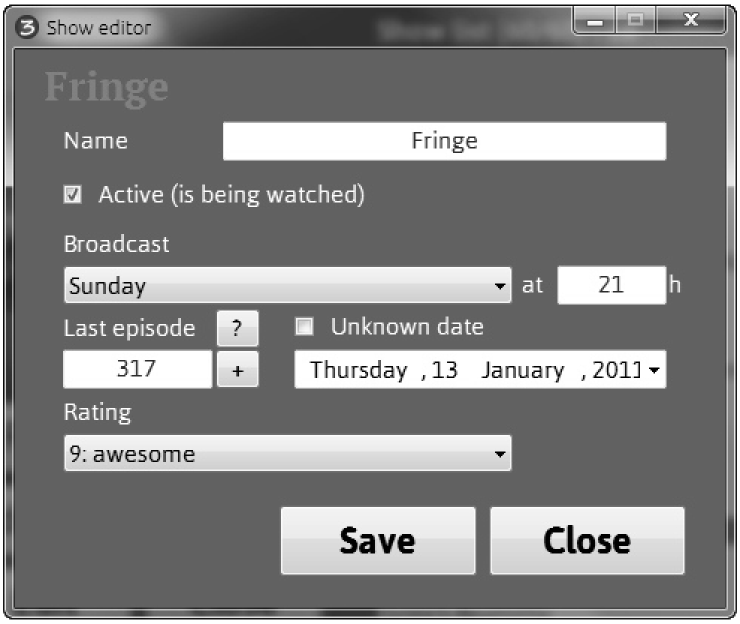
When adding a new show or editing an existing one, the „*Show editor*” form is displayed. This form is simple to fill out; the user has to choose if the show will appear in the list boxes (the „*Active*” property), the day and hour of broadcast, then information about the last episode watched and its date. Finally, the show can be rated (this is purely for statistical purposes), and the data could be saved.

The code below is executed when adding a new show to the database:

```
procedure TFS.add_bClick(Sender: TObject);

var x: string;

begin
  try
  if InputQuery('Add a new show', 'Type in the name of the show:', x) then
    begin
        inc(ns); setlength(s, ns+1);
        sel.show:=ns; sel.day:=0;
        s[ns].name:=x;
        s[ns].dow:=0;
        s[ns].broadcast_hour:=0;
        s[ns].rating:=0;
        s[ns].last_episode_season:=0;
        s[ns].last_episode:=0;
        s[ns].last_episode_date:=dateX;
        s[ns].active:=true;
        FEd.Show
    end
  except on E:Exception do
        showmessage(e.ClassName+' raised an error with the message '''+e.Message+'''. That''s not right!')
  end
end
```

When saving the data, the following code is used:

```
procedure save_data;
var i: word;
begin
  try
        log('Saving data (XML shows)...');
sx.Clear;
sx.Root.NodeName:='WTvSH3_Show_Database';
sx.Root.SetAttribute('DateSaved', formatdatetime('dddd, mmmm d, yyyy, hh:nn', now));
  for i:=0 to ns-1 do
    begin
        sx.Root.AddChild('show');
        sx.Root.ChildNodes[i].SetAttribute('ID', inttostr(i+1));
        sx.Root.ChildNodes[i].SetAttribute('active', booleantotruefalse(s[i+1].Active));
      with sx.Root.ChildNodes[i] do
        begin
          if s[i+1].last_episode_modified<>dateX then
                SetAttribute('LEp_modified', formatdatetime('dd.mm.yyyy hh:nn', s[i+1].last_episode_modified));
          AddChild('name');
          Find('name').Text:=s[i+1].name;
          AddChild('day_of_week');
          Find('day_of_week').Text:=inttostr(s[i+1].dow);
          AddChild('broadcast_hour');
          Find('broadcast_hour').Text:=inttostr(s[i+1].broadcast_hour);
          AddChild('last_episode');
          Find('last_episode').Text:=inttostr(s[i+1].last_episode);
          if s[i+1].last_episode_date<>dateX then
            begin
        AddChild('last_episode_date');
        Find('last_episode_date').Text:=formatdatetime('dd/mm/yyyy', s[i+1].last_episode_date)
      end;
     AddChild('rating');
     Find('rating').Text:=inttostr(s[i+1].rating)
    end;
  end;
 sx.SaveToFile('data\shows.xml')
 except on E:Exception do showmessage('An error was encountered while saving XML
data.'+dnl+E.ClassName+' error raised, with message : '+E.Message) end
end;
```



Show editor

Fringe

Name: Fringe

☑ Active (is being watched)

Broadcast: Sunday at 21 h

Last episode: 317

□ Unknown date

Thursday , 13 January , 2011

Rating: 9: awesome

Image 5. The show editor window

4 Conclusions and future development

The application has been developed to meet the needs of a public that is, after all, quite specific. Therefore, it is expected that it will not be of much interest to persons not part of this public. Yet for those who are a part, it could prove to be of real help with the issues previously described in this document. We consider our attempt of solving the announced issues to be quite successful, as the program has already proven to be useful.

The most difficult issues we've faced during development are related to structuring the data, as well as streamlining the connections between the interface objects, so that the program functions properly at all times. This has cost us many programming hours, but it assures a stable application.

For the near future, we are taking into consideration to implement the automatic update of show details following the internet information regarding the last watched episode of each show – such as the broadcast date and the episode title, when it is saving.

Besides, it may be also convenient for users to add a feature for allowing an even easier management of photo and music files. Overall, our WTvSH3 application serves its purpose well as it is, and it will simply become more and more attractive with any improvement it receives.

5 References

[1] Wikipedia, http://en.wikipedia.org/wiki/Embarcadero_Delphi
[2] Camelia Berceanu and Elena Toader. *Inițiere in Borland Delphi*. Editura Arves, 2003.
[3] Mihai Oltean and Crina Groșan. *Delphi 7.0 în 200 de aplicații*. Editura Albastră, 2004.
[4] Delphi Basics, reference information for the Delphi programming language. http://www.delphibasics.co.uk/

Bogdan-Andrei BLĂNIȚĂ
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50, Brasov, 500091
Romania
E-mail: bogdybba@gmail.com

Ana-Maria BARTI
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50, Brasov, 500091
Romania
E-mail: barti.anamaria@ymail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Digital Audio Steganography Algorithm

Alexandru Cristian Bozdog
Teacher Coordinator: prof. univ. dr. ing. Mircea Vladutiu, dr. ing. Septimiu Fabian Mare

Abstract

Steganography is the science of hiding information within other information that is not readily detectable by parties aside from the intended recipient. The present article will describe our research into digital audio steganography, along with an algorithm to further develop the current state of the art, expanding upon it to increase the amount of data that can by hidden in a stego audio file, while the output audio file is still perceptually indistinguishable from the original file. The current paper will also show some of the issues of audio steganography algorithms, and attempt to find solutions for part of these problems.

Keywords: Steganography, Stego-audio, Cover audio.

1 Introduction

Steganography is the art and science of hiding not only the message, but also the occurrence of a communication. It is a form of security through obscurity. It has obvious applications in security purposes, and in the February 2000 issue of USA Today [1], there has been a report about steganography being used by terrorists hiding secret information in pictures of posted items on widely used websites such as Amazon or Ebay. Other, less widely spread uses, could include linking important information such as a description of a digital item to the item itself, without damaging it, for example, compiling the doctor's diagnosis to an x-ray image, or description of a song playing on the radio within the actual song.

2 Background

Since the creation of the Personal Computer, steganography has evolved from a physical form to a more modern digital one. Computational devices have also evolved, such as that nowadays, internet connected handheld devices overwhelm in number the amount of personal computers that have internet access, a thing presented in a recent article written by Cynthia Harvey [2]. In this age of almost permanent inter-connectivity, privacy and security are extremely hard to manage and maintain.

In today's digital age, information security is widely considered synonym with cryptography. Ross Andersen [3] describes cryptography as the intersection between security engineering and mathematics; but a security system that uses cryptography to encode and secure data is considered obsolete when the processing power required to reverse engineer the protected data with is obtained. In the recent years, a trend has developed that tries to hide what cryptography alone cannot: the existence of secure information exchange actually taking place. This trend is called steganography. Digital steganography represents the use of the science of steganography for hiding secret information (text, data) inside other digital content. The older, physical embedding techniques prove limited in the comparison amount of data that can be embedded or the wide array of carriers to choose from presented by digital steganography.

2.1 Information Hiding

Probably the best classification that is available for steganography is the one by P. Pfitzman [4] and P. Petricolas [5], in which it is being classified as a member of the information hiding domain, together with watermarking, covert channels and anonymity methods. This is probably the most adequate classification for it because it considers both digital and traditional methods together in one dependency tree. As a member of this large research field, steganography covers some aspects of security that cryptography is unable to: hiding the existence of data.

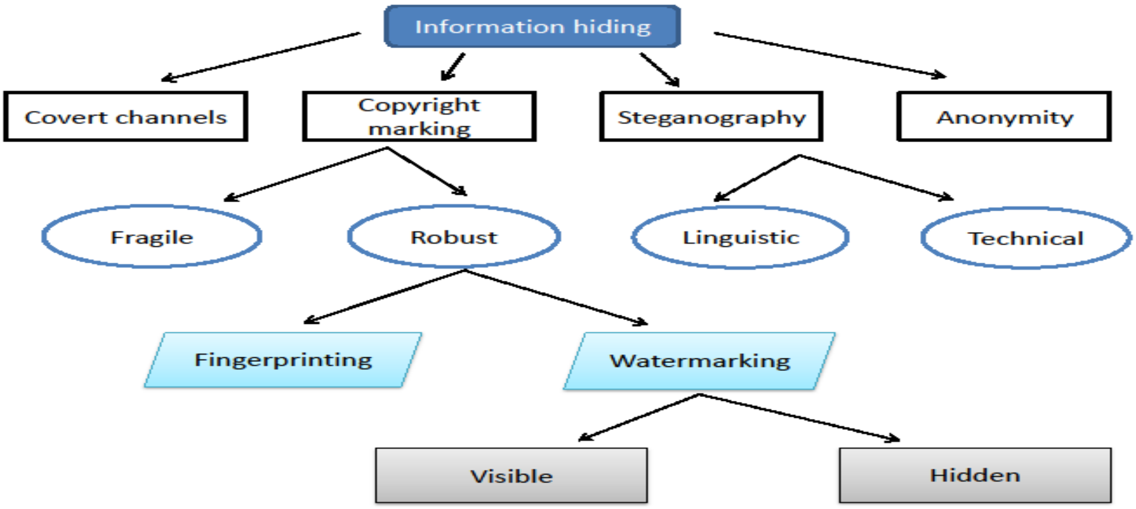


Fig. 1: Steganographic carrier classification (extended from [4][5])

In digital terms, steganography implies using digital formats as storage spare for secret information. The choice of the carrier varies, but some of the typically used categories of digital cover files are as follows: images, videos, audio, archives, executable and libraries, documents and cyphertext. This list is non-exhaustive and contains only the most used formats. In its most basic description, steganography can be used on any format that has overhead data that can be manipulated without affecting the statistical or logical integrity of the input file. Some of the most widely used digital steganography techniques include LSB(Least Significant Bit) embedding, Echo editing, packet delay injection, including data in ignored parts of files, such as after the logical end of the carrier, pictures embedded in video, and redundancy exploitation in the targeted instruction set of executable files. These is also content-aware steganography, which uses semantics to hide the information, therefore offering increased security against non-human adversaries.

Although considered a separate research direction, watermarking is related to steganography. Both are used for information hiding, the main difference lying in the scope and usage. Watermarking refers to a way of hiding information in a carrier so that information is robust to modification, thus removing the watermark without damaging the carrier should prove impossible. This also describes the difference between watermarking and steganography: the latter refers to hiding fragile information that might be destroyed upon modification of the carrier. Another important such difference is that while steganography requires the data to be invisible to the presence of a third party viewer, for watermarking this is optional.

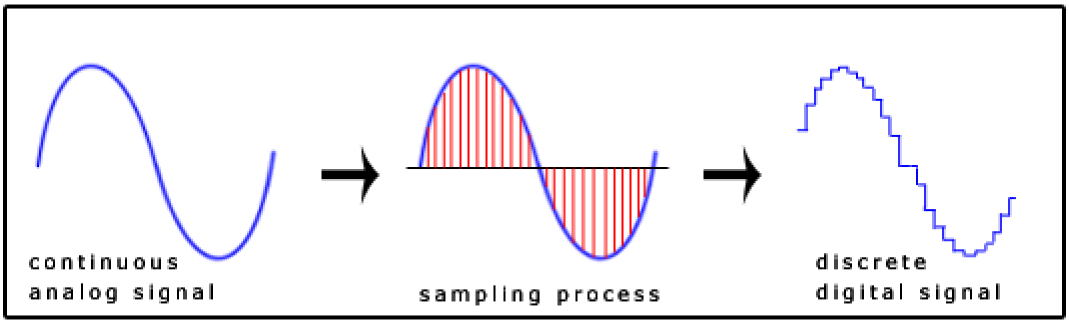
The goal of steganalysis is to figure out if a suspect file has embedded secret data in them, usually by employing statistical analysis, and estimating the chance of hidden data existence.[6][7]. An individual using steganalysis to determine if a carrier contains secret information would require samples that are known to be clean from the same source as the inspected set (photos from the same type, or even same camera for example), and analyse them for various statistics. Therefore, every steganographic algorithm is only as strong as its resistance to advanced detection algorithms [8]. As well as determining if a carrier has embedded data, steganalysis also has another purpose: to benchmark newly developed steganographic techniques.

3 Digital audio steganography

In a computer-based audio steganography system, secret messages are embedded in digital sound. The stego-data is embedded by slightly altering the binary sqence of the carrier. Using digital sound as a carrier is usually a more difficult process than embedding data in other formats, such as images. In order to conceal messages successfully and hide them from detection algorithms employed by steganalysis, a variety of methods for embedding data have been developed.

3.1 Digital audio

Digital audio differs from traditional analogue sound by being a discrete rather than a continuous signal. A discrete signal is created by sampling a continuous signal at a specified rate. The standard rate for sampling an audio CD is 44.1 kHz. The Nyquist-Shannon sampling theorem [9][10] states that perfect reconstruction of a signal is possible only when the sampling rate is greater than double the maximum frequency of the signal being sampled. The entire range of human hearing goes between 20 hz and 20 kHz, therefore 44.1 kHz is the minimum required sampling rate to satisfy the theorem. The sampling process is emphasized in figure 2, however, normally, sampling rates are set at a level where the discrete signal is visually indistinguishable from the original continuous one.

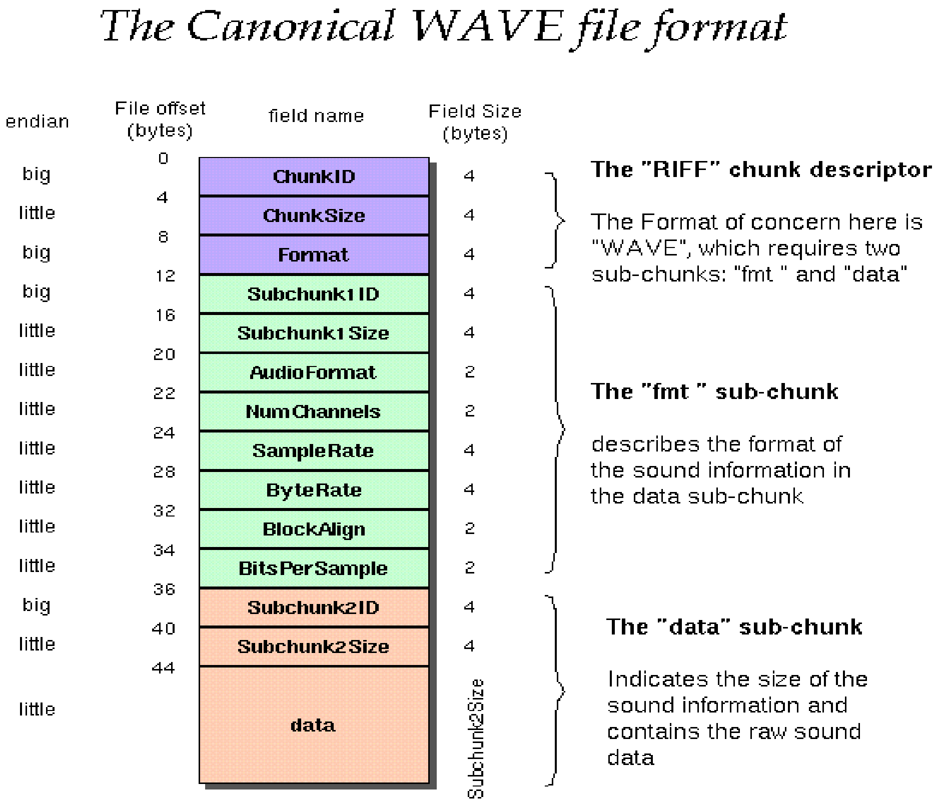


.

Fig. 2: Sampling process

3.2 WAV file format

Since steganographic mediums are robust to change and therefore any damage to the carrier can destroy the information, steganographic algorithms usually employ uncompressed file formats. Waveform audio file format (WAVE, more commonly known as WAV), is such a format. It is an application of the Microsoft Resource Interchange File Format (RIFF), therefore it stores data in "chunks", and is the format usually used by Windows for raw and uncompressed data. A RIFF file is a tagged format, has specific chunks that each include tags and the size of the chunk in bytes. The tags specify how the data should be interpreted. The advantage in such a file format is that it can easily be extended while still being backwards compatible. The rule of thumb for such a format is that it should ignore any tags it does not recognize. The reader will not be able to use the new information, but the old one will still be available.



endian	File offset (bytes)	field name	Field Size (bytes)
big	0	ChunkID	4
little	4	ChunkSize	4
big	8	Format	4
big	12	Subchunk1 ID	4
little	16	Subchunk1 Size	4
little	20	AudioFormat	2
little	22	NumChannels	2
little	24	SampleRate	4
little	28	ByteRate	4
little	32	BlockAlign	2
little	34	BitsPerSample	2
big	36	Subchunk2ID	4
little	40	Subchunk2Size	4
little	44	data	Subchunk2Size

The "RIFF" chunk descriptor

The Format of concern here is "WAVE", which requires two sub-chunks: "fmt " and "data"

The "fmt " sub-chunk

describes the format of the sound information in the data sub-chunk

The "data" sub-chunk

Indicates the size of the sound information and contains the raw sound data

Fig. 3: The canonical WAV file format [10]

The WAV file format is limited to 4 GB of data because of its 32-bit unsigned integer used to record the file size header. This is equal to around 6.8 hours of cd-quality audio, which is sampled at 44.1 kHz (meaning 44100 samples per second, in 16-bit stereo mode). 10 minutes of the same cd-quality audio occupies 10.9 MB of data. The big size of the format is due to the fact that is the only lossless encoded audio file format.

Lossless compression algorithms are able to decompress the original data 1:1 without any loss of information. Lossy compression algorithms, which cannot reconstruct the original data with 100% accuracy, are able to reduce the size of the file more, by eliminating all the overhead data based on the fact that the human hearing cannot distinguish every bit from each sample. Comparing these two methods highlights that the obvious starting place for a steganographic algorithm is represented by the overhead digital information that cannot be perceived, and since the lossy compressed file formats (such as MP3) have this data removed, it is obvious the file format used for a steganographic algorithm would have to be WAV.

3.3 Audio steganography embedding methods

Digital audio steganography employs different methods for hiding information. One of the most commonly used methods is LSB insertion. Others are parity coding, spread spectrum, echo hiding and phase coding. Some of these embedding techniques require prior knowledge of signal processing, Fourier analyses, or other high level mathematics.

3.3.1 LSB and Parity coding

By substituting the LSB of each sample from the original carrier with a bit from the hidden data, LSB coding allows for a large amount of information to be transported secretly, since in an idea form where all samples are used, the transmission rate is 1 kbps for 1 kHz.. As opposed to this, parity coding does not evaluate each sample, but series of samples, and encodes the secret information in the parity bits. If the parity bit is not the same as the bit that is to be encoded, the process flips the LSB of one of the samples in the region. The issue with both these methods lies in the sensitivity of the human year. Although parity coding goes a long way towards alleviating this concern in comparison to the LSB coding, it can hide a smaller amount of data.

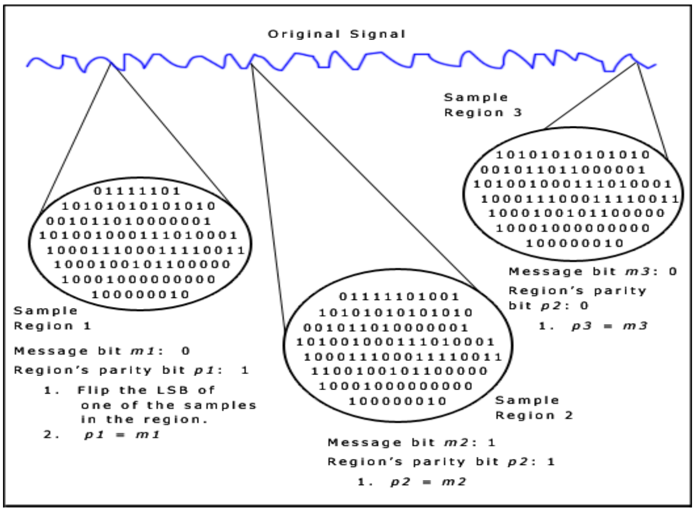


Fig. 4: Parity coding embedding method [11]

3.3.2 Phase coding

Phase coding relies on the fact that the phase component of the human hearing is not as perceptive as noise is, and thus tries to remove the noise issue that comes up with LSB or Parity coding. It tries to achieve this by encoding the message in bit shifts in the phase spectrum, rather than introducing perturbations. The method breaks down the audio file into samples that are equal in length to the secret data, and uses a Discrete Fourier Transform function on these samples to create a matrix of the phases and Fourier magnitudes. It then uses this matrix to encode the secret data. Disadvantages of this technique are a low transmission rate, and the fact that for decoding, the user would have to know the segment length

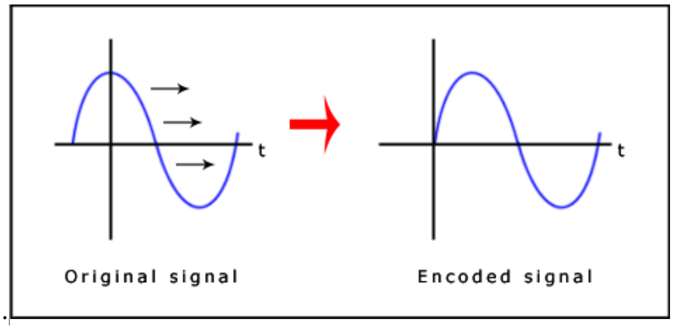


Fig. 5: Phase coding embedding method [11]

3.3.3 Spread spectrum

The basic spread spectrum method attempts to spread the secret information across the frequency spectrum of the digital audio as much as possible. This can be compared to a LSB coding that spreads the bits randomly across the samples, but with this method, the message is spread over the frequency spectrum, using a code with no connection to the actual signal. Two versions of SS have been developed: direct sequence, and frequency hopping. Direct sequence spreads the message by a constant, and then modulates it with a pseudorandom signal. Frequency hopping, on the other hand, the audio file's frequency itself is modified so that it alternates quickly between frequencies. This method can bring an advantage towards the transmission rate, and robustness of the data, but much like other techniques, it also might introduce a noise in the sound. The mathematic principles involved in this technique are extensive, and can be observed in Katzenbeisser's and Petitcolas direct sequence spread spectrum system [12].

3.3.4 Echo hiding

Echo hiding embeds the secret information by introducing an echo into the discrete signal that describes the digital audio. It varies three parameters, amplitude, decay rate and offset, all of which are below the threshold of the human hearing; therefore the echo is not easily discovered. Offset is also used to represent the encoded bit: one offset is 1, another offset is 0. The original signal is broken into blocks before the encoding begins, and is put back together when the encoding ends. It then adds echoes to encode the secret information in each block. This method improves transmission rate and also robustness, being susceptible to the change of the data, but the introduction of the echoes might present at a statistical analysis of the sound file.

3.4 Related Study

The human auditory system (HAS) perceives sounds over a range of power greater than 109:1 and a range of frequencies greater than 103:1[13]. The sensitivity of the HAS to the Additive White Gaussian Noise (AWGN) is high as well: the noise in a sound file can be detected as low as 70 dB below ambient level. The higher the LSB layer we choose for embedding is, the higher the probability for the secret communication to be detected is. This provides a limit to the embedding layer we choose. In this paper, we will present a modification that is applied to the bits after embedding that allows for a higher embedding layer to be employed.

4 Proposed work

With the current algorithm we propose an alternative method to the LSB embedding technique for a digital audio file. The hidden information to be hidden will be called target text, and the audio file in which the information will be encoded will be called cover audio. For this algorithm we will use 3 of the 4 LSBs of each chosen sample to encode information on, namely bits number 1, 3 and 4. Using the 4th bit might introduce an error as high as 2^4-1, which would induce noise. Therefore we will be using a bit correction particular to the algorithm after the encoding of the target data, to reduce this to maximum 2. The maximum error introduced after embedding all 3 bits will be 5. Samples will be chosen in a pseudorandom way such that detection will be lowered and also no consecutive samples will be altered, since noise in the same sample region might also improve detection rate. This method improves upon embedding capacity and lowers detection rate.

4.1 Encoding technique and string length

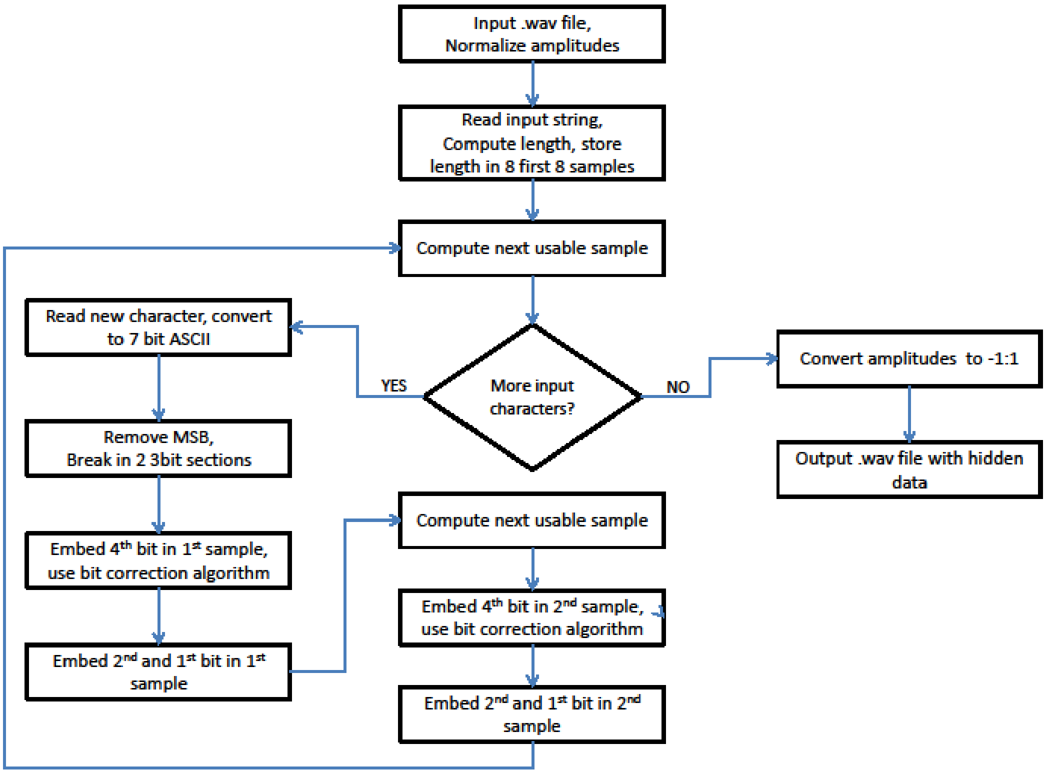

Input .wav file,
Normalize amplitudes
Read input string,
Compute length, store
length in 8 first 8 samples
Compute next usable sample
Read new character, convert
to 7 bit ASCII
YES
More input
characters?
NO
Convert amplitudes to -1:1
Output .wav file with hidden
data
Remove MSB,
Break in 2 3bit sections
Compute next usable sample
Embed 4th bit in 1st sample,
use bit correction algorithm
Embed 4th bit in 2nd sample,
use bit correction algorithm
Embed 2nd and 1st bit in 1st
sample
Embed 2nd and 1st bit in 2nd
sample

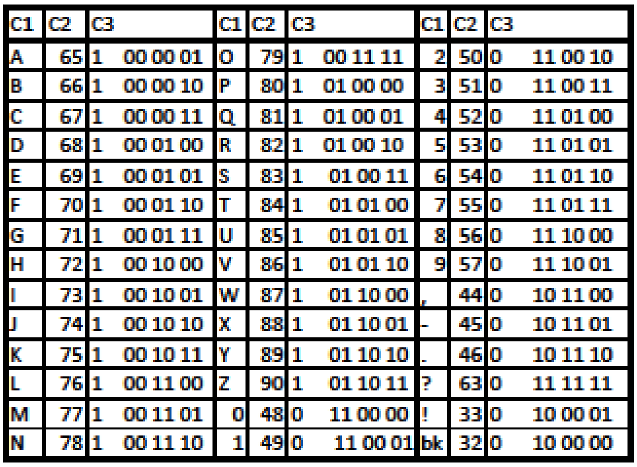
Fig. 6: Encoder workflow

For the current algorithm a WAV file will be used as cover audio. Each sample of the data chunk of a WAV file is a number ranging from -1 to 1, representing the value of the amplitude at the specific moment in time when it was quantized. For ease of access, before embedding begins, these values will be normalized to values ranging from -254 to 255, by multiplying them with multiples of 10. Numbers in the -254 to 255 range can easily be represented on 8 bits.

Also before the encoding of the actual encoding begins, the length of the target data will be calculated and converted to 8 bits. Afterwards it will be hidden on the LSB of each of the first 8 samples of the encoding.

4.2 ASCII conversion

To increase the theoretical maximum size of the target data that can be embedded, the algorithm will use only a small set of characters. These characters (capital A-Z, 1-9, and some special characters such as “,”, “.”, “-“, “?”, “!” and “blank) can make up a majority of text data that is desired to be transported. The intent behind using these specific characters is clear from the figure: the most significant bit in either of these 7-bit ASCII characters can be removed when encoding the target data, and then added back upon decoding, thus having a 6 bit data sequence to encode for each character. Hiding 3 bits in each sample, this means that one character can be encoded on 2 samples, as opposed to the original LSB method which would require up to 7 samples to be used for each character.



C1	C2	C3	C1	C2	C3	C1	C2	C3
A	65	1 00 00 01	O	79	1 00 11 11	2	50	0 11 00 10
B	66	1 00 00 10	P	80	1 01 00 00	3	51	0 11 00 11
C	67	1 00 00 11	Q	81	1 01 00 01	4	52	0 11 01 00
D	68	1 00 01 00	R	82	1 01 00 10	5	53	0 11 01 01
E	69	1 00 01 01	S	83	1 01 00 11	6	54	0 11 01 10
F	70	1 00 01 10	T	84	1 01 01 00	7	55	0 11 01 11
G	71	1 00 01 11	U	85	1 01 01 01	8	56	0 11 10 00
H	72	1 00 10 00	V	86	1 01 01 10	9	57	0 11 10 01
I	73	1 00 10 01	W	87	1 01 10 00	,	44	0 10 11 00
J	74	1 00 10 10	X	88	1 01 10 01	-	45	0 10 11 01
K	75	1 00 10 11	Y	89	1 01 10 10	.	46	0 10 11 10
L	76	1 00 11 00	Z	90	1 01 10 11	?	63	0 11 11 11
M	77	1 00 11 01	0	48	0 11 00 00	!	33	0 10 00 01
N	78	1 00 11 10	1	49	0 11 00 01	bk	32	0 10 00 00

Table 1: Character ASCII coding (with MSB removal)

As well as this, the MSB does not need to be transported anywhere. Upon decoding of the hidden data, this MSB is added back to each character by the following logic: if the 6th bit is 0, we add 1 as the MSB. If it is 1, we add 0. This behaviour can be observed in Table 1, where C1 is the original character, C2 is the decimal ASCII value of the character in, and C3 is the ASCII value on 7 bits. For the decoding of the target data, we concatenate the 4th, 2nd and 1st LSB from 2 samples, and add the MSB then transcode the 7bit ASCII to a character. The fact that the 7th bit of each character is not transported at all only adds more resistance to detection to the algorithm.

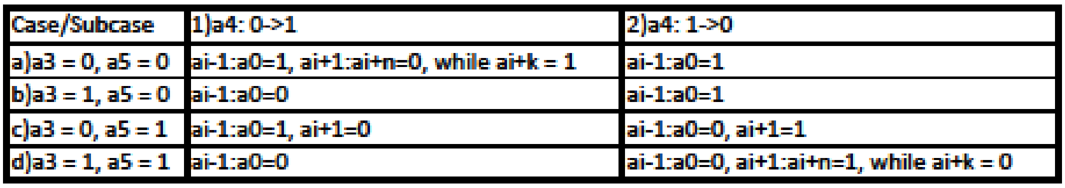
4.3 Sample selection

The algorithm uses a pseudo-random selection of samples to avoid detection. From experiments performed, if consecutive samples are chosen and modified, or even samples in a small cluster, the additive noise generated will be detected by third parties that might intercept the file, thus making the algorithm obsolete. Therefore, the samples modified need to be sufficiently far from each other to avoid this behaviour. To this end, the samples will be selected as follows: the first 8 samples of the cover audio data will always contain the length of the target data on each of the LSB. The next sample after these 8 to be modified will be first prime number that follows after the length of the target data if target data is longer than 8 characters, otherwise it is the first prime number after length + 8. The following samples are all the following prime numbers, to which we add the number of samples already modified. For example, in a phrase that is 15 characters long, the first 8 samples' LSBs will be changed to 00001111. The next samples to be modified are: 17+1=18, 19+2=21, and so on up to 73+15=88. Keeping in mind a wav file is sampled at 44.1 kHz, this pseudorandom algorithm adds an extra layer of detection resistance, while still using a good number of samples.

4.4 Bit correction algorithm

Because of using a higher LSB layer, the proposed algorithm might increase the noise it adds to the sound file. To avoid this, we have devised a bit correction algorithm, to decrease the difference between the original carrier and the output file containing secret data to a minimum. The secret data is embedded in bits number 1, 2 and 4, therefore the maximum value between the input and output amplitudes is 2^4-1. After using the bit correction algorithm, this is reduced to 3.

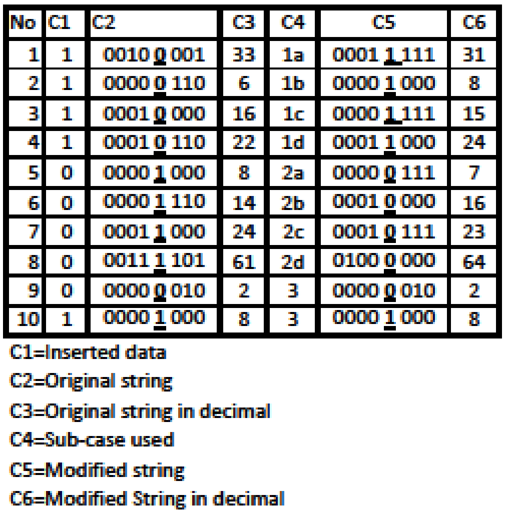
There are 3 cases that can appear when embedding secret data: the bit needs to be changed either from 0 to 1, 1 to 0, or stay the same. In the best case scenario when the bit does not need modification, the error introduced is 0. For the other cases, there are 4 subcases considered for each of them. These subcases take into consideration the 3rd and 5th bit, bits that can be modified regardless of the target data. These subcases can be observed in Table 2.



Case/Subcase	1)a4: 0->1	2)a4: 1->0
a)a3 = 0, a5 = 0	ai-1:a0=1, ai+1:ai+n=0, while ai+k = 1	ai-1:a0=1
b)a3 = 1, a5 = 0	ai-1:a0=0	ai-1:a0=1
c)a3 = 0, a5 = 1	ai-1:a0=1, ai+1=0	ai-1:a0=0, ai+1=1
d)a3 = 1, a5 = 1	ai-1:a0=0	ai-1:a0=0, ai+1:ai+n=1, while ai+k = 0

Table 2: Bit correction subcases

These cases will change the other bits of the sample so that the noise will be lowered. For example, in the case 1a, where the 4th bit is changed from 0 to 1 and 3rd and 5th bits are 0, we also change bits 3, 2 and 1 to 1 and bits above bit 5 to 0 until we encounter a bit that is already 0, when it stops. All these cases are exemplified in Table 3.



No	C1	C2	C3	C4	C5	C6
1	1	0010 0 001	33	1a	0001 1 111	31
2	1	0000 0 110	6	1b	0000 1 000	8
3	1	0001 0 000	16	1c	0000 1 111	15
4	1	0001 0 110	22	1d	0001 1 000	24
5	0	0000 1 000	8	2a	0000 0 111	7
6	0	0000 1 110	14	2b	0001 0 000	16
7	0	0001 1 000	24	2c	0001 0 111	23
8	0	0011 1 101	61	2d	0100 0 000	64
9	0	0000 0 010	2	3	0000 0 010	2
10	1	0000 1 000	8	3	0000 1 000	8

C1=Inserted data
C2=Original string
C3=Original string in decimal
C4=Sub-case used
C5=Modified string
C6=Modified String in decimal

Table 3: Bit correction examples

After applying this bit correction on the 4th bit, we can see the biggest error introduced is 3 in the worst case scenario, and in a good number of cases it is either 1 or 2. To this error we add another 3 in 2 cases, or 0 and 1 in the other cases, from modifying bits 1 and 2 of the sample.

4.5 Decoding the message

The steganographic decoder is the reverse of the encoding stage. Upon receiving the stego-audio file, the user's target is to extract the hidden data embedded in it. This process is divided into two steps. First, the length of the target data is extracted from the first 8 samples. The second step involves computing which samples were affected, removing bits 1, 2 and 4 from each sample and concatenating them 2 by 2. This is followed by the MSB each character being added to each combination of 2 samples by the rule: if the 6th bit is 1 we add 0, otherwise add 1. In the end, the 7 bit coded ASCII characters are converted to decimal, ending up with the secret message.

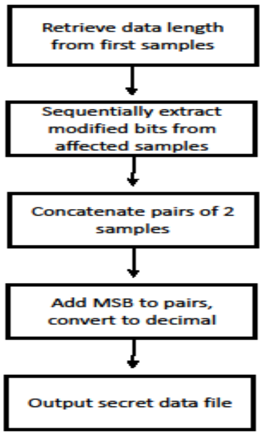

Retrieve data length from first samples
Sequentially extract modified bits from affected samples
Concatenate pairs of 2 samples
Add MSB to pairs, convert to decimal
Output secret data file

Fig. 7: Decoder workflow

5 Conclusion

This paper introduced an audio steganographic algorithm capable of embedding data on a higher layer level, while trying to handle the effect of the extra noise this would normally involve. For this algorithm we used one fixed parameter, the length of the secret message, embedded in the first 8 samples. To expand upon this, we are considering taking the audio file as a continuous stream as opposed to a table of samples, at the end of the file we start again from the beginning. This way, more samples can be used. To avoid stacking samples in one are, and therefore introducing unwanted noise, for each sample that is found to be acceptable by the pseudo-random sample picker, we will take a fixed sized block of samples around it, and seeing if samples in that block have already been picked. As well as this, continuing work will introduce a noise metric. Using this metric, we can determine if a sample can be safely modified, or skipped, every step of the way. This should not interfere with the decoding algorithm in any way, but add storage room without increasing the size of the file, and adding extra security against detection.

We are also considering switching from a fixed set of ASCII characters to any Unicode changing the way characters are represented, not using samples for bit embedding, but employing the use of a bit mask of a length to be determined. This way we will have the possibility of embedding any type of file, such as an image or an executable.

References

[1] Kelley, Jack. *Terror groups hide behind Web encryption.* USA Today. 19 June 2001.

[2] C. Harvey, *The Smartphone is the New PC*, Processor.com Featured Article, PP. 29, June 2011

[3] R. Andersen, *Security Engineering: A Guide to Building Dependable Distributed Systems,* University of Cambridge Computer Laboratory Paper, Chapter 5-Cryptography

[4] B. Pfitzmann, "Information hiding terminology", Information hiding: first international workshop, Cambridge, England, May 1996

[5] F. A. Petitcolas, R.J. Anerson, M. G. Kuhn, *Information Hiding – A survey*, Proceedings of the IEEE: special issue on protection of multimedia content, July 1999

[6] P. Reddy, S. Kumar, *Steganalysis Techniques: A Comparative Study*, University of New Orleans Theses and Dissertations, 2007

[7] N. F. Johnson, S. Jajodia, *Exploring Steganography: Seeing the Unseen*, IEEE Computer, pp. 22-34, February 1998

[8] S. Lyu and H. Farid, *Detecting Hidden Messages Using Higer-Order Statistics and Support Vector Machines*, 5th International Workshop on Information Hiding, Noordwijkerhout, The Netherlands, 2002

[9] C. E. Shannon, *Communications in the presence of noise,* Proceedings of the IRE, vol. 37, pp 10-21, January 1949

[10] https://ccrma.stanford.edu/courses/422/projects/WaveFormat

[11] P. Jayram, H. R. Ranganatha, H. S. Anupama, *Information Hiding Using Audio Steganography – A Survey*, The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.3, August 2011

[12] S. Katzenbeisser, F. A. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House Computer Security Series, 2000

[13] Basu, P. N.; Bhowmik T., *On Embedding of Text in Audio – A case of Steganography*, International Conference on Recent Trends in Information, Telecommunication and Computing, 2010

Alexandru Cristian BOZDOG
“Politehnica” University of Timisoara, Romania
Department of Computer Science
Pta Victoriei No. 2, 300006 Timisoara
Romania
E-mail: bozdogalex@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

SMLib – music browser

Georgi Chavdarov
Teacher Coordinator: Assoc. Prof., PhD Katalina Grigorova

Abstract

Collectors of digital music have almost as many different needs as they do tastes in music. Music fans need programs that do more than just play music. The presented application SMLib (Smart Music Library) helps the users bring order to large collections of music.

1 Introduction

Since the Internet has become a huge source of files that occupy hundreds of gigabytes of hard disks, users often begin to encounter a very nasty problem - thousands of files from different devices such as hard drives, DVD drives, and flash drives are scattered in different directories and the user is not always possible to find what he/ she is looking for. But when it comes to music - the situation is even more critical. Hundreds MP3s, collected from various sources are stored in a bunch of directories that may be difficult to reach due to the lack of a unique system for organizing music. It is known that Windows Explorer offers searching, sorting and grouping files, but if you have a directory in the tens of thousands of files, how much CPU time it will take to retrieve from each file required information such as artist name, song tempo, tonality, release year? Surely, too much. In these cases, the applications that classify music files by different criteria are very helpful.

There are various classification systems of music files according to purpose, the developers have set and the results obtained. One aspect, the object of extensive research, is associated with extracting information from audio files and their classification by genre [1]. As a result of the implementation of such systems audio files can automatically be marked as belonging to a particular genre or even subgenre, which is useful for subsequent searches.

Wide range of users, however, require a means to provide convenient functionalities for the selection of a particular set of audio files from a number of available ones. Applications such as iTunes offer playlists-creating. Using it or similar applications the users are able to achieve the desired goal - quickly process audio files and create so-called Smart Playlists.

The selection of specific files and organizing them into playlists is well developed in many applications of a similar nature, as this activity is one of the most commonly used by music fans. But the addition of a new song on the hard drive or flash drive has to be done manually in the collection so that the application will "know" about its existence.

The proposed application (SMLib) offers a mixed version of the above two types of applications - applications that use existing music files on any source connected to the computer and at the same time smooth scrolling of large numbers of directories and files.

With SMLib, life becomes easier. Marking drives and/ or directories that you want to load you can start arrangement - arrangement in a real time. Whether you want to listen to your favorite tunes of an artist or want to hear all the songs from 10 years ago in a certain genre – state your need in SMLib and it will quickly find what you are looking for.

For the convenience of users SMLib can be used in two ways - as a player and as a browser. In the first case you have an excellent player that allows listening to tracks. In the second case you have on your disposal Drag & Drop area where you can place the found files set by specified criteria, which are subsequently passed to external programs - for production, audio-processing, radio keeping and others.

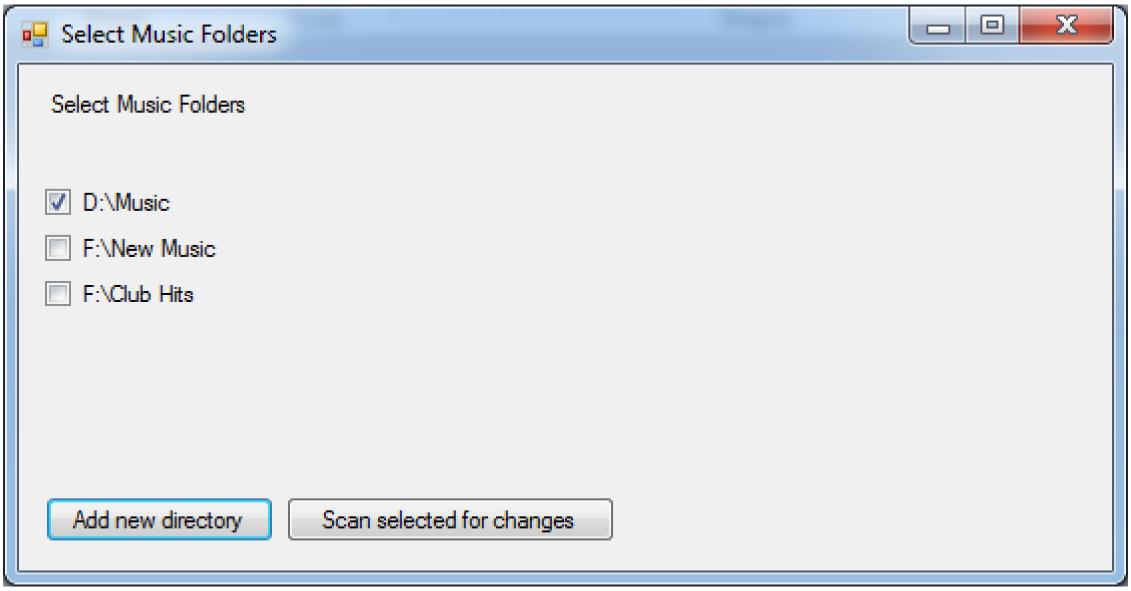
The application is written using Visual C# and is intended only for the operating system Microsoft Windows.

2 Browsing and playing music

2.1 Main functionality

Collectors of digital music have almost as many different needs as they do tastes in music. The best programs, dealing with audio files, do more than just play music; they also help bring order to large collections of music files and they are simple and fun to interact with [3].

The presented SMLib allows browsing music files from external devices with the ability for sorting and multi grouping them on various criteria.



Select Music Folders

Select Music Folders

☑ D:\Music

□ F:\New Music

□ F:\Club Hits

Fig. 1. Inserting and updating folders window

2.1.1 Loading application

When the application loads, a dialog box in which the user is able to choose which folders to scan (Fig. 1.). When first starting the application the user has the opportunity to specify a folder which supposedly contains music files. With the recent launch SMLib loads already known folders and asks the user whether these need to be re-scanned for changes. Possible changes are the following types:

- If you have added new songs the program adds them to the database, if the songs are deleted it removes them from the database;
- If you have changed tracks (ID3 tags or file names) – the program changes them in the database.

2.1.2 Reading MP3 files and the corresponding ID3 tags

For reading the tags the standard ID3 library is used. If the user wants to update a tag, this is done in the same way, no matter what kind of encoding the user uses.

ID3 is a metadata container most often used in conjunction with the MP3 audio file format. It allows information such as the title, artist, album, track number, and other information about the file to be stored in the file itself.

There are two unrelated versions of ID3: ID3v1 and ID3v2.

After the creation of the MP3 standard, there appeared a problem with storing data about the file. Standalone MP3s didn't have any special method of doing this. In 1996 Eric Kemp had the idea to add a small chunk of data to the audio file, thus solving the problem.

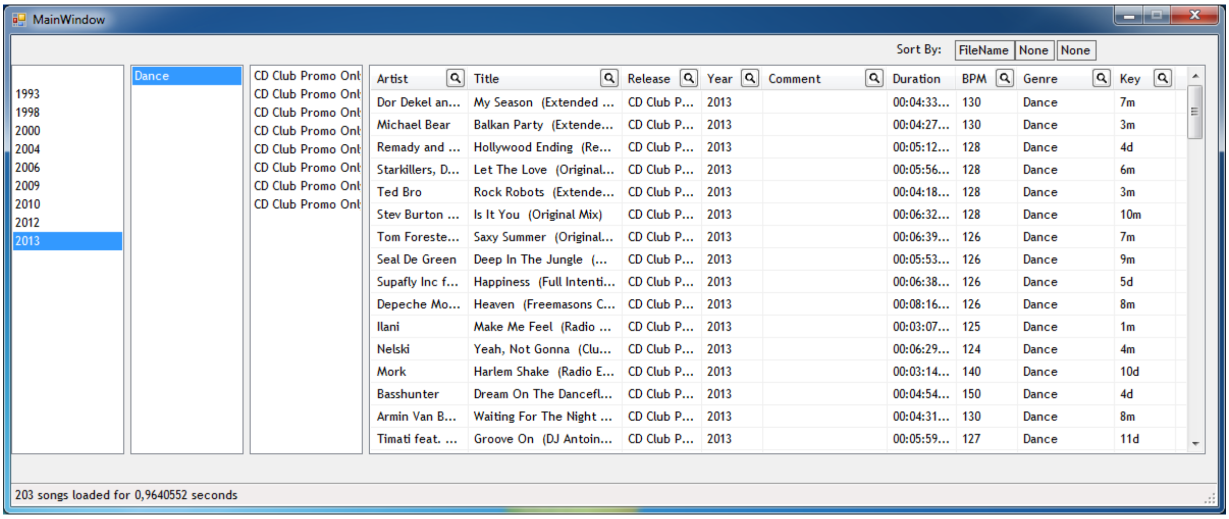
In 1998, a new specification called ID3v2 was created by multiple contributors. Although it bears the name ID3, it has little to no relation to ID3v1.

ID3v2 tags are of variable size, and usually occur at the start of the file, to aid streaming media [2].

In the introduced application ID3 version 2.4 is used. During the reading the file tags are extracted from version 2, whatever the minor version is. When updating tags always version 2.4 (the latest available) is used. This allows avoiding problems with reading the pieces tagged with old software, while for updating the latest version of ID3 is used.

2.1.3 Storing information in the database

The information about audio files is stored in a local database that is located in the SMLib program directory. The database is created within MSAccess 2003 format. It contains information about the artist, track name, album name, year, comment, time, tempo and tonality. OleDbDriver is used to connect the program with the database, as it does not require the server to be installed.



Sort By: FileName None None

1993
1998
2000
2004
2006
2009
2010
2012
2013

Dance

CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl
CD Club Promo Onl

Artist	Title	Release	Year	Comment	Duration	BPM	Genre	Key
Dor Dekel an...	My Season (Extended ...	CD Club P...	2013		00:04:33...	130	Dance	7m
Michael Bear	Balkan Party (Extende...	CD Club P...	2013		00:04:27...	130	Dance	3m
Remady and ...	Hollywood Ending (Re...	CD Club P...	2013		00:05:12...	128	Dance	4d
Starkillers, D...	Let The Love (Original...	CD Club P...	2013		00:05:56...	128	Dance	6m
Ted Bro	Rock Robots (Extende...	CD Club P...	2013		00:04:18...	128	Dance	3m
Stev Burton ...	Is It You (Original Mix)	CD Club P...	2013		00:06:32...	128	Dance	10m
Tom Foreste...	Saxy Summer (Original...	CD Club P...	2013		00:06:39...	126	Dance	7m
Seal De Green	Deep In The Jungle (...	CD Club P...	2013		00:05:53...	126	Dance	9m
Supafly Inc f...	Happiness (Full Intenti...	CD Club P...	2013		00:06:38...	126	Dance	5d
Depeche Mo...	Heaven (Freemasons C...	CD Club P...	2013		00:08:16...	126	Dance	8m
Ilani	Make Me Feel (Radio ...	CD Club P...	2013		00:03:07...	125	Dance	1m
Nelski	Yeah, Not Gonna (Clu...	CD Club P...	2013		00:06:29...	124	Dance	4m
Mork	Harlem Shake (Radio E...	CD Club P...	2013		00:03:14...	140	Dance	10d
Basshunter	Dream On The Dancefl...	CD Club P...	2013		00:04:54...	150	Dance	4d
Armin Van B...	Waiting For The Night ...	CD Club P...	2013		00:04:31...	130	Dance	8m
Timati feat. ...	Groove On (DJ Antoin...	CD Club P...	2013		00:05:59...	127	Dance	11d

203 songs loaded for 0,9640552 seconds

Fig. 2. Grouped and sorted audio files

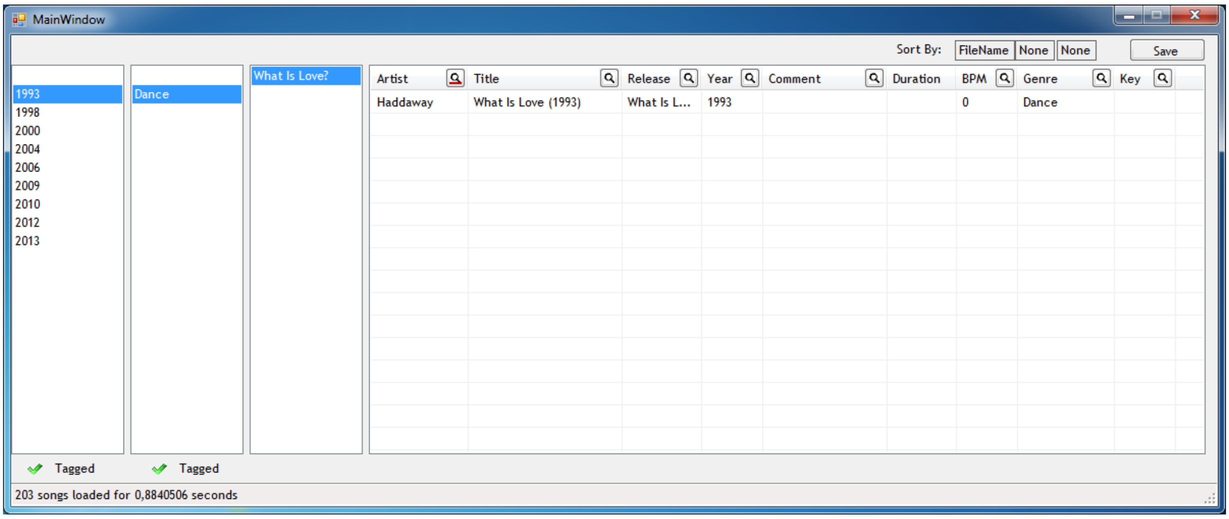
2.1.4 Sorting and grouping

The user has full rights to change the rules for sorting and grouping, thus he has complete freedom to choose the criteria. Sorting is performed by dragging the button 🔍 and dropping it into one of three categories for grouping. After launching, the songs are immediately grouped according to the new criteria. Sorting also becomes easier - for this purpose the same button is used, but it has to be dropped into the sorting area. Fig. 2. shows an example of files selected and grouped on determined criteria.

Sorting and grouping can be performed by three criteria at most. First, the criteria for grouping has to be chosen (on the left side), then the sorting criteria can be set (on the right side of the work area).

2.1.5 Updating tags

Sometimes there are audio files which do not contain ID3 tags. In these cases the user is able to add or update the data about the file. This function is performed by dropping the song on the actual value and after 2 seconds the new value is selected. The user can move the cursor to another location to continue tagging. When placing the mouse over the button Save, the updated information is stored in the ID3 fields. An example of process of tagging is presented on Fig. 3.



Artist	Title	Release	Year	Comment	Duration	BPM	Genre	Key
Haddaway	What Is Love (1993)	What Is L...	1993			0	Dance	

203 songs loaded for 0,8840506 seconds

Fig. 3. Example of tagging files

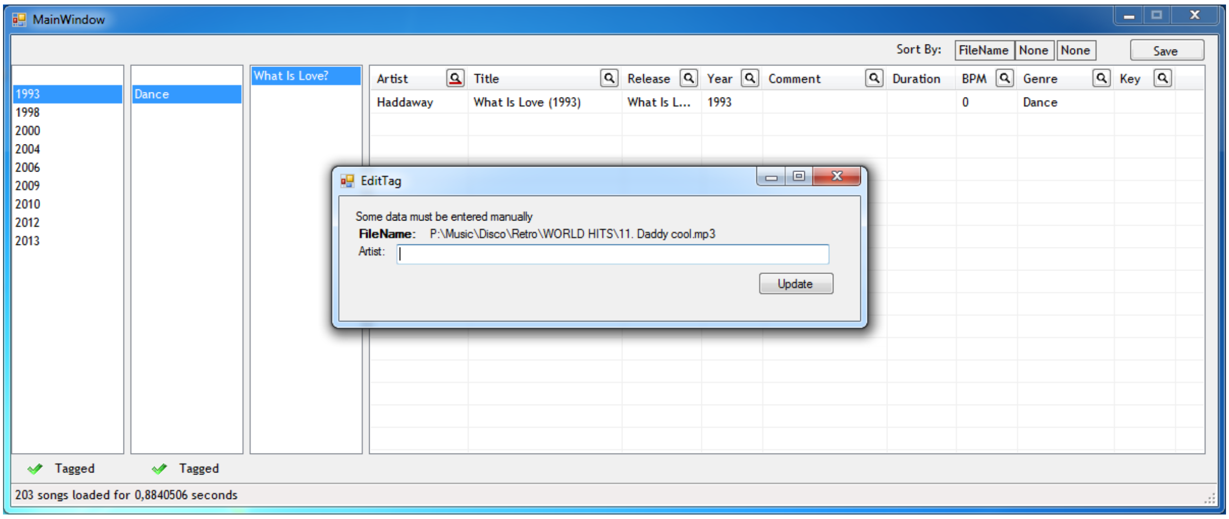
For example, there is a song X, which is not tagged correctly and tag information about artist, year, and genre is missing. You have to drag the song on the list of years and put it on the correct year. Then drag it in the same way to the proper genre. When the right value does not exist in the database, it has to be added. To do this, the cursor has to be placed on 🔍 and has to be kept 2 seconds in order to inform the program that the value has to be entered manually. When all the tagging is done the cursor is moved on the Save button, thus the data is automatically stored. The program knows that there are tags, that need to be defined and the appropriate window appears (Fig. 4.). It is allowed to add manually as many tags as you need.

2.2 SMLib implementation

With SMLib users can browse, search and compile lists of audio files that match certain criteria.

When starting the application and specify the folder to update, SMLib synchronize information from database and the content of those folders.

The executed tests show, that the application handles approximately 170 - 190 songs per second, which means that the collection of 2,000 pieces is scanned in about less than a minute, if the files are in the same directory. If the pieces are in different directories which have to be searched one by one, the recursion takes time, and the time needed depends entirely on the speed of hard drive or flash drive.



Artist	Title	Release	Year	Comment	Duration	BPM	Genre	Key
Haddaway	What Is Love (1993)	What Is L...	1993			0	Dance	

Some data must be entered manually

FileName: P:\Music\Disco\Retro\WORLD HITS\11. Daddy cool.mp3

Artist:

203 songs loaded for 0,8840506 seconds

Fig. 4. Editing a tag manually

3 Conclusion

The designed application holds convenient and easy to use features. Music lovers have at their disposal browser and player that helps them to arrange their favorite music files and to compose collections as they wish.

The author plans expanding the capabilities in the following areas:

- Supporting for more formats such as AAC/ FLAC;
- Conversion from one format to another with preservation of contents of the ID3 tags;
- Integration with Windows 8 Metro UI;
- Virtual File Browser tagging of entire folders and subfolders;
- Creating music CDs.

References

[1] C. McKay and I. Fujinaga, *Combining Features Extracted from Audio, Symbolic and Cultural Sources*. In Proceedings of the 9th International Conference on Music Information Retrieval, Izmir, 2008, 597-608

[2] Wikipedia. *ID3*. 2013, http://en.wikipedia.org/wiki/ID3.

[3] Jukebox Software Review. *Why Do I Want Jukebox Software?* 2013, http://jukebox-software-review.toptenreviews.com/

Georgi CHAVDAROV
University of Ruse
Department of Informatics and
Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
powerbsd@abv.bg

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

R&D smart TV

Octavian George Cimpu
Teacher / Coordinator: Anca Vasilescu

Abstract

Purpose – This article aims to introduce the reader in the world of smart TVs. It presents general information on smart TVs, a market research and the possibilities of development in that direction. The goal is to present a new concept about Smart TVs: the possibility of combining the applications on smart phones with the ones on smart TVs- different applications that can communicate together via UPnP.

Research limitations/implications – The reliability of the research presented below is however limited, as the smart TV is an evolving technology in the phase of market introduction. Documentation is rare and vague, and so is the community.

The definition and characteristics of smart TVs are still uncertain. Still, if you think out of the box, you will see beyond the concept of a smart TV and get an image on its potential. The phone is the best example to give, since the current smart phone does more than anyone could have predicted 10years ago. So who knows where the smart TVs will take us! Combine those two and the possibilities are unimaginable.

1 Introduction to smart TVs

Smart phones, smart pads, and smart TVs appeared last decade and have since changed the concept of media. The smart TV, in particular, is expected to radically change the television broadcasting industry. Until recently, the traditional TV was a device that provided most of the entertainment and information in a normal household. Over the years, once with the increasing use of the internet and the need to access more and more often, the conventional TVs have turned into smart TVs and their role is now ever expanding.

1.1 Definition

According to Businessinsider.com, a smart TV device is either a television set with integrated Internet capabilities or a set-top box for television that offers more advanced computing ability and connectivity than a contemporary basic television set. Smart TVs may be thought of as an information appliance or the computer system from a handheld computer integrated within a television set unit, as such smart TV often allows the user to install and run more advanced applications or plugins / add-ons based on a specific platform. Smart TVs run complete operating system or mobile operating system software providing a platform for application developers.[1234]

Smart TV platforms or middleware have a public Software development kit (SDK) and/or Native development kit (NDK) for apps so that third-party developers can develop applications for it, and an app store so that the end-users can install and uninstall apps themselves. The public SDK enables third-party companies and other interactive application developers to “write” applications once and see them run successfully on any device that supports the smart TV platform or middleware architecture which it was written for, no matter of who the hardware manufacturer is.

Basically smart TVs can be used for the following:

1. "Deliver content from other computers or network attached storage device on a network like photos, movies and music using either a Digital Living Network Alliance / Universal Plug and Play media server or similar service program like Windows Media Player or Network-attached storage (NAS), or via iTunes.
2. Provides access to Internet-based services including traditional broadcast TV channels, catch-up services, video-on-demand, Electronic program guide, interactive advertising, personalisation, voting, games, social networking, and other multimedia applications"[5].

1.2 Functions

Smart TVs can also provide access to user-generated content (either stored on an external hard drive or in cloud storage) and to interactive services and Internet applications, such as YouTube.

Smart TV devices deliver the traditional content more efficiently, by combining information from the Internet with content from TV Providers. Users can track and receive reminders about favourite shows or sporting events, as well as the ability to change channels from immediate viewing.

Some devices even feature additional interactive organic user interface / natural user interface technologies for navigation controls and other human interaction with your Smart TV, with such as second screen companion devices, spatial gestures input like with Xbox Kinect and even for speech recognition for native tongue user interface.

1.3 Technology used

The concept of Smart TVs is continuously developing and is constantly promoted. Some smart TVs have the ability to run applications (sometimes available via an „app store‟ digital distribution platform), interactive on-demand media, personalized communications, and social networking features.[6]

2 Research

[1] Steve Kovach (December 8, 2010). *"What Is A Smart TV?"*. Businessinsider.com.

[2] *"Using the UpnP Control Point API"* Microsoft Retrieved 2011-03-02

[3] *"Android Holds the Key to Samsung’s Smart TV Plans"*. *Wired*. January 17, 2012.

[4] *"What Smart TVs Need to Succeed"*. Cepro.com. January 17, 2012.

[5] http://en.wikipedia.org/wiki/Smart_TV

[6] *"TVs get smart with Internet access in 2011"*. Abc15.com. August 23, 2010.

2.1 Market research

Smart TVs began to be adopted worldwide in 2012, as the prices of renowned connected TV sets continued to fall. According to a report made by IHS iSuppli, smart TV shipments climbed 27% in 2012 to reach 66million units. By 2015, the smart TVs are expected to make up 55% of the market as global shipments reach 141 million units. “Despite a decline in global television shipments in 2012, customers demand for internet-connected televisions soared during the year – and the surge in sales shows no signs of abating,” IHS analyst Veronica Thayer said. “Smart TV are rapidly joining the mainstream as manufacturers refine their products to add new features and to make them easier to use.”[7]

We can look at the smart TV high market penetration as an opportunity, because of its on growing store, or at least with the curiosity of a software developer.

2.2 Smart TV Manufacturers- Samsung

Samsung, perhaps more than any other manufacturer, has been promoting the Smart devices for a number of years and has been first to market a number of innovations: gesture, voice and face recognition (similar to Xbox 360 Kinect), sleek design, quality at its finest. They had a close competition with another manufacturer of mobile devices and computers over patent violation but, in the Smart TV world – where they lead the market in sales - it seems its them that they lead whilst others follow.

With their very own App Store, there are numerous streaming possibilities and new control interaction. Now let‟s lift the curtain and go into detail to see what makes everything work:

2.2.1 Smart TV SDK

Samsung has its own SDK (Software Development Kit). SDK is a set of software development tools which is used to develop Apps.
Samsung platform only supports web technologies based apps (JS, CSS, HTML, FLASH).

2.2 Universal Plug and Play

“**UPnP** is a set of networking protocols that permits networked devices, such as personal computers, printers, Internet gateways, Wi-Fi access points and mobile devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment. UPnP is intended primarily for residential networks without enterprise class devices.

The concept of UPnP is an extension of plug-and-play, a technology for dynamically attaching devices directly to a computer, although UPnP is not directly related to the earlier plug-and-play technology. UPnP devices are "plug-and-play" in that when connected to a network they automatically establish working configurations with other devices.”[8]

[7] http://bgr.com/2013/02/22/smart-tv-sales-2012-340405/

[8] *"Using the UpnP Control Point API"* Microsoft Retrieved 2011-03-02

3 Program Code

3.1 General info

Through the UPnP that we presented above, we will control our smart TV app with another app developed for an android smart phones. Here is how it works:

3.1 Smart TV application

A smart TV application consists of three big components:

1. Htmls
2. Scenes (Java script)
3. Stylesheets (CSS)

1. “HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.”[9]

2. “JavaScript (JS) is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that was displayed.” [10]

3. “CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts.”[11]

3.1.1 HTML homepage.html

```
<!-- this html will be a contents of div Scenehomepage -->
 <img id="svecImage_bg1" />
 <img id="svecImage_bg_poster" />
 <img id="svecImage_bg_title" />
 <img id="svecImage_poster" />
 <div id="svecKeyHelp1"></div>
...
```

3.1.2 Stylesheet homepage.css

```
#Scenehomepage{
	position : absolute;
	left : 0px;
	top : 0px;
	width : 1280px;
	height : 720px;
	background-color : #ffffff;
	background-image : url('');
}
#Scenehomepage #svecImage_bg_title{
	position : absolute;
	left : 0px;
	top : 67px;
	width : 1280px;
	height : 82px;
```

[9] http://en.wikipedia.org/wiki/HTML

[10] Flanagan 2006, p. 1.

[11] *"What is CSS?"*. World Wide Web Consortium. Retrieved December 2010.

```
}

#Scenehomepage #svecKeyHelp1{
        position : absolute;
        left : 0px;
        top : 675px;
}
...
```

3.1.3 Scenes homepage.js

```
function Scenehomepage() {

};

Scenehomepage.prototype.initialize = function() {
        alert("Scenehomepage.initialize()");
$('#svecImage_bg_title').sfImage({
                src : 'images/bg_title.png'
        });
        $('#svecImage_poster').sfImage({
                src : 'images/poster0.jpg'
        });
        ...
};
Scenehomepage.prototype.handleShow = function(data) {
        alert("Scenehomepage.handleShow()");
        // this function will be called when the scene manager show this scene
};

Scenehomepage.prototype.handleHide = function() {
        alert("Scenehomepage.handleHide()");
        // this function will be called when the scene manager hide this scene
};

Scenehomepage.prototype.handleFocus = function() {
        alert("Scenehomepage.handleFocus()");
        // this function will be called when the scene manager focus this scene
};

Scenehomepage.prototype.handleBlur = function() {
        alert("Scenehomepage.handleBlur()");
        // this function will be called when the scene manager move focus to another
        // scene from this scene
};

Scenehomepage.prototype.handleKeyDown = function(keyCode) {
        alert("Scenehomepage.handleKeyDown(" + keyCode + ")");
        // TODO : write an key event handler when this scene get focued
        switch (keyCode) {
        case sf.key.RED:
                alert("red - romana");
                language = "ro";
                break;
        case sf.key.GREEN:
                alert("green - french");
                language = "fr";
                break;
        case sf.key.YELLOW:
                alert("yellow - deutsch");
                language = "de";
                break;
        ...
```

};

3.2 Android application

More and more devices are connected to the internet so implicit to a router. Below is a little example of how easy it is to get all the devices connected to the same network via UPnP.

An android application is divided in two big components:

4. Application Components
5. Application Resources

3.2.1 Application Components

```
public class BrowseActivity extends ListActivity {

 private ArrayAdapter<DeviceDisplay> listAdapter;
 private BrowseRegistryListener registryListener = new BrowseRegistryListener();
 private AndroidUpnpService upnpService;
 private UpnpBrowserApp app;

 private ServiceConnection serviceConnection = new ServiceConnection() {

     public void onServiceConnected(ComponentName className, IBinder service) {
         upnpService = (AndroidUpnpService) service;
         /**
          * Refresh the list with all known devices
          */
         listAdapter.clear();
         for (Device<?, ?, ?> device : upnpService.getRegistry()
                     .getDevices()) {
             registryListener.deviceAdded(device);
         }
         /**
          * Getting ready for future device advertisements
          */
         upnpService.getRegistry().addListener(registryListener);
         /**
          * Search asynchronously for all devices
          */
         upnpService.getControlPoint().search();
     }

     public void onServiceDisconnected(ComponentName className) {
         upnpService = null;
     }
 };

 @Override
 public void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);

     listAdapter = new ArrayAdapter<DeviceDisplay>(this,
             android.R.layout.simple_list_item_1);
     setListAdapter(listAdapter);

     getApplicationContext().bindService(
             new Intent(this, BrowserUpnpService.class), serviceConnection,
             Context.BIND_AUTO_CREATE);
 }
```

```
        @Override
        protected void onDestroy() {
                super.onDestroy();
                if (upnpService != null) {
                        upnpService.getRegistry().removeListener(registryListener);
                }
                getApplicationContext().unbindService(serviceConnection);
        }

        @Override
        protected void onListItemClick(ListView l, View v, int position, long id) {

                Intent intent = new Intent(this, BrowseDeviceActivity.class);
                intent.putExtra("device", position);
                startActivity(intent);
}
...
}
```

3.2.2 Application Resources

```
<?xml version="1.0" encoding="utf-8"?>
    <TextView xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@android:id/text1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceListItemSmall"
        android:gravity="center_vertical"
        android:paddingStart="?android:attr/listPreferredItemPaddingStart"
        android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
        android:minHeight="?android:attr/listPreferredItemHeightSmall"
    />
```

References

[1] Steve Kovach (December 8, 2010). *"What Is A Smart TV?"*. Businessinsider.com.

[2] *"Using the UpnP Control Point API"* Microsoft Retrieved 2011-03-02

[3] *"Android Holds the Key to Samsung's Smart TV Plans". Wired*. January 17, 2012.

[4] *"What Smart TVs Need to Succeed"*. Cepro.com. January 17, 2012.

[5] http://en.wikipedia.org/wiki/Smart_TV

[6] *"TVs get smart with Internet access in 2011"*. Abc15.com. August 23, 2010.

[7] http://bgr.com/2013/02/22/smart-tv-sales-2012-340405/

[8] *"Using the UpnP Control Point API"* Microsoft Retrieved 2011-03-02

[9] http://en.wikipedia.org/wiki/HTML

[10] Flanagan 2006, p. 1.

[11] *"What is CSS?"*. World Wide Web Consortium. Retrieved December 2010.

CIMPU Octavian-George
Universitatea Transilvania
Facultatea de Matematica si Informatica, Informatica Aplicata
Str. Iuliu Maniu nr. 50, 500091 Brasov
Romania
E-mail: octa_ge@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Advanced Encryption Standard Implementations for Extended Key Lengths

Laura Codreanu
Teacher Coordinator: Flavius Opritoiu, Mircea Vladutiu

Abstract

This paper presents a hardware model implementation for the AES encryption algorithm using 192 and 256 bits keys. Both designs were developed using the Verilog hardware description language, and were modelled and simulated in ModelSim. Taking advantage of the Verilog we used a testbench unit for the hardware model verification. The unique aspect of the design suggested in our paper is the fact that all the round keys used for the encryption and decryption iterations are generated by the key unit in parallel with the encryption or decryption process. The key unit works independently of the control unit and of the AES datapath unit. On a global scale, the whole module AES uses multiple functions, while each function was created based on a certain algorithm. For a correct verification and validation of the results obtained in our simulation we compared them with the test vectors offered by AES which stands for a publicly standardization developed by the United States federal government for use in computer systems.

Keywords: Advanced Encryption Standard, encryption, decryption, cipher- key, round key

1 Introduction

In the past few years computers have made our life easier in many ways that we can actually say that it became part of our existence. Without paying too much attention to this detail, using a computer we can accomplish all sorts of activities that normally would take us a time. Here we can mention a few of them by starting with the most simple and used one: online orders. Let us say that such a simple and routine action won't raise any suspicion regarding the security of money transfer from our account to destination. But when it is performed in order to transfer large amounts of money for paying the bills, for example, we should start wondering is this transfer process safe enough? How about if we want to transfer some highly important data that we want no one else to know except the person that should receive the information? Here is where AES intervenes: without seeing it, even if we transfer money or data, all the information that we want to send to someone is being encrypted before it leaves our computer and travels to the destination where what we transferred is being decrypted. In this way the person from the destination sees just the message as we send it and not the encrypted one. This whole process takes place without knowing it, we are just taking it as a transparent operation. But what happens when an encryption protocol such as AES-128 is not enough and our private information is being endangered?

Furthermore, how can be prevented a hacking operation attacking a banking system, a military database and why not even our medical records? This question finds its answer in our solution: using a larger cipher-key. This is the reason why the subject of our paper is the simulation of encryption and decryption using bigger keys, conferring this way a higher security with the costs of increasing the area, the power consumption and operation latency [4].

2 Background

In January 1997, The National Institute for Standards and Technologies (NIST) [1] from SUA announced the initiation development process of a new cryptographic standard (AES - Advanced Encryption Standard) which will replace the now obsolete DES. This new algorithm had to be available for all people keen on cryptography and also for those who just wanted to use it. AES had to encrypt block lengths of 128 bits and to support key lengths of 128, 192 or 256 bits. In August 1998, NIST announce that only a group of 15 algorithms had remain in the competition, later their number dropped to 5 (MARS, RC6, Rijndael, Serpent and Twofish) in april 1999. Finally (in october 2000) the winner was the Rijndael algorithm proposed by Joan Daemen and Vincent Rijmen (Belgium) [2]. So this algorithm become the new AES standard. The AES algorithm is characterized by flexibility and simplicity, which can be seen in the use of basic mathematical operations and also the symmetry of the levels [2]. Also the high cryptographic performance can be seen in its easy implementation on various platforms, both in software and hardware [2].

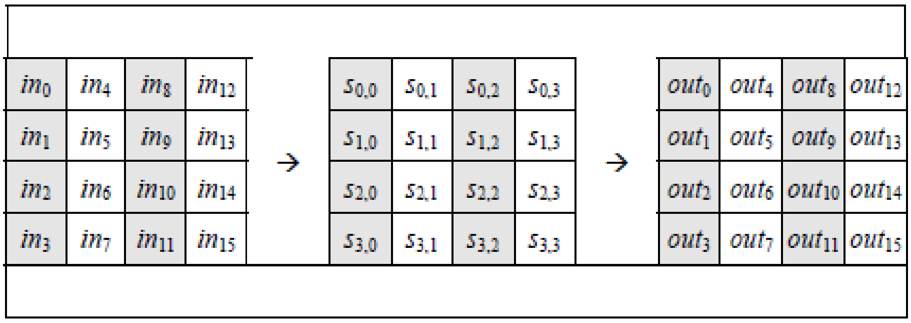
2.1 Notations

AES algorithm encrypts data blocks of 128 bits. We consider the blocks organized in words of 32 bits length. The number of columns in the state is denoted by Nb and is equal to the block length divided by 32. So the number of words are Nb = 4. The key (K) length can take the values 128, 192 or 256 bits, this length being expressed by Nk = 4, 6 or 8, meaning the number of the words (32 bits) of the key. Table 1 presents the possible variants of AES implementation and also the number of rounds needed for the encryption process.

AES type	Nk	Nb	Number of rounds
AES - 128	4	4	10
AES - 192	6	4	12
AES - 256	8	4	14

Table 1, [1]

Both the input block (in) and the output block (out) of 128 bits, are represented as tables where the bytes are mapped on columns and rows while the plaintext represents the initial state table.
The input table, state table and output table are illustrated in Figure 1:



in_0	in_4	in_8	in_{12}		$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$		out_0	out_4	out_8	out_{12}
in_1	in_5	in_9	in_{13}	→	$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	→	out_1	out_5	out_9	out_{13}
in_2	in_6	in_{10}	in_{14}		$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$		out_2	out_6	out_{10}	out_{14}
in_3	in_7	in_{11}	in_{15}		$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$		out_3	out_7	out_{11}	out_{15}

Figure 1, [1]

For encryption as well as for decryption, AES algorithm uses a given number of rounds, which contains 4 types of transformations called steps:

- SubBytes
- ShiftRows
- MixColumn
- AddRoundKey

2.2 AES Encryption

The encryption unfolds as shown in the pseudocode illustrated in Algorithm 1:

```
Cipher (byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]
    state = in
    AddRoundKey(state,w)
    for round = 1 step 1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state,w+round*Nb)
    end for
    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state,w+Nr*Nb)
    out = state
end
```

Algorithm 1, [1]

The algorithm starts by copying the input (in) into the state table. After a first addition of the initial roundkey, the specific round transformations follows, with the observation that the final round (the one situated just after closing the *for* loop) is slightly different. This final round do not contains the MixColumns(·) transformation. At the end, the state table is copied into the output state. Another apect worth mentioning is that the array w[ ] contains all the roundkeys generated by the KeyGenerator. The transformations which intervene (SubBytes(·), ShiftRows(·), MixColumns(·) and AddRound Key(·)s) and which operates on the state table are described in the next paragraphs.

2.3 AES Transformations

2.3.1 SubBytes(·) Transformation

This transformation is mapping the present state table to the next state table by using a specific substitution box denoted S. As we can see in Figure 2, the SubBytes step is a permutation which affects the bytes of the state by applying an S-box [5,6].

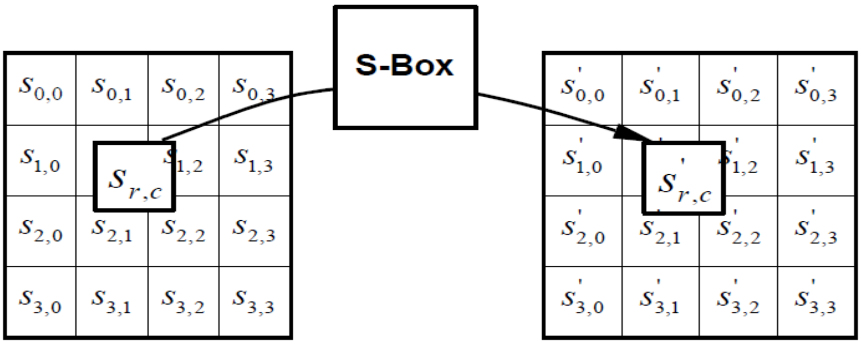
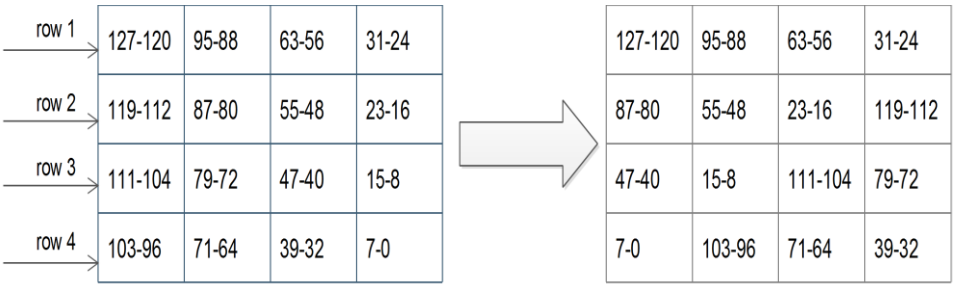


Figure 2, [1]

2.3.2 ShiftRows(·) Transformation

ShifteRows (·) step operates only on rows. In this transformation, the bytes from the state table are shifted to the left depending on the selected row: row 1 remains the same (this row is not shifted), row 2 is shifted to the left with one position, row 3 is shifted to the left with 2 positions and finally, row 4 is shifted with 3 positions. It is worth mentioning that every positions is shifted 1 byte to the left. In Figure 3 we can see an example of ShiftRows (·) for the case AES - 128.



row 1	127-120	95-88	63-56	31-24
row 2	119-112	87-80	55-48	23-16
row 3	111-104	79-72	47-40	15-8
row 4	103-96	71-64	39-32	7-0

127-120	95-88	63-56	31-24
87-80	55-48	23-16	119-112
47-40	15-8	111-104	79-72
7-0	103-96	71-64	39-32

Figure 3

2.3.3 MixColumns(·) Transformation

This transformation operates on state table in a *column-by-column* mode. MixColumns affects only the columns of the state table as shown in Figure 4.

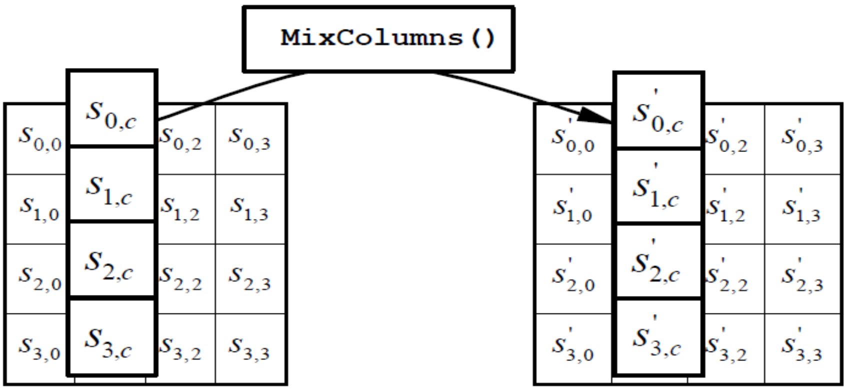


Figure 4, [1]

2.3.4 AddRoundKey(·) Transformation

With this step we modify the state table by applying a bitwise EXOR operation with the current roundkey, this way resulting the next state table:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s2,c, s3,c] + [w_{round*Nb+c}] \text{ for } 0 \leq \text{round} < \text{Nb} \qquad (1)$$

where $[w_j]$ represents a roundkey from the roundkey array *w[ ]* and *round* is the number of the current round, with $0 \leq$ round < Nr.

2.4 Key Generator

The AES algorithm uses the key K to generate the roundkey through a specific procedure which generates a total of *Nb(Nr+1)* words of 32 bits. The result of the procedure is an array of words $[w_i]$ of 32 bits with $0 \leq i <$ Nb(Nr+1). Algorithm 2 illustrates the pseudocode for key generator.

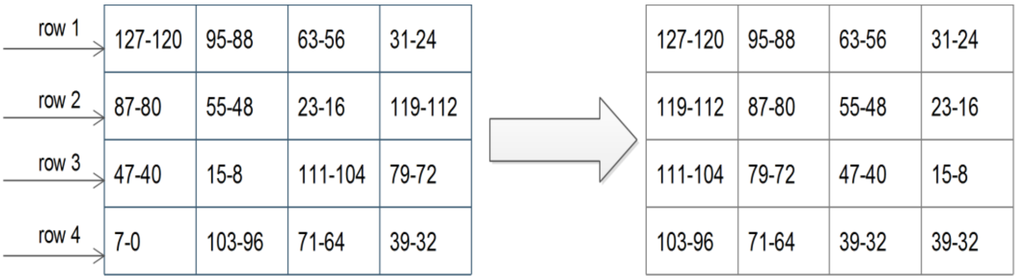
```
KeyExpansion(byte key[4 * Nk], word w[Nb * (Nr + 1)], Nk)
begin
    i=0
    while (i < Nk)
        w[i] = word[key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]]
        i = i + 1
    end while
    i = Nk
    while (i < Nb * (Nr + 1))
        word temp = w[i - 1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i / Nk]
        else if (Nk = 8 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        w[i] = w[i - Nk] xor temp
        i = i + 1
    end while
end
```

Algorithm 2, [1]

SubWord(·) is a function which maps a word of 32 bits (4 bytes) into a word, which will have the same size, by applying the function SubBytes(·) to each byte individually. RotWord(·) is also a function which permute cyclic on a word $[a_0, a_1, a_2, a_3]$ at its input and returns a word $[a_1, a_2, a_3, a_0]$ as output. Rcon[i] is a vector which contains values $[\{02\}^{i-1}, \{00\}, \{00\}, \{00\}]$, starting with *i* (where *i* indicates the current round) from value 1 and not 0. By taking a close look at Algorithm 2, we can observe that the first Nk words of the extended key are basically identical with the original key. All the next words w[i] are obtained by applying a bitwise XOR operation between the previous word w[i-1] and the word w[i-Nk]. For the words which are in positions of multiple of Nk, before applying the XOR operation on w[i-1], we apply an additional transformation: SubWord(RotWord(·)) XOR Rcon[i].

2.5 AES Decryption

The AES Decryption algorithm can be seen as an inverse operation of the encryption algorithm. It consists of the inverses of the same 4 important transformations with the mention that these 4 steps are made in an inverse way: SubBytes(·) will now become InvSubBytes(·), ShiftRows(·) will be InvShiftRows(·) and finally MixColumn(·) will be called now InvMixColumn( ). The only transformation which is not affected is the AddRoundKey(because AddRoundKey is its own inverse) [4]. Figure 5 describes best the InvShitRows(·) function.



row 1	127-120	95-88	63-56	31-24
row 2	87-80	55-48	23-16	119-112
row 3	47-40	15-8	111-104	79-72
row 4	7-0	103-96	71-64	39-32

127-120	95-88	63-56	31-24
119-112	87-80	55-48	23-16
111-104	79-72	47-40	15-8
103-96	71-64	39-32	39-32

Figure 5

The InvSubBytes(·) transformation is a substitution box reverting the effect of the direct operation SubBytes(·). The InvShiftRows(·) will shift to the right the second row with 1 position, the third row with 2 positions and the fourth row with 3 positions. The row 1 is not shifted. Every position is shifted with 1 byte. Similarly, InvMixColumns(·) operates linearly on the state matrix's columns [4].

3 Implementation of AES- 192, AES- 256

3.1 AES -192

The key generator algorithm: we take as an input the key_data on 192 bits, we keep the round_constant of 8 bits (the same as for AES-128); after the Key_Unit process the information given at its inputs, it will return as output the next round_key on 192 bits and also the next_round_constant on 8 bits as shown in Figure 6. In order for the Key_Unit to be able to return the corect values mentioned above, some workarounds need to be done: first because we use a common datapath for all implementations and because each round uses only 128 bits of round key, we had to split the next round key generation in 2 parts: one constructing the most significant 128 bits and one generating the least significant 64 bits. As a consequence, in order to generate the next round key we will use only the most significant 128 bits and the other 64 bits will be shifted with 64 positions so that these lsb bits will now become the msb bits, as Figure 7 depicts. Furthermore at every three rounds we will generate a complete new key data as the standard defines it. To implement the above mentioned strategy, we need to separate the three cases. This is why we used a flag on 2 bit, which will trigger the use of the first 128 bits of the key data when the flag is 0 and when will use the remaining 64 bits concatenated to the next 64 bits generated when the flag it has the value 1 and when the flag is set on value 2, the final, third case triggered. In the first two cases all the 4 operation on which AES is based on are performed and in the case with the flag set on value 2, there are no operation performed except the simple XOR.

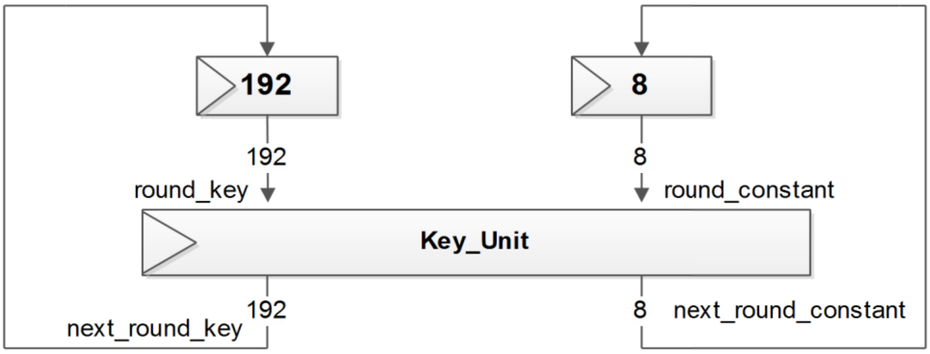
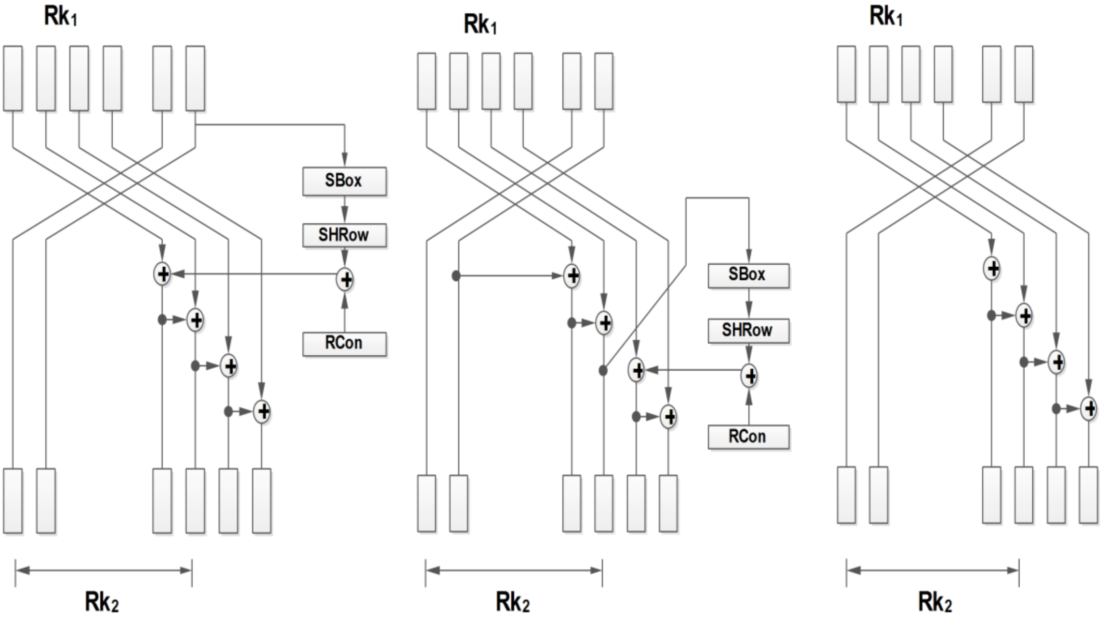


Figure 6

These workarounds can be best described by the Figure 7a) (in the left when the flag = 0), 7 b) (in the middle when flag = 1) and 7 c) (when flag = 2).



a)

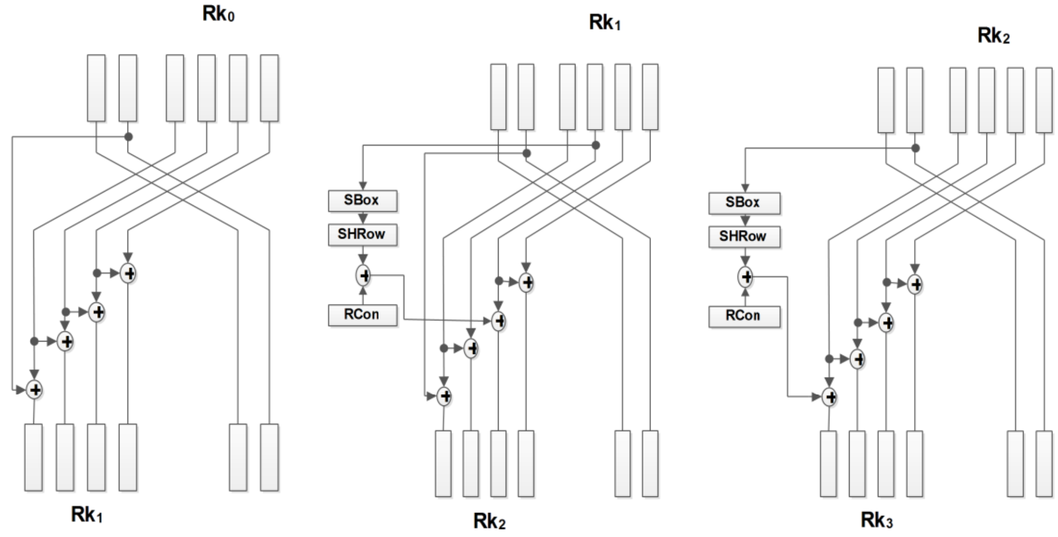
b)

c)

Figure 7

The key generator algorithm for the decryption: we take as an input the key_data on 192 bits obtained after encryption, we keep the round_constant of 8 bits (the same as for AES-128); after the Key_Unit process the information given at its inputs, it will return as output the next round_key on 192 bits and also the next_round_constant on 8 bits as shown in Figure 6. In order for the Key_Unit to be able to return the corect values mentioned above, some workarounds need to be done: first because we use a common datapath for all implementations and because each round uses only 128 bits of round key, we had to split the next round key generation in 2 parts: one constructing the most significant 128 bits and one generating the least significant 64 bits. As a consequence, in order to generate the next round key we will use only the least significant 128 bits and the other 64 bits will be shifted with 64 positions so that these msb bits will now become the lsb bits, as Figure 8 depicts. Furthermore at every three rounds we will generate a complete new key data as the standard defines it. To implement the above mentioned strategy, we need to separate the three cases. This is why we used a flag on 2 bit, which will trigger the use of the last

128 bits of the key data when the flag is 0 and when will use the remaining 64 bits concatenated to the next 64 bits generated when the flag it has the value 1 and when the flag is set on value 2, the final, third case triggered. In the first two cases all the 4 operation on which AES is based on are performed and in the case with the flag set on value 2, there are no operation performed except the simple XOR.
These workarounds can be best described by the Figure 8a) (in the left when the flag = 2), 8 b) (in the middle when flag = 1) and 8 c) (when flag = 0).



a) b) c)
Figure 8

3.1 AES -256

The key generator algorithm consists of: taking as input the key_data on 256 bits, while keeping the round_constant on 8 bits (the same as in AES-128); after the Key_Unit process the information given as input, it has to return as output the next round_key on 256 bits and also to generate the next_round_constant on 8 bits as shown in Figure 9. In order for the Key_Unit to be able to return the correct values mentioned above, some workarounds need to be done: first because we use a common datapath for all implementations and because each round uses only 128 bits of round key, we had to split the next round key generation in 2 parts: one constructing the most significant 128 bits and one generating the least significant 128 bits. As a consequence, in order to generate the next round key we will use only the most significant 128 bits and the other half will be shifted with 128 positions so that these lsb bits will now become the msb bits, as Figure 10 depicts. Furthermore at every two rounds we will generate a complete new key data as the standard defines it. To implement the above mentioned strategy, we need to separate the two cases. This is why we used a flag on 1 bit, which will trigger the use of the first half of the key data when the flag is 0 and when will use the other half when it has the value 1. In the first case all the 4 operation on which AES is based on are performed and in the case with the flag on value 1, only the S-box is performed, as described by the AES standard [1].

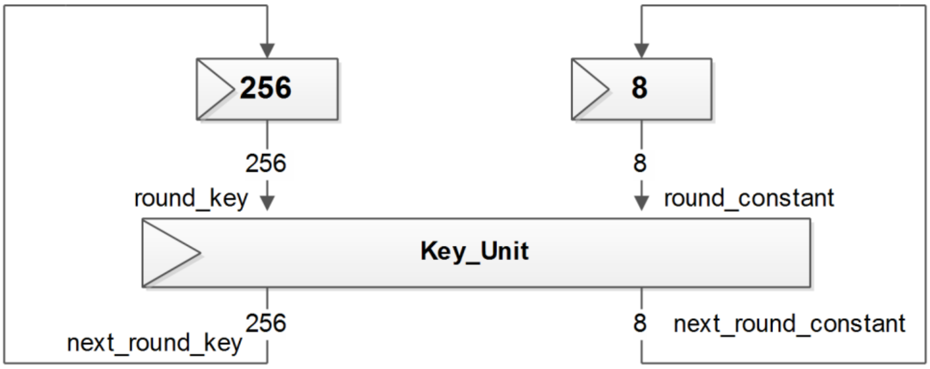
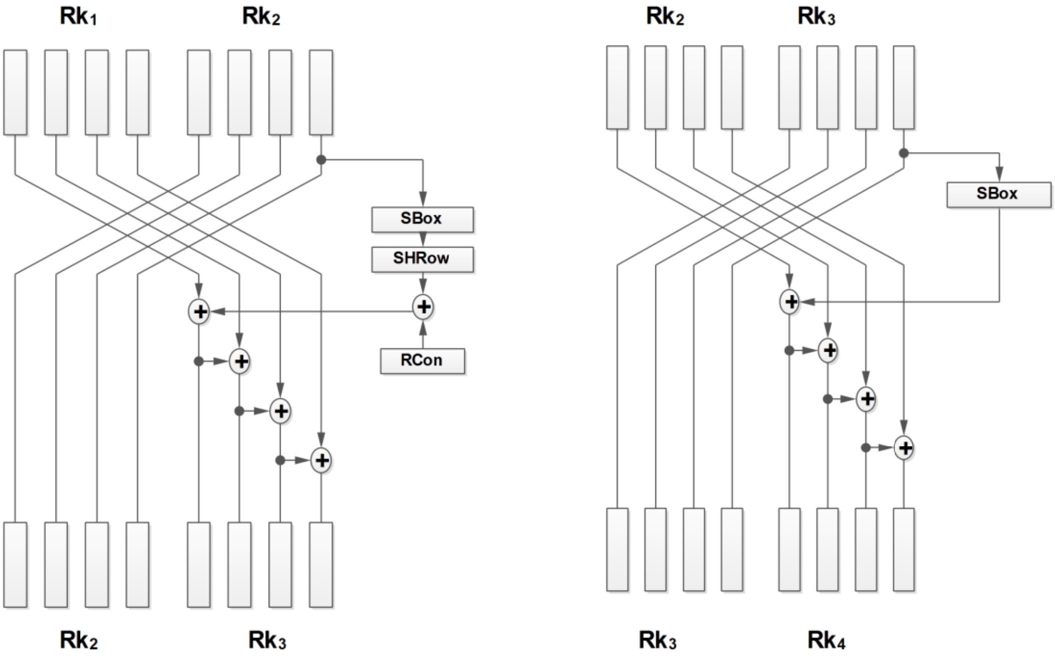


Figure 9

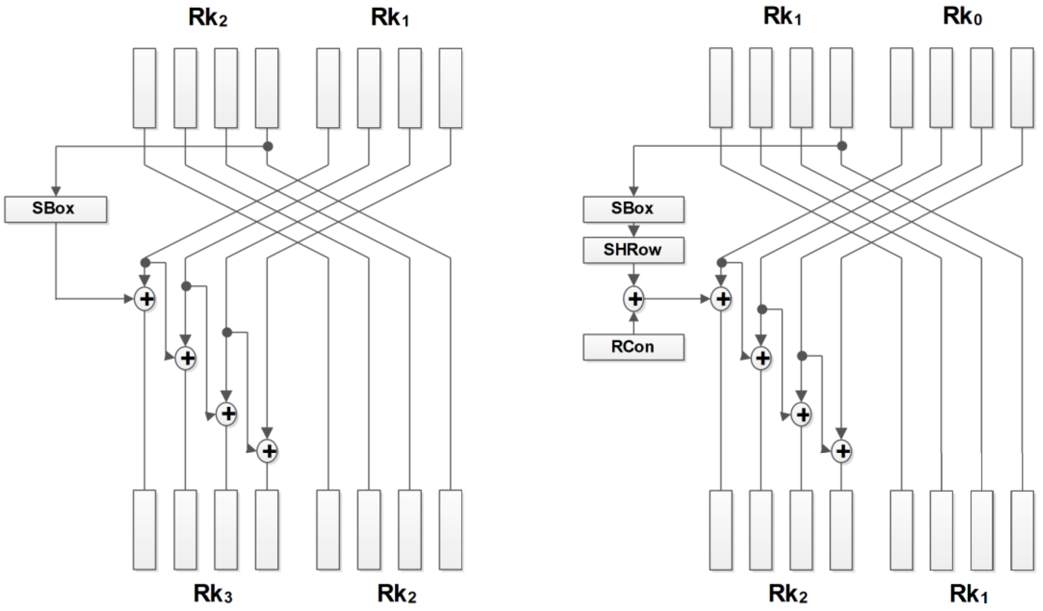
These workarounds can be best described by the Figure 9 where the right part represents the case of flag = 1 and the left part the flag = 0.



a) b)

Figure 10

The decryption process is similar to encryption except that now we use the least significant 128 bits to generate the next round key and the most significant 128 bits will now be shifted to the right with 128 positions in order concatenate the new round key to this msb bits. Similiar to the encryption we have to split the next round key generation in 2 cases: when we use only the least significant 128 bits and the other half will be shifted with 128 positions to the right so that these msb bits will now become the lsb bits, as Figure 11 depicts. Furthermore at every two rounds we will generate a complete new key data as the standard defines it. To implement the above mentioned strategy, we need to separate the two cases. This is why we used a flag on 1 bit, which will trigger the use of the first half of the key data when the flag is 0 and when will use the other half when it has the value 1. In the first case all the 4 operation on which AES is based on are performed and in the case with the flag on value 1, only the S-box is performed, as described by the AES standard.



a) b)

Figure 11

3.1 Design Validation

In order to verify the final results, a testbench module was designed. This unit contains the input vectors (the key data, the plaintext and the correct output data) and another unit called 'aes' which implements the whole algorithm and also the data communication protocol. The correctness of the design was assessed using the table vectors from the AES standard [1, 2]. If an error occurs in the first round, it will be propagated and amplified to the next round and so on till it reaches the final round yielding a completely different result. Another test strategy would be to choose the plaintext and just encrypt it with any encryption unit that was validated, then get the result and give it as an input for the decryption, in this way the result from decryption will have to be identical with the initial plaintext. Another important aspect is the fact that all the operation for the current round are done in parallel, in one single clock cycle, for each corresponding round.

4 Conclusions

In this paper we applied the concept of encryption and decryption based on the length of the key, from here resulting different kind of hardware schemes depending on the AES version (AES 192 or AES 256). We also presented in detail how the AES algorithm works, how the round keys are generated and how the 4 main operations on which AES is based operates. The AES design was implemented with success for 192 and 256 bits key, this being proved by the various input data which were encrypted using different keys of 192 and 256 bits and then the results were given as input for decryption, so that in the end we got the decryption result identical with the input data given at the beginning of the encryption.

Acknowledgement:
This article would not have been written without the support and help of my teachers coordinators: Flavius Opritoiu and Mircea Vladutiu.

References

[1] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, retrieved April 2013

[2] J. Daemen, V. Rijmen: “The design of Rijndael. AES - The Advanced Encryption Standard”, Springer, Heidelberg Berlin, 2002.

[3] F. Opritoiu, "Reliable implementations for Cryptographic systems with Testability facilities," PhD Thesis, Editura Politehnica - Timisoara, 2010

[4] P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core", in DSD ’06: Proceedings of the 9th EUROMICRO Conference on Digital System Design, pp. 577–583, IEEE Computer Society, 2006

[5] S. Morioka and A. Satoh, “An Optimized S-Box Circuit Architecture for Low Power AES Design,” in CHES ’02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems, pp. 172–186, Springer-Verlag, 2003

[6] D. Canright, “A Very Compact S-Box for AES,” in CHES, pp. 441–455, 2005

Laura CODREANU
Politehnica University Of Timisoara
Computer Science Department
Vasile Parvan, 3, Timisoara
ROMANIA
E-mail: codreanulaura15@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Image processing for skin detection by color

Aleksandra Cugalj, Marko Karovic
Teacher Coordinator: Milan Tuba

Abstract

Skin detection plays an important role in many applications, including face detection, human motion analysis, and objectionable image filtering. Skin detection is the process of finding skin-colored pixels and regions in an image or a video. This process is typically used as a preprocessing step to find regions that potentially have human faces and limbs in images. Several computer vision approaches have been developed for skin detection. Skin detectors typically transform a given pixel into an appropriate color space and then use a skin classifier to label the pixel whether it is a skin or a non-skin pixel. In this paper, an efficient method for skin color segmentation on color photos is implemented. The color image is transferred from the input RGB color space to YcbCr color space where skin color discrimination is more successful. After this transformation we have applied edge detection method to separate skin region and non-skin region. However, the presence of light, shadows, noise etc. can affect the efficiency of skin color recognition.

1 Introduction

Image processing is a physical process used to convert an image signal, either digital or analog, into a physical image. The actual output itself can be an actual physical image or the characteristics of an image. The most common type is photography, in which an image is captured using a camera to create a digital or analog image. In order to produce a physical picture, the image is processed using the appropriate technology based on the input source type.

In digital photography, the image is stored as a computer file. This file is translated using photographic software to generate an actual image. The colors, shading, and nuances are all captured at the time the photograph is taken, and the software translates this information into an image.

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

Skin color detection is a very popular and useful technique for detecting and tracking human-body parts. It receives much attention mainly because of its wide range of applications such as, face detection and tracking, naked people detection, hand detection and tracking, people retrieval in databases and Internet, etc. The main goal of skin color detection is to build a decision rule that will discriminate between skin and non-skin pixels. Identifying skin colored pixels involves finding the range of values for which most skin pixels would fall in a given color space. In general, a good skin color model must have a high detection rate and a low false positive rate. That is, it must detect most skin pixels while minimizing the amount of non-skin pixels classified as skin. [1]

2 Color Models for Skin Color Classification

There are various color spaces such as, RGB (Red Green Blue), NTSC, YCbCr, HSV, CMY, HIS, YUV, CIE-Lab etc. The choice of color space can be considered as the primary step in skin color classification.
Most of the research efforts on skin detection have focused on visible spectrum imaging. Skin color detection in visible spectrum can be a very challenging task as the skin color in an image is sensitive to various factors such as:

- Illumination: A change in the light source distribution and in the illumination level (indoor, outdoor, highlights, shadows, non-white lights) produces a change in the color of the skin in the image (color constancy problem). The illumination variation is the most important problem among current skin detection systems that seriously degrades the performance.
- Camera characteristics: Even under the same illumination, the skin-color distribution for the same person differs from one camera to another depending on the camera sensor characteristics. The color reproduced by a CCD camera is dependent on the spectral reflectance, the prevailing illumination conditions and the camera sensor sensitivities.
- Ethnicity: Skin color also varies from person to person belonging to different ethnic groups and from persons across different regions. For example, the skin color of people be-longing to Asian, African, Caucasian and Hispanic groups is different from one another and ranges from white, yellow to dark.
- Individual characteristics: such as age, sex and body parts also affects the skin-color appearance.
- Other factors: Different factors such as subject appearances (makeup, hairstyle and glasses), background colors, shadows and motion also influence skin-color appearance.

The choice of color space can be considered as the primary step in skin-color classification. The RGB color space is the default color space for most available image formats. Any other color space can be obtained from a linear or non-linear transformation from RGB.

The color space transformation is assumed to decrease the overlap between skin and non-skin pixels thereby aiding skin-pixel classification and to provide robust parameters against varying illumination conditions. It has been observed that skin colors differ more in intensity than in chrominance [2]. Hence, it has been a common practice to drop the luminance component for skin classification. Several color spaces have been proposed and used for skin detection. In this section, we review the most widely used color spaces for skin detection and their properties.

2.1 RGB Color Space

The RGB color space consists of three additive primaries: red, green and blue components of these colors combine additively to produce a resultant color.

The RGB model is represented by a 3-dimensional cube with red green and blue at the corners on each axis (Figure 1). Black is at the origin. White is at the opposite end of the diagonal of the cube. The gray scale follows

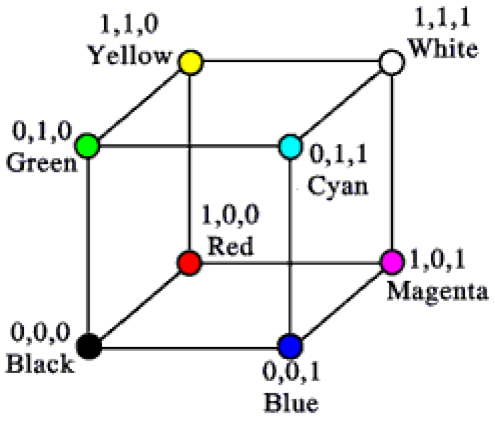


Figure 1. RGB Color Space

the diagonal from black to white. In a 24-bit color graphics system with 8 bits per color channel, red is (255, 0, 0). On the color cube, it is (1, 0, 0).The RGB model simplifies the design of computer graphics systems but is not ideal for all applications. The red, green and blue color components are highly correlated. This makes it difficult to execute some image processing algorithms. Many processing techniques, such as histogram equalization, work on the intensity component of an image only.

YCbCr color space (Figure 2) has been defined in response to increasing demands for digital algorithms in handling video information, and has since become a widely used model in a digital video. It belongs to the family of television transmission color spaces.

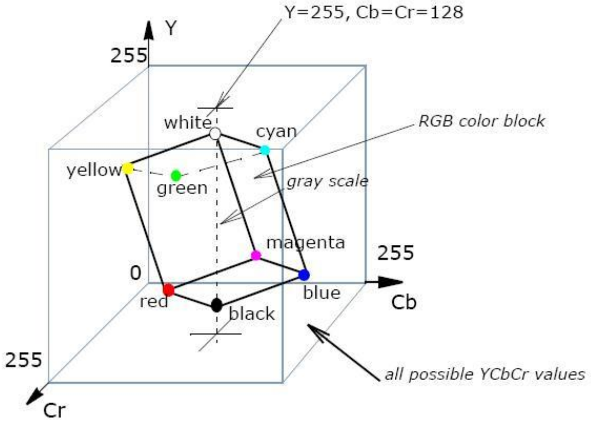


Figure 2. YCbCr Color Space

The family includes others such as YUV and YIQ. YCbCr is a digital color system, while YUV and YIQ are analog spaces for the respective PAL and NTSC systems. These color spaces separate RGB (Red-Green-Blue) into luminance and chrominance information and are useful in compression applications however the specification of colors is somewhat unintuitive.

3 Algorithms

3.1 Skin Color Detection in RGB Color Space

Crowley and Coutaz said one of the simplest algorithms for detecting skin pixels is to use skin color algorithm. The perceived human color varies as a function of the relative direction to the illumination. The pixels for skin region can be detected using a normalized color histogram, and can be further normalized for changes in intensity on dividing by luminance. And thus converted an [R, G, B] vector is converted into an [r, g] vector of normalized color which provides a fast means of skin detection. RGB color space does not separate luminance and chrominance, and the R,G, and B components are highly correlated. The luminance of a given RGB pixel is a linear combination of the R, G, and B values. Therefore, changing the luminance of a given skin patch affects all the R, G, and B components. In other words, the location of a given skin patch in the RGB color cube will change based on the intensity of the illumination under which such patch was imaged! This results in a very stretched skin color cluster in the RGB color cube.

Skin color cluster in RGB color space (R,G,B) is classified as skin if [3]:

$$R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and}$$
$$\max\{R,G,B\}-\min\{R,G,B\} > 15 \text{ and}$$
$$|R-G| > 15 \text{ and } R > G \text{ and } R >$$

3.2 Skin Color Detection in YCbCr Color Space

We have implemented a skin color classification algorithm with color statistics gathered from YCbCr color space. Studies have found that pixels belonging to skin region exhibit similar Cb and Cr values. Furthermore, it has been shown that skin color model based on the Cb and Cr values can provide good coverage of different human races. The thresholds be chosen as [Cr, Cr] and [Cb, Cb], a pixel is classified to have skin tone if the values [Cr, Cb] fall within the thresholds. The skin color distribution gives the face portion in the color image. This algorithm is also having the constraint that the image should be having only face as the skin region.

The Recommendation 601 specifies 8 bit (i.e. 0 to 255) coding of YCbCr, whereby the luminance component Y has an excursion of 219 and an offset of +16. This coding places black at code 16 and white at code 235. In doing so, it reserves the extremes of the range for signal processing foot-room and headroom. On the other hand, the chrominance components Cb and Cr have excursions of +112 and offset of +128, producing a range from 16 to 240 inclusively.

First we need to convert RGB color values to YCbCr color values.

To convert RGB color values to YCbCr color values we can use following formula [4]:

$$Y = (0.257*R) + (0.504*G) + (0.098*B) + 16$$
$$Cr = (0.439*R) - (0.368*G) - (0.071*B) + 128$$
$$Cb = -(0.148*R)- (0.291*G) + (0.439*B) + 128$$

Using these values we can classify (Y, Cr, Cb) as skin if [5]:

$$Y > 80$$

$$85 < Cb < 135$$
$$135 < Cr < 180$$

In the case of the YCbCr color space, we noticed that the intensity value Y has little influence on the distribution in the CbCr plane and that sample skin colors form a small and very compact cluster in the CbCr plane. Wang and Chang in [6] performed skin colors classification directly in the chrominance plane (CbCr) without taking the intensity value into account. A Bayesian decision rule for minimum cost is applied in order to classify a color into the skin color or the non-skin color class.

4 Proposed System

Figure 3 shows proposed system which has five stages: Input color image, Choosing suitable color space, Convert the color image into YCbCr color space, Binary Skin Detection and Output Image.

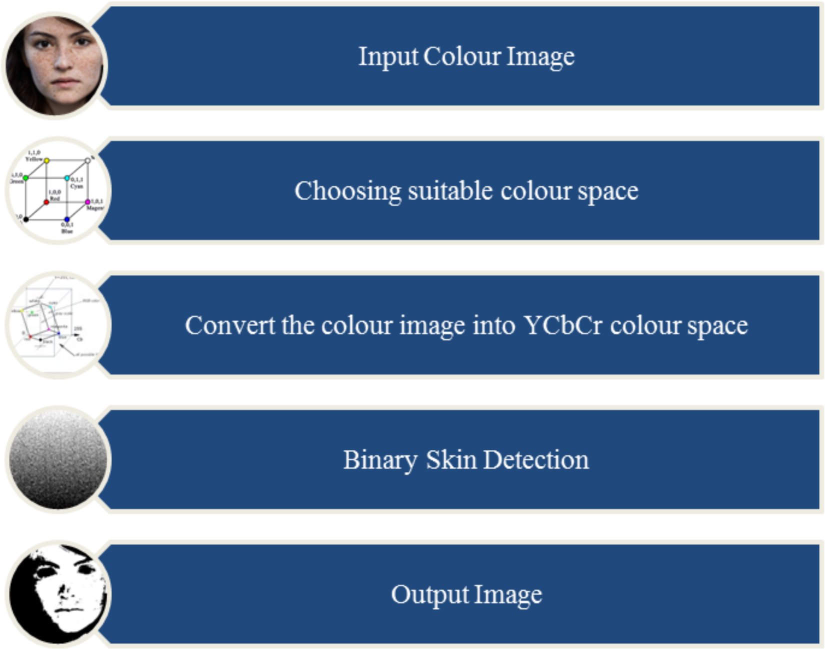

Input Colour Image
Choosing suitable colour space
Convert the colour image into YCbCr colour space
Binary Skin Detection
Output Image

Figure 3: Proposed System

First, we take an image we would like to experiment with.
The second step is to choose a suitable color space, for our paper we chose YCbCr because of the following reasons:

- YCbCr is one of two primary color spaces used to represent digital component video (the other is RGB).
- The difference between YCbCr and RGB is that YCbCr represents color as brightness and two color difference signals, while RGB represents color as a combination of red, green and blue.
- The luminance component (Y) of YCbCr is independent of the color, so it can be adapted to solve the illumination variation problem and it is easy to program.

- YCbCr has the smallest overlap between skin and non-skin data in under various illumination conditions.

The next step is to convert the image into YCbCr color space. We explained the process and formulas in Section 3 of this paper.

Classical pixel-based image classification automatically categorizes all pixels in an image into land cover classes or themes in a pixel by pixel manner. Usually, multispectral data are used and the spectral pattern present within the data for each pixel is used as the numerical basis for categorization. The classical pixel-based methods are minimum-distance/nearest neighbour, parallelepiped and maximum likelihood classifiers (MLC), whose detailed information can be found in Lillesand and Kiefer (1994).

5 Experimental Results

In this section the results achieved by using the proposed software are presented. The proposed algorithm is implemented using the C++ programming language. The input color image should be in RGB format with color intensity values ranging from 0 to 255. Due to restrictions on speed and performance, this project used images no larger than 1023x1023 pixels in size and needs to be in the Microsoft Windows bitmap (BMP) format.
The input images for testing are classified in: portrait, tattoo on skin, Hispanic, Caucasian and swimming suit costume. The first example is portrait (Figure 4)

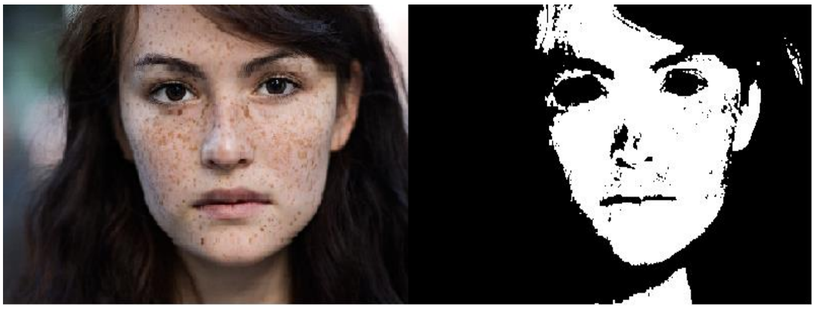


Figure 4: Portrait

The first thing we did is we took pixel samples from different body parts from all of the images that we are working with [Table 1]. We then concluded that the ranges of RGB color values are much broader than the ranges that we have in our system [Table 2]. That means that the RGB system will recognize far more pixels as skin, even if they are not, while YCbCr will be more precise. From our experimentation we came to a result that the range of RGB color values for skin pixels are R[182, 236] G[137, 207] B[106, 200], and for YCbCr they are Y[201, 254], Cb[125, 138], Cr[117, 130].

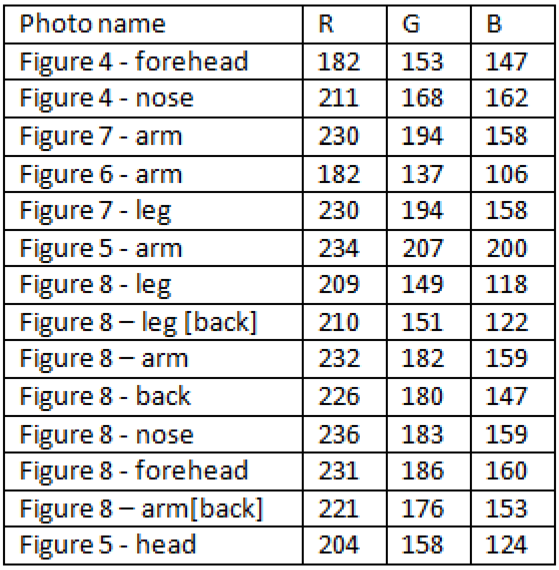


Photo name	R	G	B
Figure 4 - forehead	182	153	147
Figure 4 - nose	211	168	162
Figure 7 - arm	230	194	158
Figure 6 - arm	182	137	106
Figure 7 - leg	230	194	158
Figure 5 - arm	234	207	200
Figure 8 - leg	209	149	118
Figure 8 – leg [back]	210	151	122
Figure 8 – arm	232	182	159
Figure 8 - back	226	180	147
Figure 8 - nose	236	183	159
Figure 8 - forehead	231	186	160
Figure 8 – arm[back]	221	176	153
Figure 5 - head	204	158	124

Table 1: Pixels from testing images

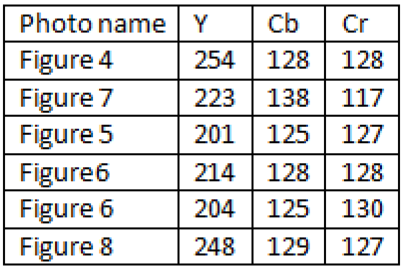


Photo name	Y	Cb	Cr
Figure 4	254	128	128
Figure 7	223	138	117
Figure 5	201	125	127
Figure6	214	128	128
Figure 6	204	125	130
Figure 8	248	129	127

Table 2: Values in YcbCr System

First, the input image is converted from the RGB color space to the YCbCr color space. This discriminates the objects in which we have no interested from the input image. After this stage, skin detection is used to identify the areas that contain some skin tone, and in this way we only get the parts of the image which contains skin.

It is not possible to completely isolate skin color using information from color space segmentation alone. If the intervals are too restrictive, many skin tones would not be detected, whereas if they are too wide, the chances of false positives dramatically increase. In order to solve this problem, Forsyth and Fleck [7] uses geometrical features to complement the skin filter.

Previous studies [8, 9] have found that pixels belonging to skin region exhibit similar Cb and Cr values. Furthermore, it has been shown that skin color model based on Cb and Cr values can provide good coverage of all human races. This is based on the conjecture that the different skin color that viewers perceived from the image cannot be differentiated from the chrominance information of that image region [10]. The apparent difference in skin color that viewers perceived is mainly due to the darkness or fairness of the skin. We showed how our method recognized skin pixels of Hispanic man who has tattoos on his back and his arm [Figure 5]. On the other side we showed Caucasian man sitting on the beach [Figure 6]. In both cases the algorithm shows high precision in recognizing skin pixels, with a small false positive rate.

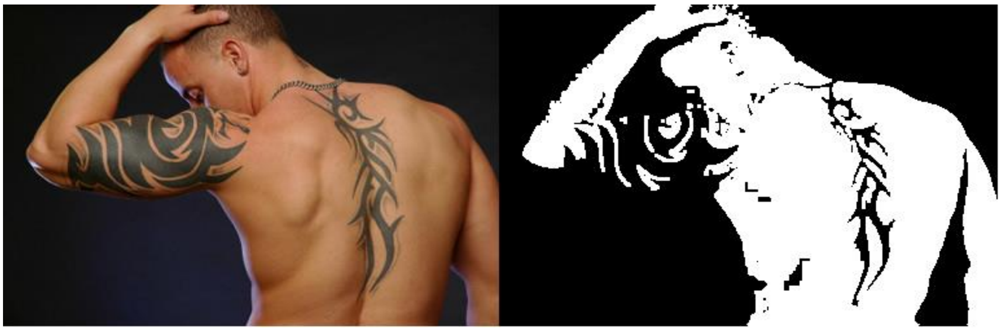


Figure 5: Hispanic man with the tatoos

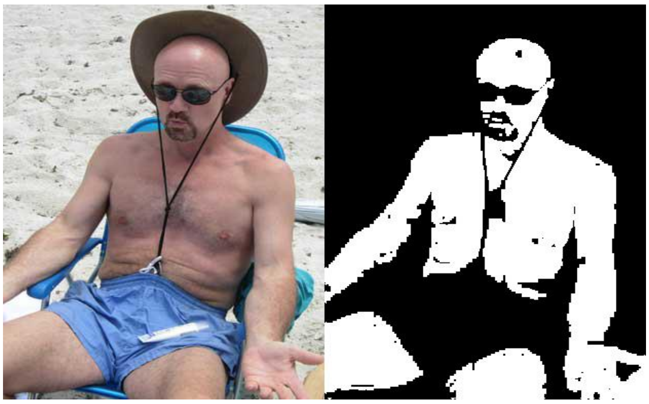


Figure 6: Caucasian man sitting on the beach

In these types of algorithms, where we get a binary result, it is possible for the system to recognize certain surfaces as skin, even though they are not (hair, sand, brick, etc) as it's shown in the Figure 7. In these situations we can use morphological operations such as erosion and dilation.

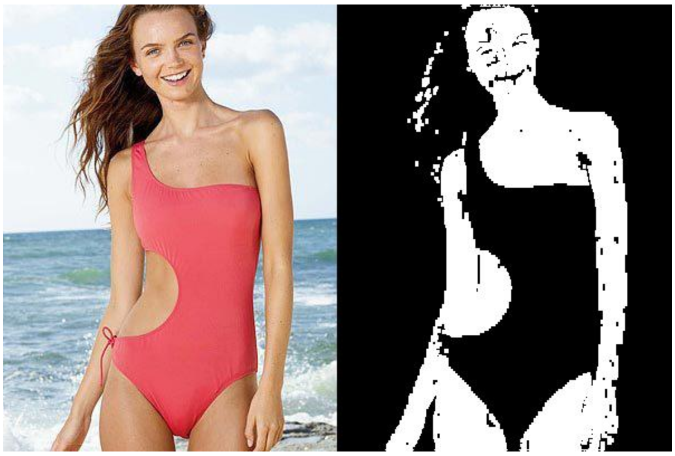


Figure 7: Caucasion woman dressed in a oddly shaped swimming suit

Dilation is a process that translates the origin of the structuring element throughout the domain of the image and checks to see whether it overlaps with 1-valued pixels. The output image is 1 at each location of the origin of the structuring element if the structuring element overlaps at least one 1-valued pixel in the input image [11].

Erosion process is similar to dilation process. Erosion shrinks or thins a binary image or object. As in dilation, the manner and extent of shrinking is controlled by the structuring element. The output of erosion has a value 1 at each location of the origin of the structuring element, such that the structuring element overlaps only 1-valued pixels of the input image.

By eroding and dilating after skin-color classification, it is a general method to reduce noise in a skin-color of bitmap. But when running up against big and a little big noise, the erosion or dilation can't effectively remove noise at a time and then the noise will be dilated again. Hence, it isn't a good method to reduce many redundant areas such as a little large missing inside of the face (as cause of the lighting in nose and forehead areas) and large (as skin-color background) outside of the facial location.

Based on [12] we assume that the skin color is determined mostly by its chrominance components Cb and Cr so the luminance component depends only on lightning conditions. There is also assumption that the skin color of various human individuals and also human skin color types is the same in its chrominance components. The difference is only according to the luminance parameter variation. This implicates that there is no need to trace luminance component probability function in skin color detection in YCbCr color space. Advantage of this method is its simplicity and minor computational costs. Our best result is represented in the Figure 8.

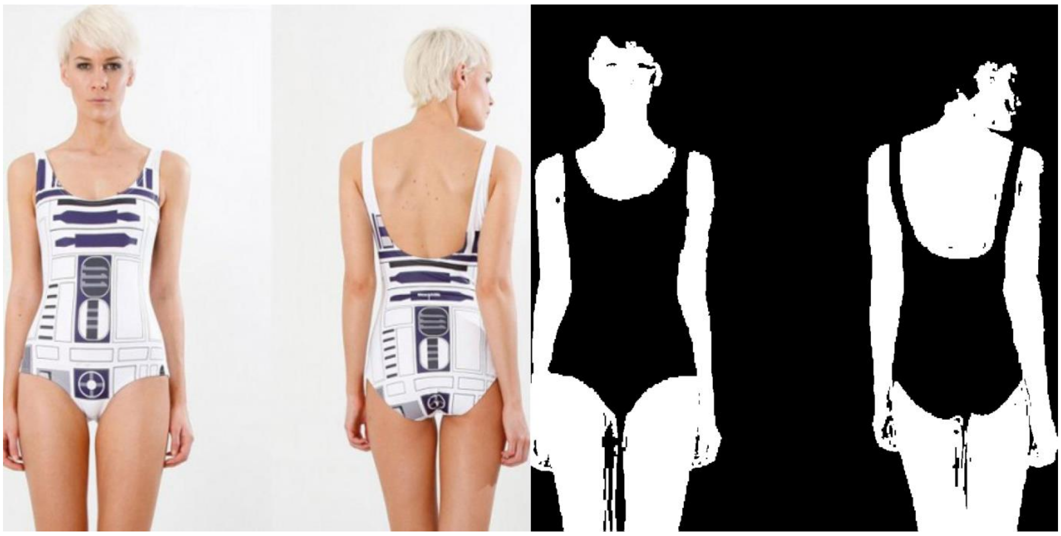


Figure 8: Caucasian woman dressed in a swimming suit

5 Conclusion

We are all in the midst of a revolution ignited by fast development in computer technology and imaging. Contrary to common belief, computers are not able to match humans in calculation related to image processing and analysis. But with increasing sophistication and power of the modern computing, computation will go beyond conventional, Von Neumann sequential architecture and would contemplate the optical execution too. Parallel and distributed computing paradigms are anticipated to improve responses for the image processing results.
Evaluation of color space goodness ’in general’ by assessing skin/non-skin overlap, skin cluster shape, etc. regardless to any specific skin modelling method cannot give the impression of how good is the color space suited for skin modelling, because different modelling methods react very differently on the color space change.

The experimental results showed that our approach in modelling skin color was able to achieve a good detection success rate. The algorithm gives computationally a very efficient as well as quite an accurate approach for skin detection which may be applied in real time.

References

[1] J. Kovac, P. Peer, F. Solina, Human skin color clustering for face detection, EUROCON2003, 2003.
[2] J. Yang, W. Lu, A. Waibel, Skin-color modeling and adaptation, ACCV98, 1998.

[3] Tarek M. Mahmoud, “A New Fast Skin Color Detection Technique”, In World Academy of Science, Engineering and Technology, 43, 2008

[4] M. Sharma, S. Verma, A. S. Mandal, “Block Based Skin Color Detection for Automated Video Surveillance System”, International Journal of Scientific and Engineering Research, vol. 3, issue 11, pp. 2, (November 2012)

[5] M. Sharma, S. Verma, A. S. Mandal, “Block Based Skin Color Detection for Automated Video Surveillance System”, International Journal of Scientific and Engineering Research, vol. 3, issue 11, pp. 2, (November 2012)

[6] H. Wang and S.-F. Chang, “A highly efficient system for automatic face region detection in MPEG video,” IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 4, pp. 615–628, 1997.

[7] D. Forsyth and M. Fleck, “Automatic Detection of Human Nudes”. Int. J. of Computer Vision 32/1, pp: 63-77, 1999.

[8] D. Chai and K. N. Ngan, “Locating facial region of a head-and-shoulders color image,” in Third IEEE International Conference on Automatic Face and Gesture Recognition (FG ’98), Nara, Japan, ,pp. 124-129, Apr. 1998.

[9] D. Chai and K. N. Ngan, “Face segmentation using skin-color map in videophone applications,” IEEE Trans. on Circuits and Systems for video Technology, vol. 9, no. 4, pp. 551-564, Jun. 1999.

[10] A. Abadpour and S. Kasaei, Pixel-based skin detection for pornography filtering, Iranian Journal of Electrical and Electronic Engineering, 1(3)(2005) 21-41.

[11] Zarit, B. D., Super, B. J.,Andoquek, F. K. H. 1999. Comparison of five color models in skin pixel classification. InICCV’99 Int’l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, 58–63

[12] Crowley, J. L. and Coutaz, J., “Vision for Man Machine Interaction,” Robotics andAutonomous Systems, Vol. 19, pp. 347-358 (1997).

Cugalj Aleksandra
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Beograd
Serbia
E-mail: fuzefightforfreedom@yahoo.com

Karovic Marko
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Beograd
Serbia
E-mail: gamzi91@yahoo.com

prof. Milan Tuba
University Megatrend of Belgrade
Faculty of Computer Science
Bulevar Umetnosti 29, N. Beograd
SERBIA
E-mail: tuba2@ieee.org

Actes de la Troisième Conférence Internationale
d'Informatique pour les Eétudiants
Imagination, Créativité, Design, Développement
ICDD 2013, 16-18 mai
Roumanie, Sibiu

Modèle flou pour l'évaluation des connaissances linguistiques et sa mise en œuvre par ordinateur

Dorokhov Mykhailo, Honcharova Mariia
Superviseurs scientifiques: Malyarets Lyudmila, Lembik Snezhana

Abstrait

Le problème de l'amélioration de la qualité de l'évaluation des connaissances en langues utilisant la logique floue sont analysés. Les critères de base de l'évaluation sont sélectionnés et formalisés. Les règles décisions pour déterminer la note finale des compétences linguistiques et des connaissances sont formulées. Le modèle informatique approprié dans l'outil FuzzyTech est développé.

1 Introduction

Dans les conditions de la réformation du système éducatif dans les pays de l'Est, dirigée à la hausse de son niveau, à l'obtention des standards européens, à la garantie des connaissances correspondants au niveau global du développement des technologies et de la société en général, deviennent actuelles les questions de l'évaluation objective, impartiale, basée sur les calculs des connaissances des étudiants et des élèves. Il est important d'étudier et de rendre le plus impartial le processus de l'évaluation des connaissances lui-même, en particulier, élaborer les moyens et les instruments objectifs pour l'évaluation des connaissances des matières linguistiques.

1.1 La mondialisation et la place du français dans le monde

Comme on le voit, les processus de la mondialisation dans le monde actuel sont généraux et universels. Donc, le côté le plus important de l'éducation contemporaine globalisée c'est la connaissance des principales langues mondiales parmi lesquelles se trouve, bien sûr, le française.

Le français acquiert de plus en plus de popularité dans le monde comme une langue étrangère à apprendre. Il est la langue officielle de l'Organisation des Nations Unies et de ses Instituts, la langue officielle dans 29 pays, il est utilisé dans 60 pays du monde. Des millions de gens sur les continents différents le connaissent et l'utilisent dans leur communication, ce qui le met à la deuxième place selon ce critère. Beaucoup d'œuvres scientifiques et littéraires de valeur mondiale sont écrites en français. Actuellement le français entre dans la première dizaine des langues des sites web sur Internet, il est l'une des langues les plus répandues de la communication Internet. Cette langue occupe la 4-me place à l'Union européenne selon le nombre de la population pour qui elle est maternelle.

Comme on le sait, l'indice le plus important des connaissances d'une langue étrangère est le test. Maintenant c'est indispensable pour presque tous les aspects de l'activité humaine au travail, à la maison, dans la société. Pour le français les deux tests les plus connus et les plus répandus qui permettent d'évaluer officiellement le niveau des connaissances sont DELF (Diplôme d'études en langue française) et DALF (Diplôme approfondi de langue française).

1.2 La mise en élaboration de l'évaluation par ordinateur des connaissances du français

L'objet de cette recherche est le test et l'évaluation des connaissances (selon les résultats des tests) pendant l'apprentissage du français. Conformément à cela le but de la recherche est l'étude des normes de l'évaluation des compétences linguistiques et l'analyse des possibilités du développement sur leur base des instruments modernes et basés mathématiquement de l'objectivation de l'évaluation par ordinateur.

Avec cela les tâches suivantes doivent être réalisées:

- étudier la documentation sur les fondements mathématiques et des outils logiciels pour de tels modèles et techniques;
- démontrer la possibilité de simulation basée sur la théorie des ensembles flous et de la logique floue pour l'évaluation des connaissances linguistiques;
- construire le modèle correspondant de l'évaluation des résultats des tests de français, pour qoi il faut sélectionner et formaliser les critères de base, formuler les règles de prise de décision pour le calcul de la note intégrée des compétences linguistiques;
- mettre en oeuvre le modèle dans le logiciel FuzzyTech.

1.3 La structure générale et le moyen de la réalisation de la tâche à la base de la logique floue

Etudions plus détaillement le processus de la formalisation des critères de l'évaluation des compétences linguistiques par les fonctions de l'appartenance. D'habitude l'évaluation des résultats de l'apprentissage du français se réalise à la base des tests standards correspondants du niveau différent. Dans la pratique cela se réalise en tenant compte du but principal (le développement linguistique universel d'une personne), du contenu éducatif (compétences, savoirs, habitudes, activité créatrice, attitude émotionnelle), et sur la base de l'expérience personnelle du professeur.

Cependant, il faut noter ici que l'absence de formalisation et la nature individuelle de la décision sur le niveau des connaissance et des compétences linguistiques déterminent leur dépendance subjective de la personne qui effectue l'évaluation. De plus, il faut avoir assez de temps pour déterminer la note finale dans la situation, quand elle dépend de plusieurs paramètres influençants.

La meilleure façon de surmonter ce problème est le calcul du niveau intégrant des connaissances et des compétences linguistiques à l'aide de la modélisation floue. Les paramètres qui influent sur la note finale, doivent être présentés comme fonction d'appartenance, et les règles d'évaluation doivent être formalisées par les critères sélectionnés. Ensuite, il est indispensable de mettre en place le système de conclusions flou.

La séquence de la solution de l'évaluation de la connaissance linguistique décrite dans [1]. La réalisation de la tâche a été la suivante (en bref).

A la première étape de la construction du modèle nécessaire, on a étudié les documents normatifs et méthodologiques sur l'évaluation et on a interrogé les professeurs en qualité d'experts (les personnes qui prennent la décision).

Ensuite, On a identifié le degré d'influence des certaines caractéristiques du niveau des connaissances et des compétences linguistiques sur la note finale. On a obtenu les conclusions générales de cette influence. On a aussi élaboré l'arbre de paramètres représentant leur hiérarchie et leur influence réciproque. Chacun de ces paramètres a été formalisé par le nombre d'erreurs. Selon les résultats des recherches d'experts chacun des paramètres d'entrée et le paramètre de sortie (le niveau intégrant des connaissances et des compétences linguistiques) sont classés dans les classes de niveau et leurs intervalles numériques sont définis pour l'utilisation dans le système de modélisation floue. Ensuite, la construction d'ensembles de fonctions d'appartenance est déterminée pour chaque paramètre d'entrée et le paramètre de sortie.

A l'étape finale on a élaboré, testé et précisé le modèle informatique approprié dans la sphère du logiciel spécialisé FuzzyTech.

Etudions plus détaillement certaines étapes importantes de cette réalisation de la tâche.

2 Les niveaux des connaissances du français et leur test

Selon CECR il existe les niveaux suivants de la connaissance du français. Les descriptions citées des niveaux sont adoptées par le Centre international d'études pédagogiques (CIEP) [2].

Table 1

Utilisateur Elémentaire	A1	Peut comprendre et utiliser des expressions familières et quotidiennes ainsi que des énoncés très simples qui visent à satisfaire des besoins concrets. Peut se présenter ou présenter quelqu'un et poser à une personne des questions la concernant - par exemple, sur son lieu d'habitation, ses relations, ce qui lui appartient, etc. - et peut répondre au même type de questions. Peut communiquer de façon simple si l'interlocuteur parle lentement et distinctement et se montre coopératif.
	A2	Peut comprendre des phrases isolées et des expressions fréquemment utilisées en relation avec des domaines immédiats de priorité (par exemple, informations personnelles et familiales simples, achats, environnement proche, travail). Peut communiquer lors de tâches simples et habituelles ne demandant qu'un échange d'informations simple et direct sur des sujets familiers et habituels. Peut décrire avec des moyens simples sa formation, son environnement immédiat et évoquer des sujets qui correspondent à des besoins immédiats.
Utilisateur Indépendant	B1	Peut comprendre les points essentiels quand un langage clair et standard est utilisé et s'il s'agit de choses familières dans le travail à l'école, dans les loisirs, etc. Peut se débrouiller dans la plupart des situations rencontrées en voyage dans une région où la langue cible est parlée. Peut produire un discours simple et cohérent sur des sujets familiers et dans ses domaines d'intérêt. Peut raconter un événement, une expérience ou un rêve, décrire un espoir ou un but et exposer brièvement des raisons ou explications pour un projet ou une idée.
	B2	Peut comprendre le contenu essentiel de sujets concrets ou abstraits dans un texte complexe, y compris une discussion technique dans sa spécialité. Peut communiquer avec un degré de spontanéité et d'aisance tel qu'une conversation avec un locuteur natif ne comportant de tension ni pour l'un m pour l'autre. Peut s'exprimer de façon claire et détaillée sur une grande gamme de sujets, émette un avis sur un sujet d'actualité et exposer les avantages et les inconvénients de différentes possibilités.
Utilisateur Expérimenté	C1	Peut comprendre une grande gamme de textes longs et exigeants, ainsi que saisir des significations implicites. Peut s'exprimer spontanément et couramment sans trop apparemment devoir chercher ses mots. Peut utiliser la langue de façon efficace et souple dans sa vie sociale, professionnelle ou académique. Peut s'exprimer sur des sujets complexes de façon claire et bien structurée et manifester son contrôle des outils d'organisation, d'articulation et de cohésion du discours.
	C2	Peut comprendre sans effort pratiquement tout ce qu'il/elle lit ou entend. Peut restituer faits et arguments de diverses sources écrites et orales en les résumant de façon cohérente. Peut s'exprimer spontanément, très couramment et de façon précise et peut rendre distinctes de fines nuances de sens en rapport avec des sujets complexes.

Le Cadre européen commun de référence pour les langues : apprendre, enseigner, évaluer (CECR), résultat de deux décennies de recherches, est comme son nom l'indique clairement, un cadre de référence. Il a été conçu dans l'objectif de fournir une base transparente, cohérente et aussi exhaustive que possible pour l'élaboration de programmes de langues, de lignes directrices pour les curriculums, de matériels d'enseignement et d'apprentissage, ainsi que pour l'évaluation des compétences en langues étrangères. [3]

Pour l'évaluation des connaissances des apprenants le français et la confirmation de leurs compétences et savoirs linguistiques pour les niveaux A1, A2, B1, B2 on utilise le test DELF, qui présente la certification des compétences pour des locuteurs non natifs du français. Il est élaboré par CIEP pour le Ministère de l'Éducation nationale. Pour l'évaluation des connaissances plus profondes du français on utilise le test DALF.

On met les notes sur les deux tests selon le barème de 100 points. Si l'examiné reçoit 51 points sur 100, il réussit le test. Avant la passation du test le candidat choisit le niveau du test à passer et, dans le cas de la réussite, il obtient le diplôme confirmant les compétences linguistiques au niveau choisi. Les étudiants du FLE (français langue étrangère) choisissent le plus souvent le test du niveau A2. C'est que le niveau A1 trop bas pour le travail ou les études, B1 est parfois difficile pour les débutants. Pour le voyage de stage en France ou pour les programmes d'échange, le certificat DELF donne l'avantage et le plus souvent les candidats passent A2. En rapport avec cela, nous élaborerons et étudierons la réalisation du modèle sur l'exemple du test DELF du niveau A2.

3 L'étude de la structure du test DELFA2

Le test étudié se compose de quatre parties principales (schéma 1). Conformément à cela, la note finale pour le test est déterminée par quatre notes composantes, dont chacune influe sur la note finale de manière égale, c'est-à-dire chaque partie est évaluée sur 25 points.

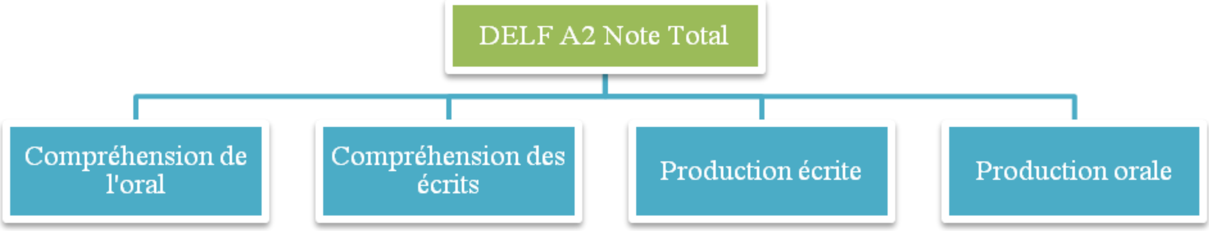


Fig. 1.Vue générale du test

Avec cela la note pour la compréhension de l'oral et pour la compréhension des écrits est déterminée exactement comme somme des exercices corrects. Les deux dernières parties présentent plus d'intérêt - Production écrite et Production orale. La particularité de ces parties consiste en ce qu'elles n'ont pas de critères précis de l'évaluation et, en tout cas, on a besoin d'un professeur expert pour l'évaluation des compétences dans ce type d'activité.

Ces deux dernières parties nous étudierons et analyserons. Ensuite, nous élaborerons le modèle approprié de multitude floue. Mais avant tout il faut construire la hiérarchie des critères dans la détermination de la note générale pour chaque partie du test.

2.1 Les critères de l'évaluation de la production écrite

Dans cette exercise etudiant dois savoir comment décrire un événement ou des expériences personnelles et écrire pour inviter, remercier, s'excuser, demander, informer, féliciter. La structure de cette partie est présentée dans le schéma 2.

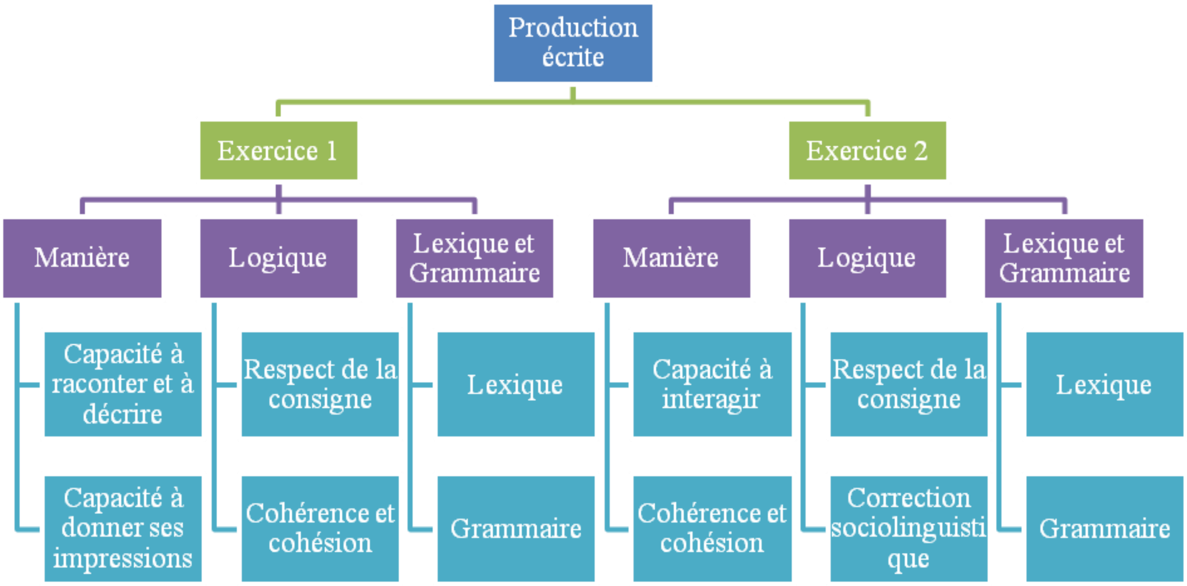


Fig. 2. Construction de Production écrite

2.1.1 Les critères de l'évaluation pour l'Exercice 1

Respect de la consigne - peut mettre en adéquation sa production avec la situation proposée. Peut respecter la consigne de longueur minimale indiquée.
Capacité à raconter et à décrire - peut décrire de manière simple des aspects quotidiens de son environnement (gens, choses, lieux) et des événements, des activités passées, des expériences personnelles.
Capacité à donner ses impressions - peut communiquer sommairement ses impressions, expliquer pourquoi une chose plaît ou déplaît.
Lexique/orthographe lexicale – peut utiliser un répertoire élémentaire de mots et d'expressions relatifs à la situation proposée. Peut écrire avec une relative exactitude phonétique mais pas forcément orthographique.
Morphosyntaxe/orthographe grammaticale - peut utiliser des structures et des formes grammaticales simples relatives à la situation donnée mais commet encore systématiquement des erreurs élémentaires.
Cohérence et cohésion - peut produire un texte simple et cohérent. Peut relier des énoncés avec les articulations les plus fréquentes.

2.1.1 Les critères de l'évaluation pour l'Exercice 2

Respect de la consigne - peut mettre en adéquation sa production avec la situation proposée. Peut respecter ta consigne de longueur minimale indiquée.
Correction sociolinguistique - peut utiliser les registres de langue en adéquation avec le destinataire et le contexte. Peut utiliser les formes courantes de l'accueil et de la prise de congé.
Capacité à interagir - peut écrire une lettre personnelle simple pour exprimer remerciements, excuses, propositions, etc.
Lexique/orthographe lexicale - peut utiliser un répertoire élémentaire de mots et d'expressions relatifs à la situation proposée. Peut écrire avec une relative exactitude phonétique mais pas forcément orthographique.
Morphosyntaxe/orthographe grammaticale - peut utiliser des structures et des formes grammaticales simples relatives à la situation donnée mais commet encore systématiquement des erreurs élémentaires.
Cohérence et cohésion - peut produire un texte simple et cohérent. Peut relier des énoncés avec les articulations les plus fréquentes.

2.1 Les critères de l’évaluation de la production orale

Production orale se compose de trois parties, ce sont entretien dirigé; monologue suivi; exercice en interaction. La structure des composants de l’évaluation de la production orale est présentée dans le schéma. 3

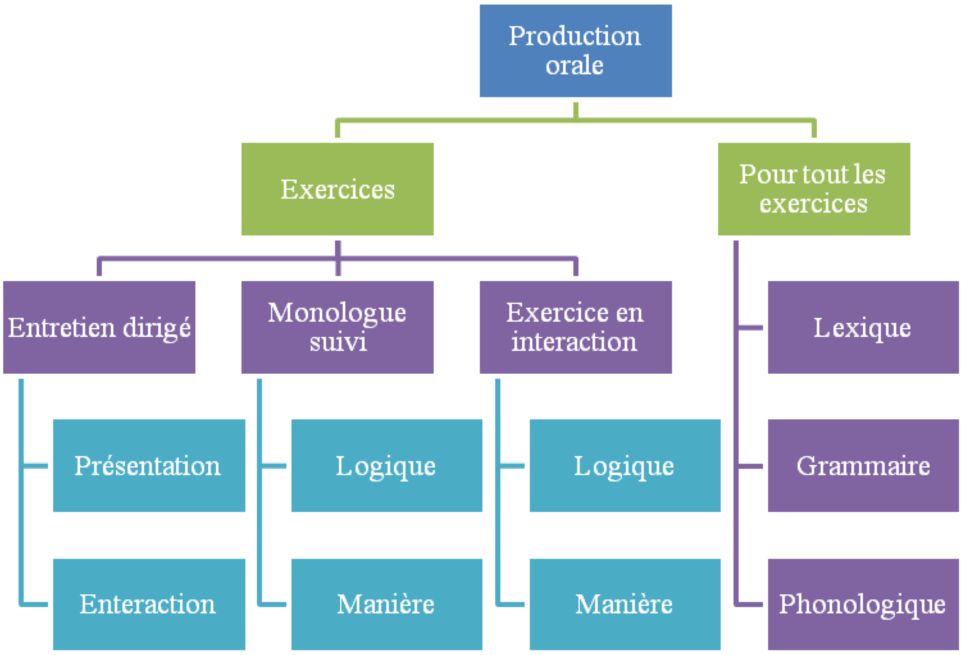


Fig. 3. Construction de Production orale

2.1.1 Les critères de l’évaluation pour l’ entretien dirigé

Présentation - peut établir un contact social, se présenter et décrire son environnement familier.
Enteracion - peut répondre et réagir à des questions simples. Peut gérer une interaction simple.

2.1.1 Les critères de l’évaluation du monologue suivi

Manière - peut présenter de manière simple un événement, une activité, un projet, un lieu etc. liés à un contexte familier.
Logique - peut relier entre elles les informations apportées de manière simple et claire.

2.1.1 Les critères de l’évaluation pour l’exercice en interaction

Logique - peut demander et donner des informations dans des transactions simplesde la vie quotidienne. Peut faire, accepter ou refuser des propositions.
Manière - peut entrer dans des relations sociales simplement mais efficacement en utilisant les expressions courantes et en suivant les usages de base.

2.1.1 Les critères de l’évaluation pour l'ensemble des 3 parties de l'epreuve

Lexique (étendue et maîtrise) - peut utiliser un répertoire limité mais adéquat pour gérer des situations courantes de la vie quotidienne.
Morphosyntaxe - peut utiliser des structures et des formes grammaticales simples. Le sens général reste clair malgré la présence systématique d'erreurs élémentaires.
Maîtrise du système phonologique - Peut s'exprimer de façon suffisamment claire. L'interlocuteur devra parfois faire répéter.

4 La réalisation du modèle dans FuzzyTech

A la première étape de l'élaboration du modèle on a formalisé les variables linguistiques à l'aide des fonctions d'appartenance correspondantes, leurs valeurs sont présentées dans la table 2.

Table 2

№	Code	Nom du paramètre	Faible niveau		Niveau moyen			Haut niveau	
			Égal 1	Réduire (1;0)	Augmenter (0;1)	Égal 1	Réduire (1;0)	Augmenter (0;1)	Égal 1
1	0	DELF A2 Note Total	0	45	15	50	75	50	100
2	1	Compréhension de l'oral	0	8	3	11	18	13	25
3	2	Compréhension des écrits	0	8	5	12	20	16	25
4	3	Production écrite	0	8.5	4.5	9	16	12	25
5	31	Exercice 1	0	4	2.5	5.5	9	6	13
6	311	Manière	0	2	1	2.4	5	3	6
7	3111	Capacité à raconter et à décrire	0	1	0	1.5	3	2	4
8	3112	Capacité à donner ses impressions	0	0.5	0.25	0.5	1.5	1	2
9	312	Logique	0	0.8	0.5	1	1.6	1.1	2.5
10	3121	Respect de la consigne	0	0.5	0.3	0.6	0.8	0.65	1
11	3122	Cohérence et cohésion	0	0.5	0.25	0.5	1.25	1	1.5
12	313	Lexique et Grammaire	0	2.3	1.25	2.75	3.5	2.8	4.5
13	3131	Lexique	0	0.5	0	0.75	1.5	1	2
14	3132	Grammaire	0	1	0.4	1.25	2	1.5	2.5
15	32	Exercice 2	0	5	3	6	9	7	12
16	321	Manière	0	1.5	1	2	3	2.5	5.5
17	3211	Capacité à interagir	0	1	0.25	2.2	3	2	4
18	3212	Cohérence et cohésion	0	0.5	0.25	0.9	1.25	0.8	1.5
19	322	Logique	0	0.65	0.35	0.75	1.25	1	2
20	3221	Respect de la consigne	0	0.35	0.15	0.4	0.8	0.5	1
21	3222	Correction sociolinguistique	0	0.25	0	0.25	0.7	0.5	1
22	323	Lexique et Grammaire	0	1.5	1	1.6	3	2	4.5
23	3231	Lexique	0	0.5	0.2	0.5	1.5	1	2
24	3232	Grammaire	0	1	0.5	1.6	2	1.4	2.5
25	4	Production orale	0	6	4	11	15	10	25
26	41	Exercices	0	6	4	7	11	8	15
27	411	Entretien dirigé	0	1.4	0.8	1.6	3	2.4	4
28	4111	Présentation	0	1.4	0.8	1.7	2.3	1.8	3
29	4112	Enteraction	0	0.45	0.25	0.4	0.75	0.55	1
30	412	Monologue suivi	0	1.6	0.9	2.15	3.4	2.3	5
31	4121	Logique	0	1.2	0.8	1.5	2.1	1.65	3
32	4122	Manière	0	0.4	0.15	0.5	1	0.75	2
33	413	Exercice en interaction	0	2	1	2.5	3.5	3	6
34	4131	Logique	0	1	0.5	1.5	2.5	2	4
35	4132	Manière	0	1	0.5	1.2	0.6	1.3	2
36	42	Pour tous les exercices	0	5	3	5.5	7.5	6	10
37	421	Lexique	0	1	0.55	1.2	2	1.4	3
38	422	Grammaire	0	1	0.6	1.6	2.5	1.7	4
39	423	Phonologique	0	1.4	0.8	1.80	2.3	1.85	3

Les valeurs numériques ont été reçues par l'interrogation des experts et son analyse statistique par les moyens décrits dans [4]. Sans la limitation de la communauté et le détriment de la sûreté du modèle, pour sa représentation simplifiée dans l'article, nous avons utilisé de linéaires fonctions d'appartenance triangulaires. Toutes les variables linguistiques dans l'outil logiciel FuzzyTech (im. 4) ont été présentées par les ensembles de trois termes dont la construction est décrite dans [5]. Les variables présentées dans la table et les graphiques peuvent être changées directement dans le régime graphique à l'aide de la souris ou bien dans les champs numériques correspondants de l'interface.

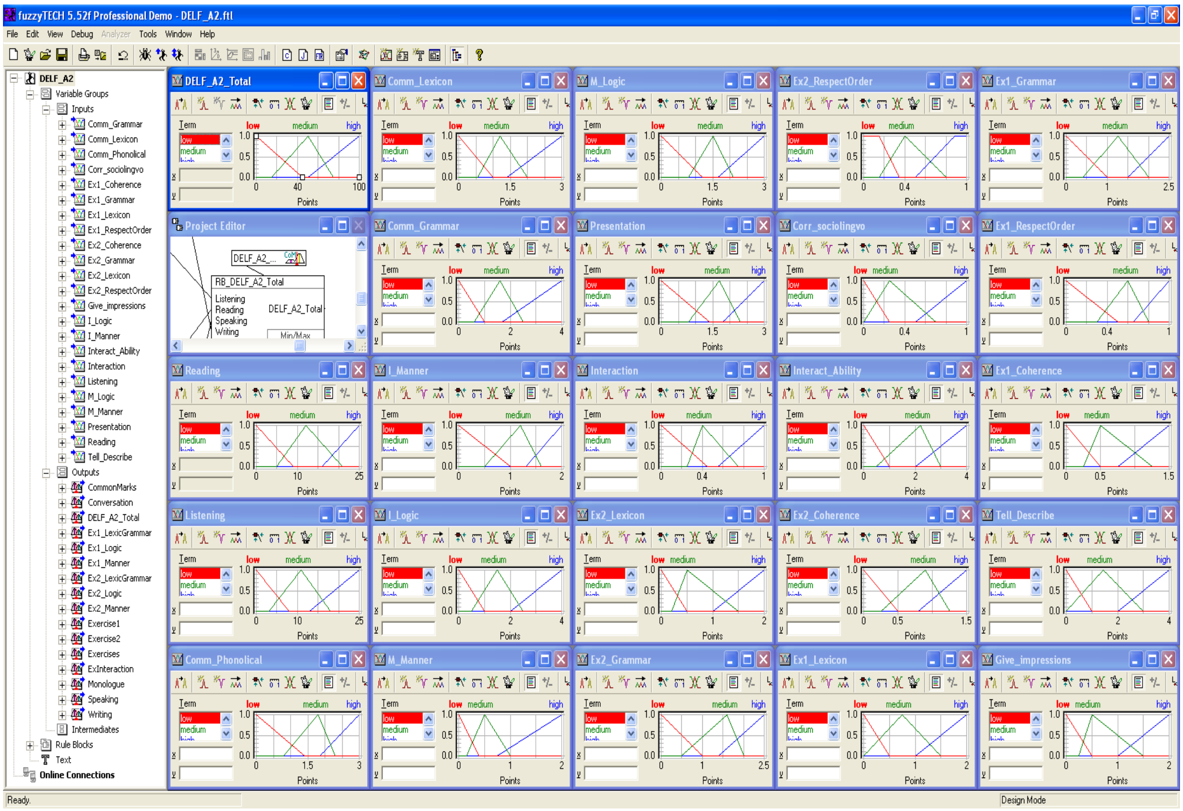
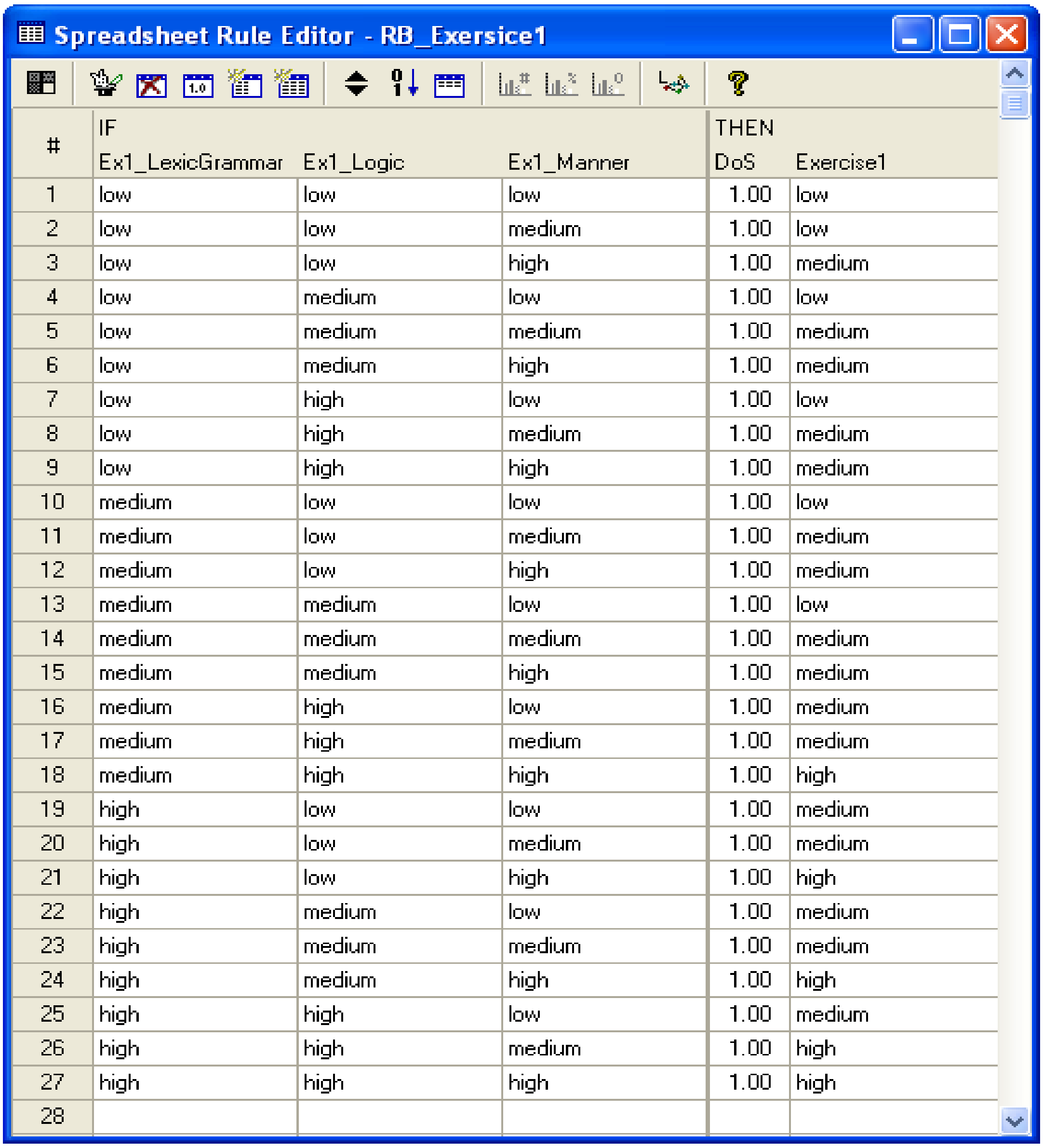


Fig. 4. Vue générale des fonctions d'appartenance du modèle

L'étape suivante de la construction du modèle c'est la construction des règles floues de sortie pour toutes les variables du modèle et la multitude de leurs termes linguistiques. Pour chaque ensemble "groupe de variables d'entrée – variable de sortie" on construit les règles de la conclusion floue (la décision sur la note). Avec cela on prend en compte toutes les combinaisons possibles des valeurs des variables linguistiques des paramètres d'entrée. La fenêtre de l'entrée des règles est présentée sur l'image 5.

Dans ce fragment est présenté l'exemple des règles floues de la prise de décision sur la détermination de la note du premier exercice selon trois critères d'entrée.



#	IF			THEN	
	Ex1_LexicGrammar	Ex1_Logic	Ex1_Manner	DoS	Exercise1
1	low	low	low	1.00	low
2	low	low	medium	1.00	low
3	low	low	high	1.00	medium
4	low	medium	low	1.00	low
5	low	medium	medium	1.00	medium
6	low	medium	high	1.00	medium
7	low	high	low	1.00	low
8	low	high	medium	1.00	medium
9	low	high	high	1.00	medium
10	medium	low	low	1.00	low
11	medium	low	medium	1.00	medium
12	medium	low	high	1.00	medium
13	medium	medium	low	1.00	low
14	medium	medium	medium	1.00	medium
15	medium	medium	high	1.00	medium
16	medium	high	low	1.00	medium
17	medium	high	medium	1.00	medium
18	medium	high	high	1.00	high
19	high	low	low	1.00	medium
20	high	low	medium	1.00	medium
21	high	low	high	1.00	high
22	high	medium	low	1.00	medium
23	high	medium	medium	1.00	medium
24	high	medium	high	1.00	high
25	high	high	low	1.00	medium
26	high	high	medium	1.00	high
27	high	high	high	1.00	high
28					

Fig. 5. Exemple de règles floues

La vue générale du modèle élaboré directement dans FuzzyTech est représentée dans le schéma 6. Avec cela «Compréhension de l'oral» et «Compréhension des écrits» ne sont pas détaillés, on fait entrer les valeurs déjà calculées, la noté générale est calculée à la base des règles floues.

La pression sur chaque élément de la structure donne l'accès aux paramètres correspondants, au champ correspondant des changements de caractéristiques de la variable choisie et aux règles de la prise de décisions sur l'évaluation de leur intéraction.

Après la description de toutes les variables, l'introduction de toutes les règles floues, l'introduction des données numériques d'entrée, toute l'information d'entrée et le résultat des calculs sont reflétés dans la fenêtre du calcul intéractif et de la correction qui est présentée sur l'image 7. Ici on peut voir le modèle dans l'action, changer de valeurs des variabls d'entrée et calculer celles de sortie.

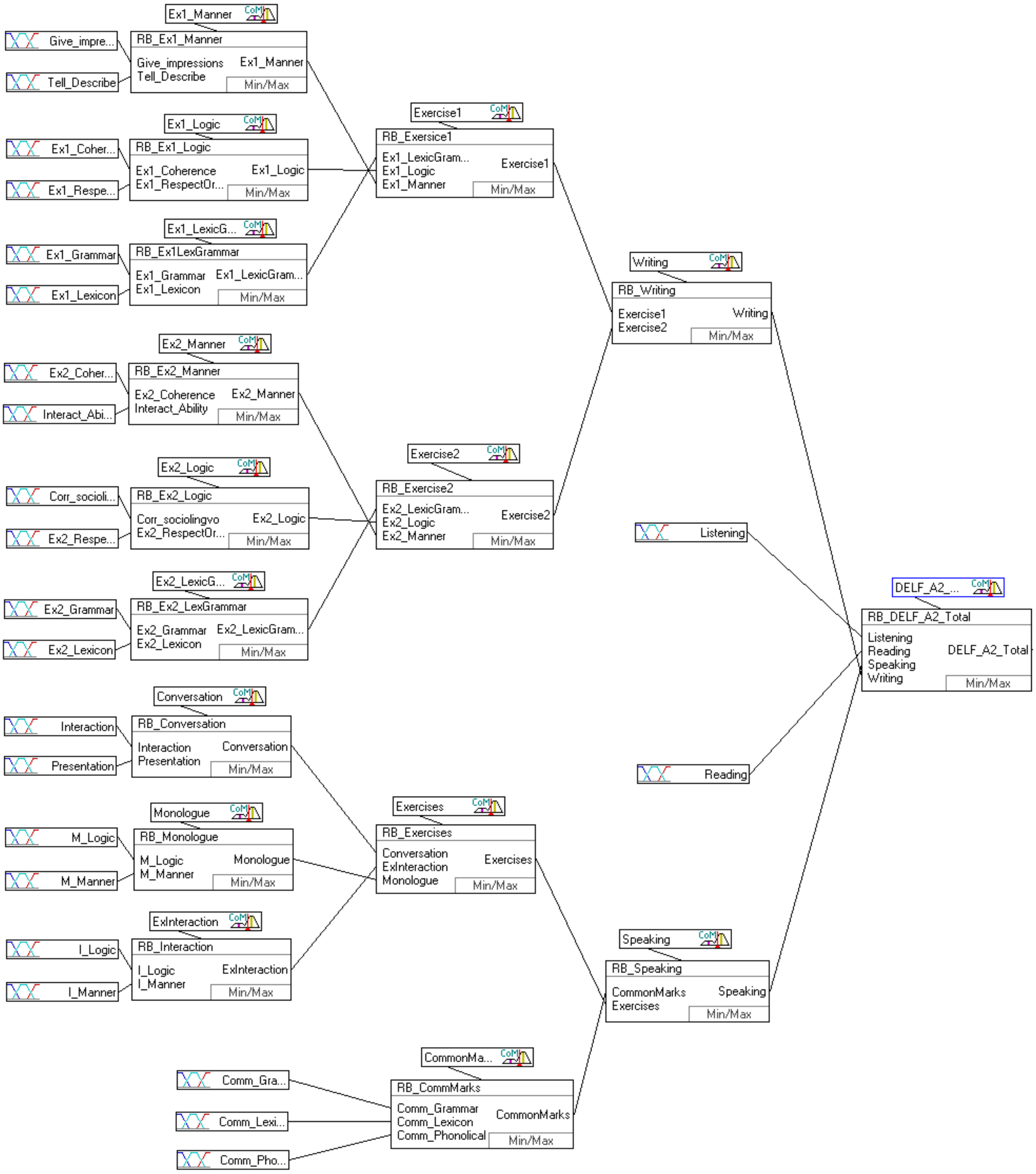
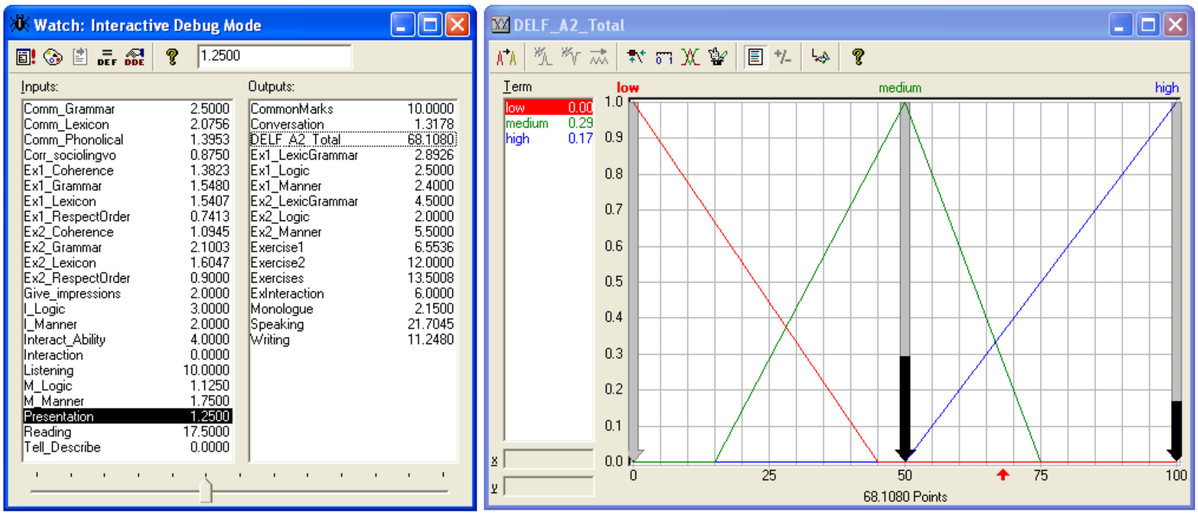


Fig. 6. Vue générale du modèle



Watch: Interactive Debug Mode

1.2500

Inputs:

Comm_Grammar	2.5000
Comm_Lexicon	2.0756
Comm_Phonolical	1.3953
Corr_sociolingvo	0.8750
Ex1_Coherence	1.3823
Ex1_Grammar	1.5480
Ex1_Lexicon	1.5407
Ex1_RespectOrder	0.7413
Ex2_Coherence	1.0945
Ex2_Grammar	2.1003
Ex2_Lexicon	1.6047
Ex2_RespectOrder	0.9000
Give_impressions	2.0000
I_Logic	3.0000
I_Manner	2.0000
Interact_Ability	4.0000
Interaction	0.0000
Listening	10.0000
M_Logic	1.1250
M_Manner	1.7500
Presentation	1.2500
Reading	17.5000
Tell_Describe	0.0000

Outputs:

CommonMarks	10.0000
Conversation	1.3178
DELF_A2_Total	68.1080
Ex1_LexicGrammar	2.8926
Ex1_Logic	2.5000
Ex1_Manner	2.4000
Ex2_LexicGrammar	4.5000
Ex2_Logic	2.0000
Ex2_Manner	5.5000
Exercise1	6.5536
Exercise2	12.0000
Exercises	13.5008
ExInteraction	6.0000
Monologue	2.1500
Speaking	21.7045
Writing	11.2480

DELF_A2_Total

Term

low	0.00
medium	0.29
high	0.17

Fig. 7. Démonstration d'un modèle

Pour l'analyse de la dépendance réciproque des paramètres on peut obtenir et utiliser les surfaces de la conclusion floue dont l'une est présentée dans le schéma 8. On représente ici la dépendance de la note «Manière» des critères «Capacité à raconter et à décrire» et «Capacité à donner ses impressions». Les surfaces analogues permettent d'analyser avec évidence les rapports entre les variables et les règles de la prise des décisions sur les niveaux de la note pour la correction et la précision du modèle.

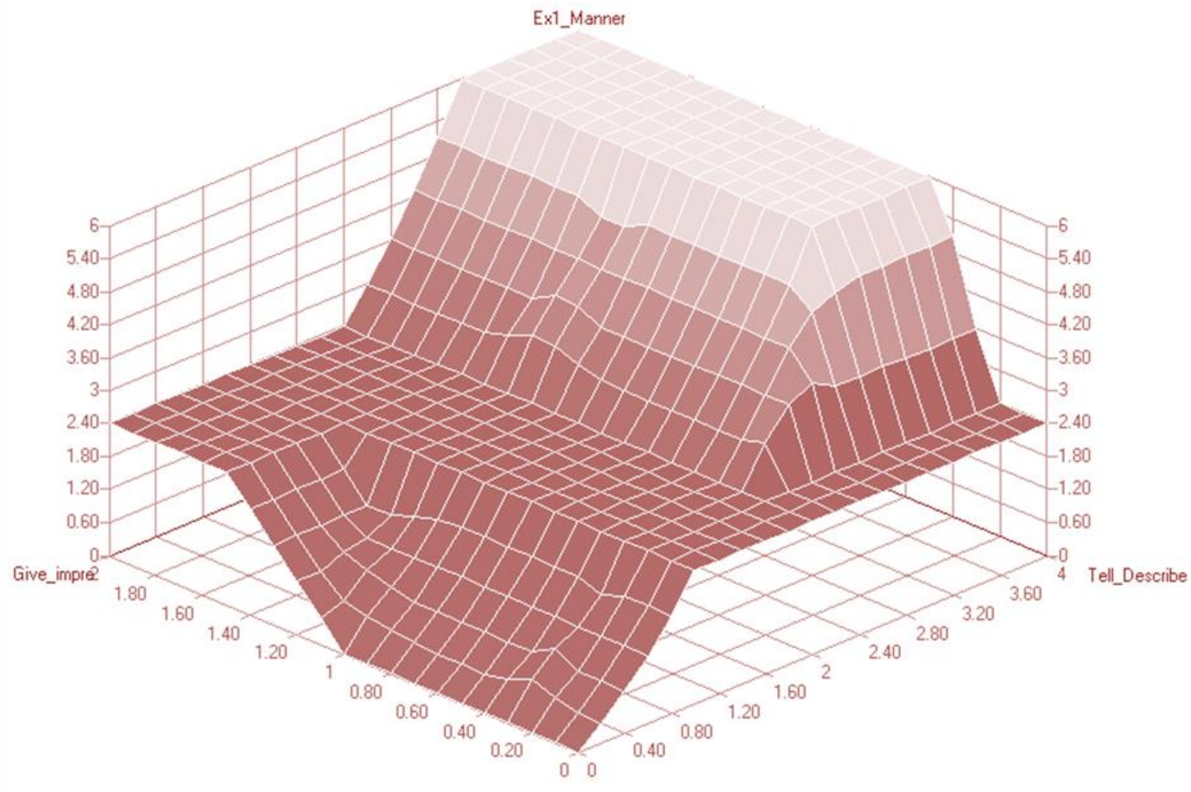


Fig. 8. Démonstration d'intrigue de dépendance

5 Conclusion

L'analyse plus détaillé de la modélisation numérique directe dans le modèle déjà construit sera l'objet de notre recherche suivante. On prévoit étudier les questions suivantes:

- la construction des fonction d'appartenance à la base des notes statistiques des experts;
- l'utilisation des formes plus complexes des fonctions d'appartenance;
- l'accordage du système élaboré de la conclusion floue avec l'utilisation des réseaux de neurones;
- l'utilisation des valeurs différentes des coefficients importants pour les différents paramètres d'entrée et intermédiaires et des règles correspondants de la prise des décisions;
- la construction des modules exécutants qui ne demandent pas à l'utilisateur final la présence du programme fuzzyTECH, par voie de transformation du modèle de fuzzyTECH au programme en langue de programmation C++ ou Java, et ensuite la compilation dans un code exécutant. Cela permettra de créer un programme autonome qui peut être utilisé par les examinateurs pendant la réalisation du test, en particulier, dans les mécanismes portables.

Donc, on a élaboré le modèle à plusieurs critères qui peut être largement utilisé pour le contrôle courant de tous les jours et pour le contrôle final du niveau de connaissance de la langue française. L'approche proposée peut être utiliées avec succès pour d'autres langues.

La base théorique du modèle est la théorie de multitudes floues, l'instrument pratique de sa construction c'est le logiciel spécialisé fuzzyTECH.

La méthode d'évaluation des connaissances d'une langue qu'on propose est peu exigeante, pratique, simple du point de vue de l'entrée des données d'utilisation et donne des résultats très précis. Il prend en compte un grand nombre des paramètres de l'évaluation du niveau des compétences linguistiques et aussi les règles correspondantes de la prise des décision sur les notes finales.

L'utilisateur peut présenter, obtenir, conserver les donnés et les résultats sous une forme claire et convenable pour l'analyse ultérieur. On peut aussi faire des calculs dans le régime de l'intéraction. Le modèle est implémenté dans l'environnement FuzzyTech sous une telle forme qui est facile à utiliser pour les utilisateurs (professeurs et élèves) afin de contrôler leur niveau de connaissance pendant l'apprentissage d'une langue.

References

[1] Fedorchenko V., Dorokhov M. Fuzzy set model for the assessment of language knowledge and its computer implementation in MATLAB // Systems of information processing. – Kharkiv (Ukraine): KhUAF, 2010, – Issue 9(90), pp. 169-172

[2] http://www.ciep.fr/

[3] http://www.coe.int/t/dg4/linguistic/Cadre1_fr.asp

[4] Malyarets L. The probability theory and the statistics: study guide / Malyarets L., Lebedeva I., Zheleznyakova E. – Kharkiv (Ukraine): KhNUE, 2010, – pp. 201-216

[5] Malyarets L., Dorokhov M. Fuzzy-set approach to shaping of information components of economic safety of enterprises // Proceedings of int. Conf. "Securitatea informaţională". - Ch. (Moldova): ASEM, 2011, - pp. 54-56

DOROKHOV MYKHAILO
l'Université National d'Economics de Kharkiv
Département des Systèmes d'Information
9A, Prospect Lenina, Kharkiv
UKRAINE
E-mail: michaeldorokhov@gmail.com

HONCHAROVA MARIIA
l'Université Nationale Pédagogique Skovoroda de Kharkiv
Département des Langues Romanes
29, str. Artem, Kharkiv
UKRAINE
E-mail: morganmuse8@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

HDR Studio – A Windows 8 app for creating HDR content

Fălămaş Vasile Marian
Coordinator: Ralf Fabian

Abstract

In an era where the choices we make and places we visit become more often published to the others via certain networks, people increasingly tend to keep these memorabilia of the past. Although the concept of High Dynamic Range is not a new subject to speak of, the novelty of the application consists in the implementation as a Windows Store (Windows 8) App, which provides the user, a clean way to interact with it. Besides the HDR feature that this app offer, there are also basic operations of image processing and various effects to be applied. Conclusions and future work outline the opportunity of approaching the development of an app for creating HDR content

This paper represents a research on creating High Dynamic Range content and merging this technology area with the Windows 8 technology for creating Windows Store applications. It has a greater accent on creating the application for the Windows 8 operating system and shows the main principles of the modern UI and how they should be used. Furthermore, it also details the application functionality based on the WinRT API, but also there are presented general pieces of information about how the HDR content can be acquired, how it can be displayed and as well how it fits in the application itself.

1 Introduction

Let us begin with defining using simple words, what in fact HDR represents. High dynamic range imaging is the field that studies digital images whose associated colour information has full precision.

To cut a long story short: High dynamic range imaging is a method to digitally capture, store, and edit the full luminosity range of a scene. We are talking about all the visible light here, from direct sunlight down to the finest shadow details. Having all that available in one image opens the door for immense opportunities in post-processing. [1] If this still leaves empty holes in understanding HDR, let’s head back to some more basic concepts:

What is dynamic range? Dynamic range (DR) is the highest overall contrast that can be found in an image, often called also as contrast ratio. Mainly, the Dynamic Range depends on two factors: overall range of brightness and the smallest step. The dynamic range of an image can be enhanced if we use smaller steps in brightness. [5]

As a result of the implementation chosen, the piece of software is a Windows 8 (Windows Store) Application, written with C# and XAML. This choice was made because of support of C# classes of image handling (BitmapImage, WriteableBitmap). XAML is used mainly for creating the view, or in other words, for handling the user interface, which has been created according to the Modern UI concept and conventions proposed by Microsoft.

2 Image Processing and HDR

2.1 Image Processing

In the Windows Store application that is discussed in this paper, there are a few image processing algorithms implementation that are applied on the images that the users chooses for selection. These implementations refer to Brightness Adjustment, Grayscale Transform, Negative Effect, Threshold-based Effect, RGB Adjustment, and Hot & Cold Effects, as well as others that will be later treated in the HDR Studio application section in this paper.

Image pixel values in the default WinRT API are stored as a raw array of bytes (BGRA) that is equals four time the composition of the image's width and height. When working with images, in WinRT, the approach is the following: Thinking that the object is read-only, the system can make certain optimizations with it, so the best use in this case will be having to deal with BitmapImage class. If there are manipulations needed to be made on the image, WriteableBitmap is the best choice to start with. The translation from a BitmapImage to a WriteableBitmap is a weary process that requires many steps such as assigning the URI source of the BitmapImage to the source of the WriteableBitmap, which actually requires a stream.

2.2 HDR Acquisition and Processing

Basically, it is possible to generate an HDR image with three methods:

The first, to fully create them synthetically with a 3D software. Nowadays, all 3D render engines are working internally with 32 floating-point values. So, actually, the data exists as long as it is needed and afterwards can be saved on the disk.

The second way would be to acquire the HDR image directly with a high dynamic range camera. Although this can be thought as an easy way, think again. The equipment needed to accomplish such a task has been proved to be not accessible to most of us, the people that need to create a HDR, and so this way is used mostly by professionals.

The third one, would be to shoot a series of images with different exposures and merge them digitally into one HDR image. This last method is about to be discussed in this paper because the application itself, requires multiple images from which, according to the selected algorithm, a HDR content will be displayed and further available for saving. [5]

2.2 HDR Tone Mapping and Display

The HDR content cannot be displayed directly on the devices we mostly possess or use. This is because the CRT or LCD devices we handle have a limited dynamic range that is inadequate to reproduce the full range of light intensities present in the HDR image. This is where the so called, tone mapping shows up. The discrepancy between the wide ranges of illumination that can be captured and the small ranges that can be reproduced by existing displays makes the accurate display of the images of the captured scene difficult. This is the high dynamic range display problem or HDR tone-mapping problem.

Tone mapping is a technique used in image processing and computer graphics to map one set of colours to another in order to approximate the appearance of high dynamic range images in a medium that has a more limited dynamic range.

For a display to exhibit realism, the images should be faithful visual representations of the scenes they depict. This is not a new problem. Artists and photographers have been addressing this problem for a long time. The core problem is that the light intensity level in the environment may be beyond the output level reproduced by the display medium. Also, the contrast experienced in a real environment may greatly exceed the contrast range that can be reproduced by those display devices. Reproducing the visual appearance is the ultimate goal in tone mapping. However, defining and quantifying visual appearance itself is a challenge, and currently an open research problem.

The Fig. 1, it is shown briefly the steps needed in order to obtain HDR content:

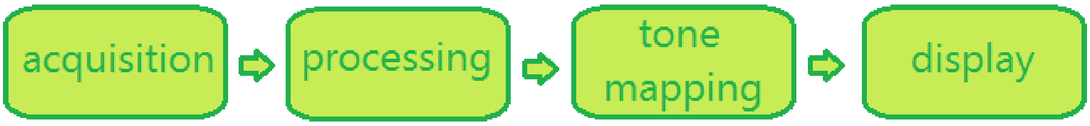


Fig. 1

3 Novelty of Windows 8

3.1 General Facts

Windows 8, is a relatively new technology launched by Microsoft at the late of October, 2012. The main goal of this operating system is to provide easy and simplified use of the possessed device, which now, is not limited to a simple PC or laptop, but also tablets. There are many versions of this operating system such as Professional, Enterprise, Windows 8 Server, or Windows RT (which is a more limited version of Windows 8 that doesn’t allow running desktop applications, used mainly for ARM devices).

It has been redesigned to meet the user’s requirements in terms of simplicity, therefore, the missing Start Button on the screen which has been replaced with a whole new Start Screen that possesses various features.

The new Start Screen is now composed of tiles, actually „live tiles”, representing applications that, depending to their implementation, provides information to the user, directly through its tile (its icon), without having the user to run the app, furthermore enhancing the simplicity and beauty this operating system has to offer.

3.2 Windows Store Apps

Windows 8 introduces a new style of applications, the Windows Store Apps. These are mainly optimized for touchscreen experience and are be more specialized than current desktop applications. A great feature of these applications is that they can be run side by side on the same screen, not referencing here about the concept of multitasking. There are three ways these apps can be run on the screen: Full-Screen Mode, Filled Mode or Snapped Mode. The first, Full-Screen Mode, well, it’s very clear what it will make, the application will occupy all the screen. On the other hand, Snapped Mode goes together with Filled Mode. A Snapped Mode run app will occupy 320 pixel width of the screen, and the remaining will be another app which will run the Filled Mode, like in the example shown in Fig. 2:

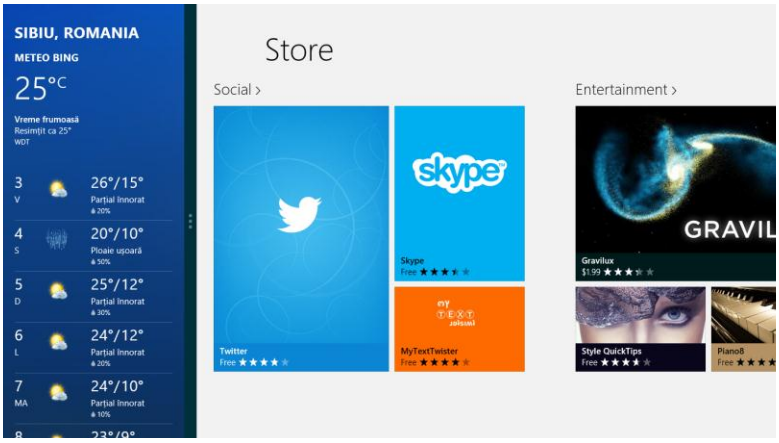


Fig. 2

In Fig. 2, we have a simple example on how apps are run side by side, and shown on the same screen. On the left, we have the Weather App, and on the right the Store App. This is only one of the very many examples that emphasize the utility of Snapped Mode, showing relevant information in a different way without having to switch between multiple open applications. In this case, the Snapped Mode forbids certain features, such as showing the app bar but its main goal is to show data in a compact view, so setting the UI with additional elements that can make it look wearisome, is not recommended.

There are many things that we must take in account, when creating a Windows Store app, mainly conventions proposed by Microsoft, in order to normalize, to make a uniform experience about how the looks of such an app must be, as it is discussed in the next section.

3.2 Modern UI

With the new feature provided by the Windows 8 operating system, the Windows Store apps, developers must obey certain rules in order to ensure that the applications they have written are accepted to be published on the Store. These conventions have been adopted so that the general experience regarding the use of Windows Store applications, remains the same. In other words, developers have to ensure that each page of their application will show pieces of information in a very light way so that the page won't get heavily loaded with visual elements.

Also, what is noticeable, is that the interface has been redesigned to focus more on clean typography, and that is why there's the use of large fonts to emphasize text so it catches the eye. Microsoft sees this design language as "sleek, quick, modern" and a "refresh" from the icon-based interfaces of previous Windows, Android and iOS.

Microsoft designed the design language specifically to consolidate groups of common tasks to speed up usage. It achieves this by excluding heavy loaded graphics and instead, relying on the actual content to function as the main UI. The resulting interfaces favour larger hubs over smaller buttons and often feature laterally scrolling canvases. Page titles are usually large and consequently, also take advantage of lateral scrolling.

Animations play a large part. Microsoft recommends consistent acknowledgement of transitions and user interactions (such as presses or swipes) by some form of natural animation or motion. This aims to give the user the impression of an "alive" and responsive UI with "an added sense of depth".

3.3 Asynchronous API Support

A very useful and important feature of the WinRT API, is the asynchronous support. Methods being run asynchronously are very much recommended if they bodies are treating various operation that can be considered time consuming. The main advantage of using an async (asynchronous) API is to ensure that the application is responsive at all times so that the user will not get frustrated like it was when he received that ”Not Responding” message. Of course, these messages are not present among the Windows 8 applications, because they are not shown (that doesn't mean they can't exist).

Maintaining the code has been also taken into consideration, so from this point of view, everything has been taken care of. This is because using asynchronous API is extremely easy if we take note of just a few things. First of all, the method that uses asynchronous calls needs the async keyword and all the calls in the body, need the await keyword.

The whole process of an asynchronous call of a method is much more complex, but the rest is handled by the compiler and all we need to think about are the correct use of async and await operators.

When developing applications for Windows 8, it is encouraged to have such an approach, to take advantage of the asynchronous API, to ensure the smooth and fluid lifecycle of the app, especially because of the fact that the WinRT offers so many built-in async methods.

4 HDR Studio – merging two technology areas

The application presented in this paper is a Windows Store App created with C# and XAML that has its main objective the creating of HDR content.

The application has its roots as an image processing app which provides basic but useful features for picture manipulations such as brightness or contrast adjustment, negativate, grayscale, RGB values adjustment, Blur effect and Rotations, as well as few more exciting features such as Sepia, Posterize, Color Enhancement, Recolor and Hot & Cold Effects.

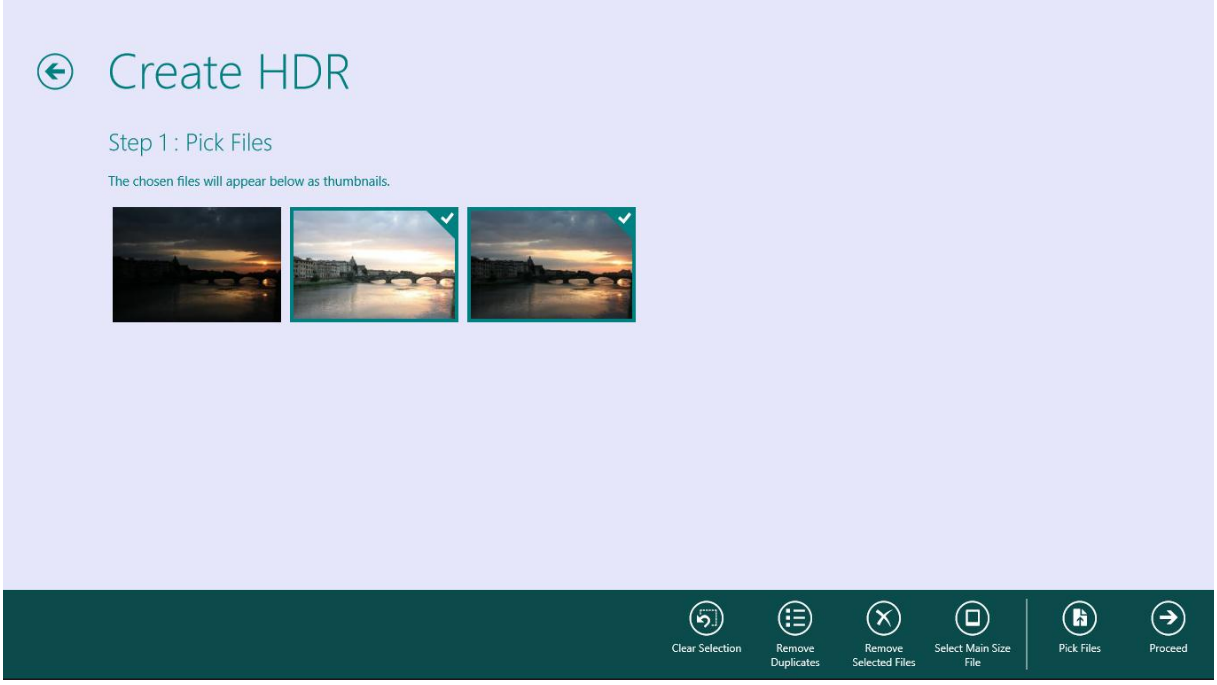
In addition to this characteristics, it's been proposed the idea of implementing certain custom algorithms for creating HDR content. As a result of this idea, the application is split at the moment in three sections: HDR Creation Section, Image Effects and Processing Section and Miscellaneous Section. The subject of this paper is HDR content, so following, we shall speak about the first section, the HDR approach.

As stated earlier, the HDR creation approach adopted, requires multiple images to be selected. As a convention, for the moment at least, all images must have the same width and height and must represent scenes taken at different exposures, so that after the merge, will be shown the result. For the first part, the acquisition, in order to create an UI consistent with the Modern UI rules proposed by Microsoft, it has been taken use of the WinRT APIs to search the computer for the desired files. For this, the following piece of code has been written:

```
var picker = new FileOpenPicker
            {
               ViewMode = PickerViewMode.Thumbnail,
               SuggestedStartLocation = PickerLocationId.PicturesLibrary
            };

picker.FileTypeFilter.Add(".jpg"); picker.FileTypeFilter.Add(".jpeg");
picker.FileTypeFilter.Add(".bmp"); picker.FileTypeFilter.Add(".png");

var files = await picker.PickMultipleFilesAsync();
```

Using the pattern of MVVM [10], on the screen, the images are shown according to the Modern UI design system as thumbnails on which various actions can be applied.

In Fig. 3, we can see the app page of the acquisition phase, and the selected images.



Create HDR

Step 1 : Pick Files

The chosen files will appear below as thumbnails.

Fig. 3

MVVM (Model View View Model) plays a big role here. Through implementing it, it is possible to easily manipulate the collection of images. The MVVM consists of a model class, in this case, called ImageThumbnailModel that handles the image model. The ViewModel class, called ImageViewModel, handles the operations that should be made on the entire collection, such as adding an image when the Pick Files button action is invoked, or deleting selected thumbnails from the View.

A part of the ModelView class, looks like this :

```
public class ImageThumbnailViewModel : BindableBase
   {
      private ObservableCollection<ImageThumbnailModel> _images;
      public ObservableCollection<ImageThumbnailModel> Images {
         get { return _images; }
         set { _images = value; OnPropertyChanged(); }
      }

      public ImageThumbnailViewModel() {
         Images = new ObservableCollection<ImageThumbnailModel>();
      }

      public void AddThumbnail(ImageThumbnailModel newThumbnailImage) {
         if (newThumbnailImage != null)
            Images.Add(newThumbnailImage);
      }

      public void RemoveThumbnail(ImageThumbnailModel toDeleteThumbnailImage) {
         if (toDeleteThumbnailImage != null)
            if (Images.Contains(toDeleteThumbnailImage))
               Images.Remove(toDeleteThumbnailImage);
      }
   }
```

The AddThumbnail() and RemoveThumbnail() methods are the ones that operate on the collection of the images that the user selects. These are the methods called when the user taps the Pick Files button or taps to delete some of the files.

Furthermore, there are some other features available for image selection manipulation such as Clear Selection, Remove Duplicates or Select Main Size File. Clear Selection is a very common feature at mostly all of the Windows Store App and has the simple role of deselecting all the images. Possibly, to be thought a very easy task, but it’s the simple things at hand that make an app great and doing that manually would rather annoy the user. The Remove Duplicates is a great way to ensure that you want to have unique files selected. It mainly searches all the selected images, and discards those that were selected multiple times, doing this by searching image content. Select Main Size File is rather a limitation (for the moment) for creating the HDR Content, and it refers that all selected files must have the same dimensions. When that feature is selected, it is required that an image to be selected, and all the rest that don’t have the same dimensions will be afterwards discarded.

After this acquisition stage, the next step would be creating the HDR. Currently, there are two methods implemented:

The first one, will take into consideration the images as equal. In other words, when building the resulting image, the pixel at (x,y) location will count as much as the other pixels in the other images at the very same (x,y) location.

The second algorithm, is a more complex one. Instead of simply making an assign of an equal contribution at the final HDR content, a more complex tone mapper is applied to ensure context sensitivity. That means, that the lighter areas in an image or the very dark (black) areas, will have a less contribution to the final image. This was made because very light or very black areas are not representative as having information, so it would be rather useful to take into account the areas in the same location from pixels that do have useful information (pixels that are not close to white or close to black).

In Fig. 4, there are present the sample pictures and the result of the implemented algorithms (Equal contribution the left large image and Context Sensitive the right image):

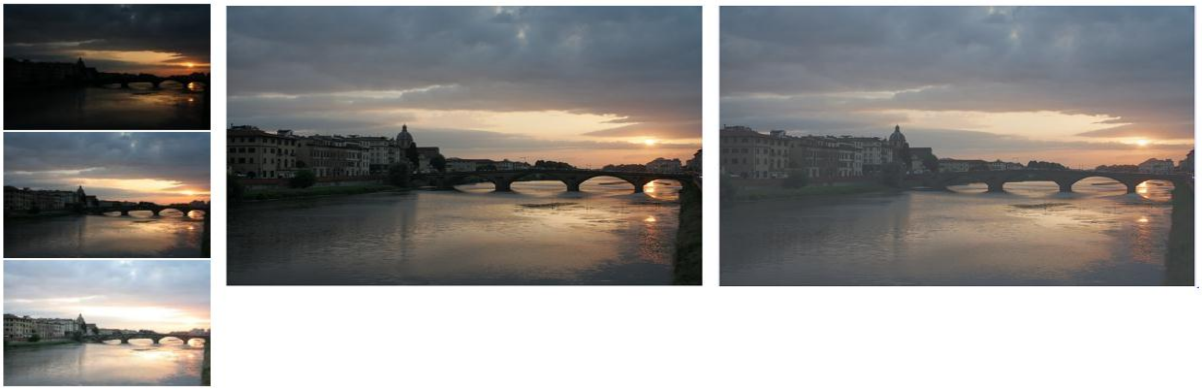


Fig. 4

4 Conclusions and Future Work

HDR Studio is a Windows Store App that specializes in image handling and processing and executes basic manipulations on the images, and as well creates HDR content from given images provided by the user. The UI is fluid and consistent with the Modern UI principles supported by Microsoft although the application subject itself requires extensive customization.

In the future, there will be various other features added to this application. Some of them will be image alignment to allow creating HDR content based on source images that are not perfectly aligned and as well the implementation of more tone mapping operators. As for the integration of the application within the Windows 8 spirit, a Live Tile will be added with the HDR pictures created by the user so far, and asynchronous progress reporting to show operation progress.

References

[1] M. Sa Asla, Carvalho Paulo Cezar, Velho Luiz, *High Dynamic Range Image Reconstruction*, Morgan & Claypool, 2006.

[2] Reinhard Erik, Ward Greg, Pattanaik Sumanta, Debevec Paul, *High Dynamic Range Imaging – Acquisition, Display and Image Based Lightning*, Morgan Kauffman, 2005.

[3] Reinhard Erik, Ward Greg, Pattanaik Sumanta, Debevec Paul, Heidrich Wolfgang, Myszkowsky Karol, *High Dynamic Range Imaging – Acquisition, Display and Image Based Lightning, Second Edition*, The Morgan Kauffman, 2010.

[4] Concepcion Rafael, *The HDR Book*, PeachPit, 2011.

[5] Bloch Christian, *The HDRI Handbook*, Rockynook, 2007.

[6] Lukac Ratislav, *Computational Photography, Methods and Applications*, CRC Press, 2011.

[7] Debevec Paul, Malik Jitendra, *Recovering High Dynamic Range Radiance Maps from Photographs*, University of California at Berkeley.

[8] Hoefflinger Bernd, *High Dynamic Range (HDR) Vision*, Springer, 2007.

[9] Petzold Charles, *Programming Windows 6th Edition*, Microsoft Press, 2012.

[10] MVVM Pattern - MSDN, http://msdn.microsoft.com/en-us/magazine/dd419663.aspx (retrieved 22nd March 2013)

FĂLĂMAȘ Vasile Marian
"Lucian Blaga" University of Sibiu
Department of Mathematics and Computer Science
No. 5-7, Dr. Ion Ratiu St., 550012 Sibiu
ROMANIA
E-mail: falamas.vasile@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Android Device Manager (ADM) review

Simeon Iliev
Teacher Coordinator: Galina Atanasova

Abstract

The subject of this project is the development of Android Device Manager application. The idea behind this application is to make development and management for Android applications easier and faster.

1 Introduction

Android makes a big difference on the mobile market – Android is fast and it uses JAVA. Nowadays JAVA is familiar to many programmers, and most importantly it's for FREE. The source code is available for download from everyone. Over the past 5 years Android has increased its popularity and now it is the worlds leading mobile platform. This has led to many innovations in the mobile industry around the globe.

The idea behind the described project is simple – to make development and testing easier. When you are developing for Android you will most likely use the QtAdb application to manage your devise. It has a lot of useful features and just some of them are a shell, a file browse, etc. This application is the one that comes closest to this project. They both share a lot of the same features. QtAdb is a free and open source project which was started in 2009 and its latest version was published in September 2011. The Android Device Manager (ADM) will be usable on the Windows operating system. The paper is organised as followed: 1 Introduction, 2 Android Device Manager Review, 3 Features, 4 Conclusions.

2 Android Device Manager Review

To use this application you will need to have set up the Android SDK. The *Android Software Development Kit* (SDK) contains the necessary tools to create, compile and package Android application. Most of these tools are command line based. The Android SDK also provides an Android device emulator, so that Android applications can be tested without a real Android phone. An user can create *Android virtual devices* (AVD) via the Android SDK, which run in this emulator. The Android SDK contains the *Android debug bridge* (adb) tool which allows to connect to an virtual or real Android device [3].

Most of the developers have already set up there enviroments and are familiar with the Android development tools provided by Google (ADT).

ADT is a set of components (plug-ins) which extend the Eclipse IDE with Android development capabilities.ADT contains all required functionalities to create, compile, debug and deploy Android applications from the Eclipse IDE. ADT also allows to create and start AVDs. The Android Development Tools (ADT) provides specialized editors for resources files, e.g. layout files. These editors allow to switch between the XML representation of the file and a richer user interface via tabs on the bottom of the editor.

Many developers and users have the need of applications who have more possibilities to operate with the Android OS. Because there is a lack of such applications is the reason behind ADM project and realise.

Nowadays the UML is accepted as a standard language for software project description due to the platform-, technology-, and program-independence of this graphical language [1]. The common use case diagram presenting the basic and the advanced features of the ADM are given on the following (Fig.1).

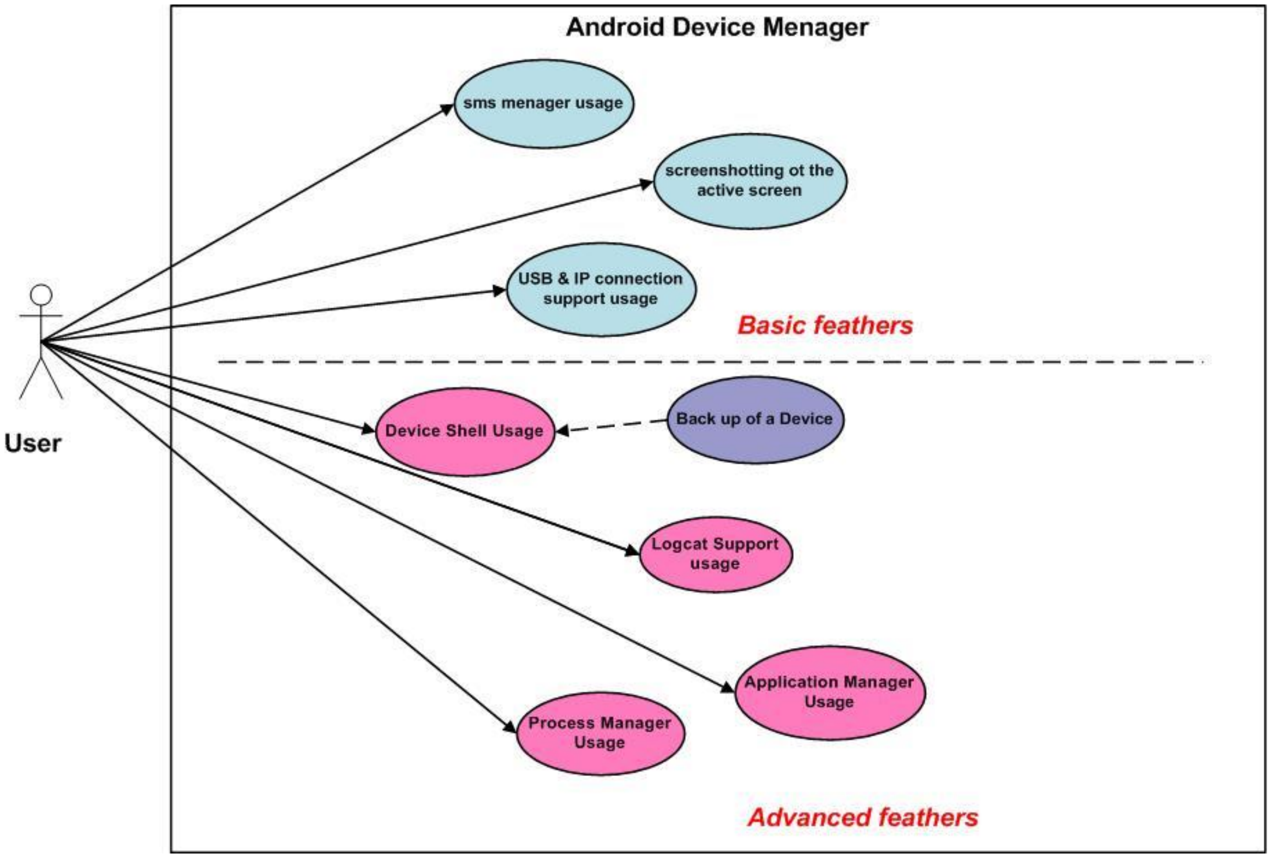


Fig. 1: Common ADM use case

3 Features

3.1 Basic Features

These are the features that the common user is most likely to use. There are a lot of application who have these basic features, but nothing more, and some of them require you to download an app from the Google Play Services to use them [2].

3.1.1 SMS Manager

The SMS manager enables you to view or send SMS on your connected devices in case there are mobile). It also allows to store or to delete them. SMS Manager is a simple yet very functional app to automatically manage SMS and organize them in a cool fashion. The user can managing his friends, family or customers in way that has never been so easy. User forms his own group and selects only people that he/she cares.This feature allows the user to transform the management of sms's into a more pleasant experience.

3.1.2 Support for USB and IP connections

One of the main features is that the Android Device Manager (ADM) application will be able to support multiple devices connected at once via USB or IP connections through two modes: USB accessory and USB host. In USB accessory mode, the external USB hardware act as the USB hosts. Examples of accessories might include robotics controllers; docking stations; diagnostic and musical equipment; kiosks; card readers; and much more. This gives Android-powered devices that do not have host capabilities the ability to interact with USB hardware. Android USB accessories must be designed to work with Android-powered devices and must adhere to the Android accessory communication protocol. In USB host mode, the Android-powered device acts as the host. Examples of devices include digital cameras, keyboards, mice, and game controllers. USB devices that are designed for a wide range of applications and environments can still interact with Android applications that can correctly communicate with the device. This of course can be quite useful for the users.

3.1.3 File manager

The file manager has a lot of features itself. When you use it you will be able to move files between the computer and the different devices. Another important option is the support for tabs and the containing of the different device folders. Creating and deleting files is a plus as well. There is a possibility of editing a file directly on the device. The access writes to the files on the device can be changed. It allows browsing files and folders stored on device and sdcard. It can sort by name, size and date. Copy/Paste, rename, delete, create folder, send file as email attachment features are available. It is able to uncompress ZIP, GZIP and TAR files too. Encrypted ZIP are supported. It can be called from another application through Intent facilities. No permission other than storage/write one is needed. AndExplorerPro provides search and ZIP archive support.

3.1.4 Screenshot of the active screen

There is even a possibility of taking a screenshots of the currently active screen of the device.

3.2 Advanced Features

3.2.1 Logcat

Logcat represents the Android devices build in logging system. With this application you will be able to see the logs of the connected devises to your computer. The logs allow you to monitor system messages. The logging system provides a mechanism for collecting and viewing system debug output. Logs from various applications and portions of the system are collected in a series of circular buffers, which then can be viewed and filtered by the logcat command. You can use logcat from an ADM shell to view the log messages. The Android logging system provides a mechanism for collecting and viewing system debug output. Logcat dumps a log of system messages, which include things such as stack traces when the emulator throws an error and messages that you have written from your application by using the Log class. User can run LogCat through ADM, which allows you to read the messages in real time.

3.2.2 Device shell

The device Shell is basically a Linux terminal in which you can input commands. The Android shell uses more specific commands which is not the same with the standard Linux terminal.

3.2.3 Backup of a device

Creates a backup of a selected device and allows you to specify the location to store it.

3.2.4 Process manager

The process manager will allows you to monitor the processes of a specific device and also to manage them.

3.2.5 Application manager

The application manager will allow you to install or delete an application on a device. This could be very useful in cases when you have the .apk files of the applications.

4 Conclusions

As you have seen I have only specified the features of the Android Device Manager (ADM) application. To summarize the main information, I will tell you what my opinion is. I think that this application will be very valuable to everyone who is programing for Android, but also it could be very useful for just the ordinary users. Everyone will find the application easy to work with. As I mention in the introduction one of the solutions that comes closes to this one is the QtAdb application. It is free and open source for both Windows and Linux operating systems. QtAdb and ADM share a lot of the same features but with its multiple device support and future development I think that the Android Device Manager (ADM) will be one of the next most useful tools for developing with Android. Of course there are a few ideas for the future development of this application. Nowadays the technologies need improvements everyday. My idea is to add contribution to the full development of the application and its future development. One of the things I intend to work on will be a port to Linux operating system. This will allow even more users to use the application. It will be easier and more practical. Another thing that I want to include in this application is to add and improve the ability to record a video from the device. For now this are the things I want to develop on this application.

References

[1] Fowler, M., *UML Distilled: A brief guide to the standard object modeling language*, 3rd edition, Addison Wesley, 2004.
[2] https://play.google.com/store.
[3] http://www.vogella.com /articles/Android/article.html#androiddevelopment.

Simeon ILIEV
University of Ruse “Angel Kinchev”
Informatics and information Technology
Ruse ,Str. Strudentska 8
Bulgaria
E-mail: simeon.iliev.bg@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

PHP Framework for Web-based Applications

Paul Merlas
Teacher Coordinator: Camelia-M. Pintea and Mara Hajdu

Abstract

The current paper illustrates a new user-friendly framework to create websites. *Fever* is the name of the newly introduced framework. The specific components and the way to use it are explained. Significant images are showing the framework interface. Any website developer could easily learn to use the Fever application.

1 Introduction

Today it is important to create dynamic and user-friendly systems for developers. In computer programming, a software framework is „an abstraction in which software providing generic functionality can be selectively changed by additional user written code, thus providing application specific software."[5]

Fever Project Framework is introduced, described and illustrated in the following sections. In Section2 are shown the definitions of PHP and JavaScript. Section 3 deals with our framework: descriptions of the Back-end and Front-end headers, illustrations of menu buttons, variable settings and instructions of using the client-platform. The paper concludes with further possible improvements of the project.

2 PHP and JavaScript

As we know, PHP is free software released under the PHP License. The following definitions are based on [5].

Definition 1 *PHP is a server-side scripting language designed for web-development but also used as a general-purpose programming language.*

PHP code is interpreted by a web server with a PHP processor module, generating a web page. PHP commands are embedded into a HTML document. PHP can be deployed on web servers and also as a standalone shell on any operating system and platform. JavaScript is defined as follows.

Definition 2 *JavaScript (JS) is an interpreted computer programming language.*

JavaScript- formalized in the ECMAScript language standard - was implemented as part of web browsers. The client-side scripts could: interact with the user, control the browser, communicate and alter the displayed document content. It is primarily used as part of a web browser: client-side JavaScript, enabling programmatic access to computational objects within a host environment.**[5]**

Definition 3 *MySQL is an open source relational database management system running as a server providing multi-user access to a number of databases*.

3 Fever Project

The current *Fever Project* was created almost from scratch. There are several classes and methods modified and used from OpenCart [2]. The personalized classes are Action.php, Controller.php, ExportXLS.php, Image.php, Loader.php, Model.php, Response.php, Request.php, Template.php, Url.php, Timer.php and Registry.php.

The folder content of the framework is friendly-user and easy to use and administrate. The framework supports from PHP version **5.3.8** to the latest one. It is made out of OOP (Object Oriented Programming) and has an already javascripts built in functions as JQuery, JQuery Tools, Colorbox, TinyMCE and other javascripts codes.

How *Fever Project works* ? There are many ways to connect to different interfaces of the project. The project includes all the useful and tools and objects needed for a specific template. The menu is dynamic and is based on a controller class. It works very fast on localhost server, you can see a web page in under 0.1 second.

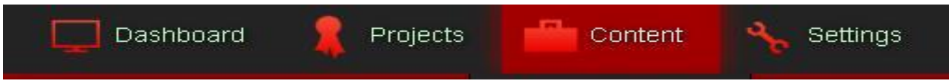


Figure 1 Back-End Header

The framework has a *Back-end (Figure 1)* - what the administrator can do and a *Front-end (Figure 2)* - what the user can see.

Why is so special? Several Fever Framework Project advantages and original features are following.

- after you install the software, it runs smoothly;
- it has a multi-framework system for web-sites with several distinct main sections;
- the templates are very easy to change, without any PHP knowledge;
- after you build a stable version of the site, you can multiply your work in different templates and the whole page will look like a new made framework;
- it is protected by URL and SQL injection.

Why PHP and MySql? They are free and open source applications. The *Fever Framework* uses TAGS to replace a string in the HTML template with the processed Php code and outputs it exactly where you need it and a Model View Controller (MVC) for the different pages.

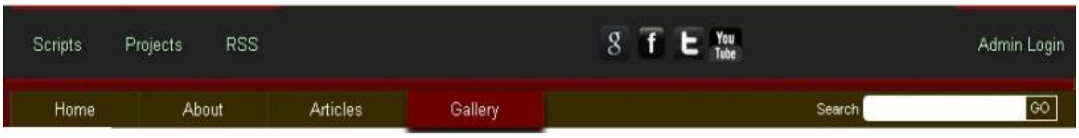
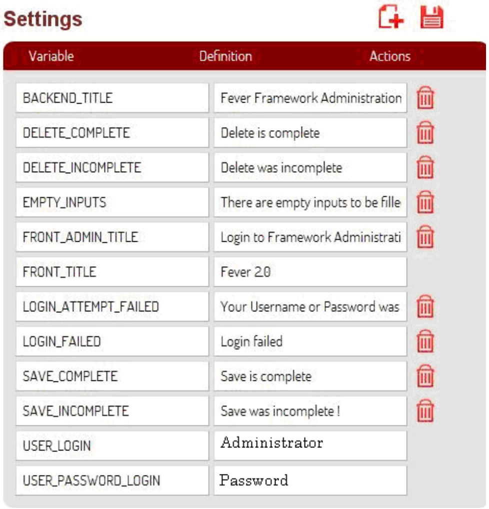


Figure 2 Front-End Header

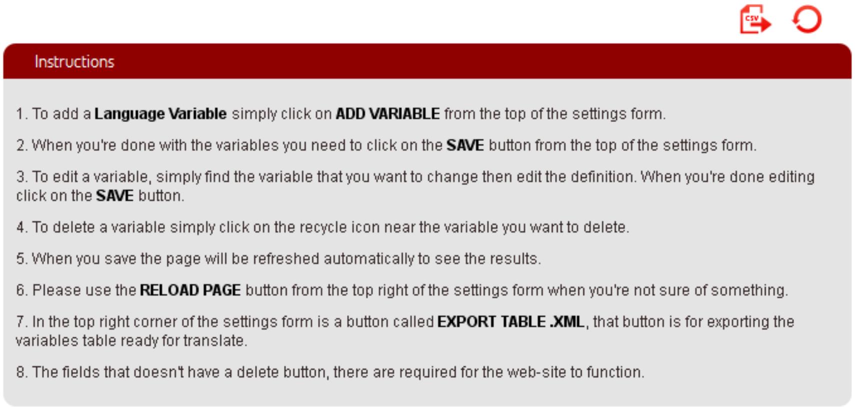


Settings

Variable	Definition	Actions
BACKEND_TITLE	Fever Framework Administration	🗑
DELETE_COMPLETE	Delete is complete	🗑
DELETE_INCOMPLETE	Delete was incomplete	🗑
EMPTY_INPUTS	There are empty inputs to be fille	🗑
FRONT_ADMIN_TITLE	Login to Framework Administrati	🗑
FRONT_TITLE	Fever 2.0	
LOGIN_ATTEMPT_FAILED	Your Username or Password was	🗑
LOGIN_FAILED	Login failed	🗑
SAVE_COMPLETE	Save is complete	🗑
SAVE_INCOMPLETE	Save was incomplete !	🗑
USER_LOGIN	Administrator	
USER_PASSWORD_LOGIN	Password	

Figure 1 Administrator Back-end: Settings

Figure 3 shows the list of variables used for the whole project. For example the variable *BACKEND_TITLE* is used for the title of the Back-end. The name of this variable is *Fever Framework Administration.* This variable, like all the others, except some core variables, can be deleted, updated and created. Some instructions for the user-client are detailed in Figure 4.



Instructions

1. To add a **Language Variable** simply click on **ADD VARIABLE** from the top of the settings form.

2. When you're done with the variables you need to click on the **SAVE** button from the top of the settings form.

3. To edit a variable, simply find the variable that you want to change then edit the definition. When you're done editing click on the **SAVE** button.

4. To delete a variable simply click on the recycle icon near the variable you want to delete.

5. When you save the page will be refreshed automatically to see the results.

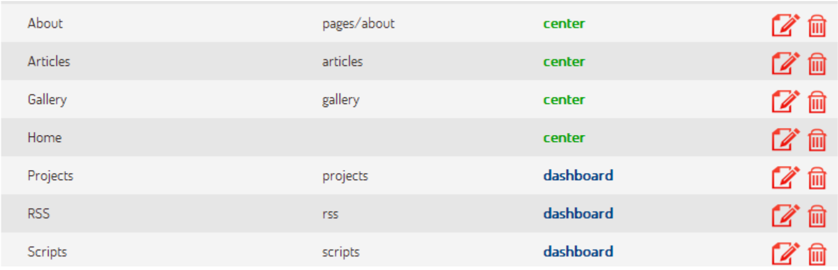
6. Please use the **RELOAD PAGE** button from the top right of the settings form when you're not sure of something.

7. In the top right corner of the settings form is a button called **EXPORT TABLE .XML**, that button is for exporting the variables table ready for translate.

8. The fields that doesn't have a delete button, there are required for the web-site to function.

Figure 4. Instructions for the Client.

Figure 5 shows the menu buttons from the Front-end user. They have a title, a link and what section they belong to. The buttons can be interactively created, modified and deleted.



About	pages/about	center	✎ 🗑
Articles	articles	center	✎ 🗑
Gallery	gallery	center	✎ 🗑
Home		center	✎ 🗑
Projects	projects	dashboard	✎ 🗑
RSS	rss	dashboard	✎ 🗑
Scripts	scripts	dashboard	✎ 🗑

Figure 5.Menu buttons.

4 Conclusions and Further Work

The paper describes a new interactiv project, called *Fever Framework*. The project is based on PHP, MySql and JavaScript. The OOP- based framework has many facilities: creates web-sites based on existing template and also other applications for computers and mobiles.

In the near future, the project will be improved with a new DashBoard. The DashBoard will include some statistics of the website: how many visits does the site has, how long and when does the site loads and when. It will be also implemented a *log* system for errors.

References

[1] Lorna Jane Mitchell, *PHP Web Services*, O'Reilly, 2013

[2] *Opencart*, http://www.opencart.com/

[3] Stoyan Stefanov, *JavaScript for PHP Developers*, O'Reilly, 2012

[4] Alex Libby, *jQuery Tools UI Library, Packt Publishing, 2012*, O'Reilly, 2012

[5] Wikipedia

MERLAS Paul
Technical University Cluj-Napoca
North University Center
Str.Victoriei, 76
ROMANIA
E-mail: crustamet@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Web-based EPC to BPMN converter

Kaloyan Mironov
Teacher Coordinator: Katalina Grigorova

Abstract

The main purpose of business modelling is to present the business logic as abstraction. This is done by diagrams called business processes. The most used business-modelling annotations are EPC and BPMN. Surprising is the lack of an automated tool to convert EPC to BPMN business processes. With this paper we will explain a potential decision of this problem and will present our software solution.

1 Introduction

The In recent years, business modelling has become a main part of each business organization. It helps to analyse and optimize business tasks and it provides a business evolution framework. One of the most common standards is EPC[1], the reason is that it's excellent integration with ARIS toolset. Over the time EPC was replaced by BPMN[2]. Unlike EPC, BPMN is more detailed and standardized. For the rapid growth of BPMN helps the fact that the standard is supported by many different tools. There are too many EPC diagrams and there is no tool to convert in BPMN. This paper will explore the both business-modelling standards and we will suggest different decisions to convert EPC to BPMN.

2 EPC elements

EPC (Event-driven process chain) is type of flowchart used for business process modelling. EPC is defined in year 1992 from Keller. The main purpose of EPC is to define clear way of business process modelling. Figure 1 presents all the core components of EPC.

2.1 Function

Functions are active elements in EPC. The functions represent business task within the company and describe transformation from an initial state to resulting state.

2.2 Event

In EPC events are passive elements. Events occur before or after a function is executed or both. Functions are linked by events.

2.3 Connectors

In EPC connectors describe how events and functions are connected. With the help of connectors it is possible to split the control flow. There are three types of connectors: xor-connector, and-connector and or-connector.

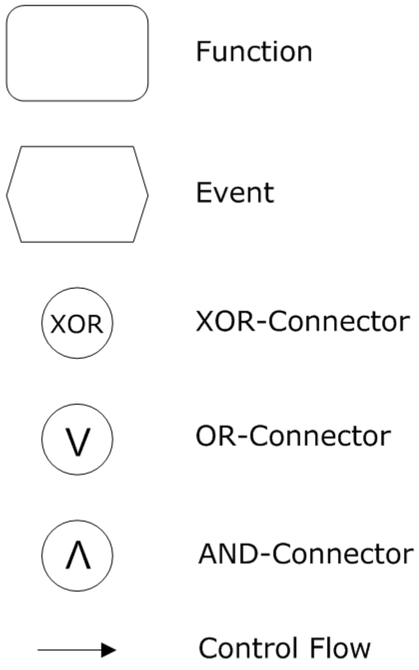


Figure 1 – Core EPC Elements

3 BPMN elements

BPMN (Business Process Model and Notation) is a graphical representation of business process in business process modelling. The objective of BPMN is to support business process management to business and technical users. BPMN is defined as "readily understandable by all business users."[3] from the OMG. BPMN is based on flowcharting technique very similar to activity diagrams in UML. Figure 2 presents all the core elements of BPMN.

3.1 Tasks

The task represents a single unit of work that cannot be broken down to a further level of business process detail.

3.2 Start event

Each business process has some form of initiating event. In BPMN terminology, it is called a start event. There are six different types of start events that we can include in a BPMN model.

3.3 End event

The end event represents the result of a process.

3.4 Gateway

Gateway semantics depend on events and conditions on incoming and outgoing flows. There are two main types of gateway - data-based and event-based.

3.5 Sequence flow

Sequence flow shows in which order the activities are performed.

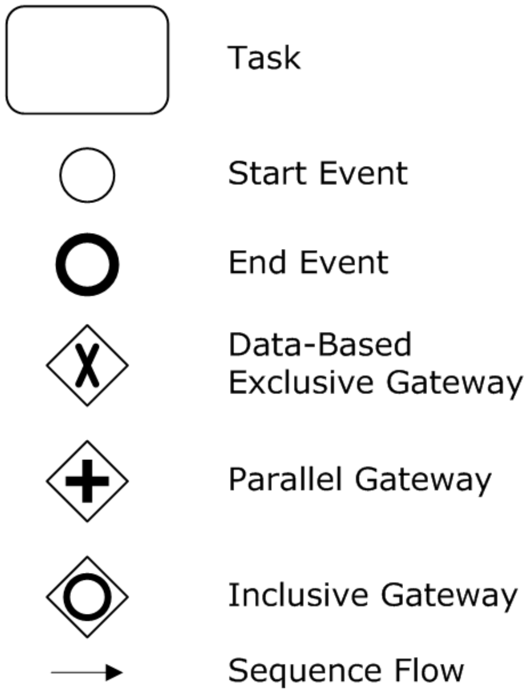


Figure 2 –BPMN elements used for core EPC

4 Converting from EPC to BPMN

The converting is specified as a direct mapping from EPC core elements to BPMN elements. It is realised by structural mapping. The mapped elements are presented on figure 3.

4.1 Function

The function in EPC is defines as an activity supported completion of business tasks. The function is process rule that transforms the input system state to output system state. In BPMN the activity is main term of business work in the business organisation. A task is specified as an activity that cannot be broken down to different activities. The both definitions map the function in EPC to a task in BPMN.

4.2 Events

It is not easy to map events from EPC to events in BPMN. The reason is that they are differently specified. The EPC events can trigger functions, events can be triggered by functions and events define business situation and conditions.

These definitions identify events as state in the system which can stop the following process flow. It is defined as a passive component in the system. EPC and BPMN processes don't use time as parameter and the both can affect the business process flow. Unlike EPC, the BPMN events are defined as something that happens during the business process flow. The difference between EPC and BPMN events is, an BPMN event needs message or signal to continue. The BPMN event is defined more as trigger. Therefore it is not possible to map directly events from EPC to BPMN. To make the conversion possible, we will divide events in two groups - without incoming control flow and without outgoing control flow.

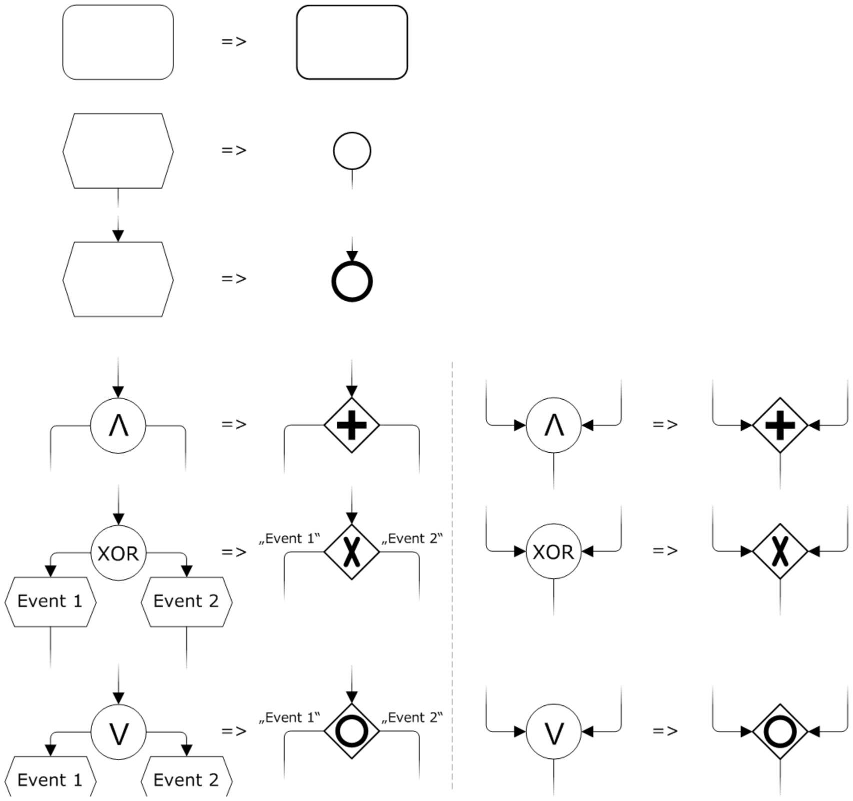


Figure 3 –EPC to BPMN

4.2.1 Event without incoming control flow

The process model in EPC can be determined by one or more events. The BPMN process model is similar. It is also implicitly or explicitly started by start event. Therefore, an EPC event without incoming control flow can be mapped to a start event in BPMN.

4.2.2 Event without outgoing control flow

The process model in EPC can be terminated by events. The BPMN a process model is terminated by an end event. Therefore, an EPC event without outgoing control flow can be mapped by end event in BPMN.

4.3 Connectors

It is easy to convert EPC connectors to BPMN objects. The EPC connectors (xor, and, or) are defined as concatenation for events and functions. It is similar in BPMN. In BPMN gateways are defined to controlling the divergence and convergence of the sequence flow. Therefore, EPC connectors - "xor", "and", "or" can be mapped to "data-based exclusive gateway", "parallel gateway", "inclusive gateway".

5 The “EPC to BPMN” converter

The converter is web-based application. To develop the system is used: jsf, xhtml, xml and primefaces. The application is hosted on JBoss Application Server 7.1. The converter uses MySQL database, hosted on SQL server. The converter needs xml file with EPC business process as input data. The output will be second xml file, contains converted BPMN business process.

6 Conclusion

The developed application is potential solution for converting EPC business processes to BPMN business processes. Nowadays the converter can translate only the core EPC elements. The development of the application continues and we plan to build fully-functional application, converting all EPC elements to BPMN standard.

References

[1] Stein S. *BPMN EPC semantic comparison*. Aris Community. Apr 2010.

[2] *Business Process Modeling Notation*, V1.1. Technical report, Object Management Group (OMG), January 2008. http://www.omg.org/spec/BPMN/1.1/PDF.

KALOYAN MIRONOV
Ruse University
Department of Informatics and
Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: mail@kaloian.net

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Droid Monitor – From code to market

Obancea Dragoș
Teacher coordinator: Lect. Dr. Lucian Sasu

Abstract

Life cycle of a software system is based on the requirement specification and design, then it goes through the development and testing stages before it reaches the final customer, when it begins the lengthy phase of maintenance and promotion. Unlike other areas, IT products continue their development even after their publication, thus generating repetitive cycles which depend on the type and objectives of the systems. The paper tackles on activities management, promotion and maintenance of a product which has undergone stages of specification, development and testing, further on highlighting both positive consequences as well as problematic ones induced by different actions.

1 Introduction

Ever since the first release of software products on market, the process of development was divided into two stages:

- Specification, development and testing phases that can be viewed as a series of nodes linked by bidirectional arcs. This stage generally consists of a budget and a fixed deadline due to the release of the final product, beta version successor.
- The management and promotion of the product which gradually evolves. This phase usually contains the development and testing stages to, pointing out the difference between the success and failure of a product.

This paper is entirely devoted to the second stage, outlining the phases of specification, development and testing that have been completed. The created system is called *Droid Monitor* and represents a three software system product that automatically monitors activity of mobile devices (GPS location, calls, messages, so on and so forth) as in following descriptions:

- An Android application that automatically monitors events on mobile devices and sends information to the server.
- A server application that receives and saves data from the Android.
- A web application where customers can view the data in different formats.

Shortly after the completion of its development in beta form, necessary procedures of administration and promotion were started. *Droid Monitor* has encountered both benefic and regressive periods, since the past, when the required authorizations were obtained and the product was released, until today, when registered users 1500 units. The causes and effects of relevant events related to the developing process of the product will be presented in following sections.

2 Authorizations

" The protection of personal data represents a new field for Romania's legislative space. Its' essence regards, in a generic form, the natural person's right of protection of those specific features which lead to his/her identification and the state's correlative obligation of adopting adequate measures to ensure an efficient protection.

For this purpose, a central authority empowered with such control competence, the National Authority for the Supervision of Personal Data Processing, came into existence in Romania, too. Recently set up, under the Law no. 102/2005, the Authority exerts the competence established mainly by the Law no. 677/2001, in terms of independence from any public authority or private entity.

Under the Law no. 677/2001 on the person's protection regarding the processing of personal data and the free circulation of these data, the acquis represented by the Directive no. 95/46/EC was implemented, which sets up the general juridical frame of the personal data protection at European Union level.

The competences of the National Authority for the Supervision of Personal Data Processing are specific for any institution of control, including the investigation of personal data processing conducted under the Law no. 677/2001 and the sanctioning, if it comes out that the legal dispositions were infringed by the personal data processors, as a result of self-notification or based on complaints filed by the people who's rights were infringed.

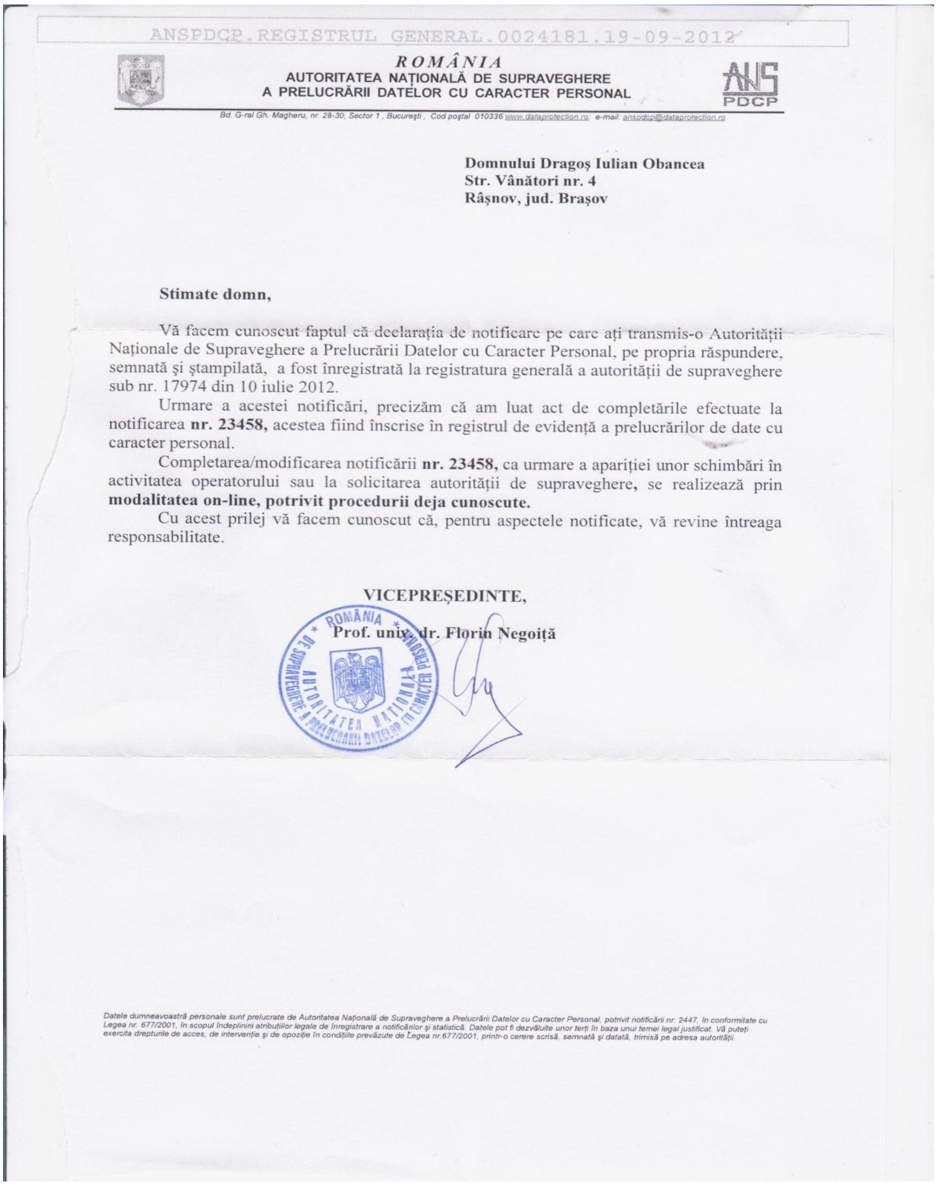
Since the correct enforcement of a new legislative act requires an adequate knowledge of its' provisions, we think that, through the information offered in this site, both personal data processors and natural persons who's personal data are processed will have the opportunity of studying thoroughly and understand which are the obligations and, respectively, the rights which are settled."[1]

Any type of web application from Romania, which deals with personal data processing, requires written approval from *the National Supervisory Authority for Personal Data Processing*. According to Law no. 677/2001, www.droidmonitor.com subject to these rules for processing the following personal data:

- Email - mandatory element
- GPS location - obtained element
- Messages details and their content - obtained element
- Calls details - obtained element
- Contacts - obtained element
- Name, address, telephone - optional element introduced by the user

The first step required in order to obtain approval is to send a notification to the national authority. The document can be filled in online at www.dataprotection.ro and is divided into two parts: a first part that describes the type of personal data and notification, and a second part, more consistent, which deals with the type of processed data, how to obtain and storage it, plus other security details.

When completing this form, you receive a registration number valid for 30 days, which must be signed and send by mail to the authority. If the agency returns an affirmative response, then you receive by post a signed and stamped approval, which holds a registration number, as shown in Figure 1.



ANSPDCP.REGISTRUL GENERAL.0024181.19-09-2012

ROMÂNIA
AUTORITATEA NAȚIONALĂ DE SUPRAVEGHERE
A PRELUCRĂRII DATELOR CU CARACTER PERSONAL

ANS PDCP

Bd. G-ral Gh. Magheru, nr. 28-30, Sector 1 , București , Cod poștal 010336 www.dataprotection.ro; e-mail: anspdcp@dataprotection.ro

Domnului Dragoș Iulian Obancea
Str. Vânători nr. 4
Râșnov, jud. Brașov

Stimate domn,

Vă facem cunoscut faptul că declarația de notificare pe care ați transmis-o Autorității Naționale de Supraveghere a Prelucrării Datelor cu Caracter Personal, pe propria răspundere, semnată și ștampilată, a fost înregistrată la registratura generală a autorității de supraveghere sub nr. 17974 din 10 iulie 2012.

Urmare a acestei notificări, precizăm că am luat act de completările efectuate la notificarea **nr. 23458,** acestea fiind înscrise în registrul de evidență a prelucrărilor de date cu caracter personal.

Completarea/modificarea notificării **nr. 23458,** ca urmare a apariției unor schimbări în activitatea operatorului sau la solicitarea autorității de supraveghere, se realizează prin **modalitatea on-line, potrivit procedurii deja cunoscute.**

Cu acest prilej vă facem cunoscut că, pentru aspectele notificate, vă revine întreaga responsabilitate.

VICEPREȘEDINTE,
Prof. univ. dr. Florin Negoiță

ROMÂNIA AUTORITATEA NAȚIONALĂ DE SUPRAVEGHERE A PRELUCRĂRII DATELOR CU CARACTER PERSONAL

Datele dumneavoastră personale sunt prelucrate de Autoritatea Națională de Supraveghere a Prelucrării Datelor cu Caracter Personal, potrivit notificării nr. 2447, în conformitate cu Legea nr. 677/2001, în scopul îndeplinirii atribuțiilor legale de înregistrare a notificărilor și statistică. Datele pot fi dezvăluite unor terți în baza unui temei legal justificat. Vă puteți exercita drepturile de acces, de intervenție și de opoziție în condițiile prevăzute de Legea nr.677/2001, printr-o cerere scrisă, semnată și datată, trimisă pe adresa autorității.

Fig. 1 Approval from ANSPDCP

It is worth mentioning that *Droid Monitor* has not received approval from the first attempt, because the notification was completed in a wrong manner: the web application was missing the following text, which was later on posted and addressed to the users:

" When creating an account on www.droidmonitor.com you automatically agree with following terms and conditions:

According to Romanian Law nr. 677/2001 specifications, in respect of person protection of personal data processing and their free travel, which was modified and finalized, and of Romanian Law nr. 506/2004 concerning personal data processing and protecting the privacy in electronic communication sector, www.droidmonitor.com has the obligation to safely administrate it with precise specifications, the personal data that you, a member of your family or any other person furnish us. Collecting data consists in personal usage of data by every user.

Furnishing personal data is not compulsory, unless you agree stocking them on our server.

According to Romanian Law nr. 677/2001, you benefit the right to access and to interfere in the database, the right not to be exposed on taking individual decisions and the right of contacting the authorities. Moreover, you have the right of refusing personal data processing and demanding data *removal*. In order to make use of these rights, we place at your disposal the address that contains an already completed, dated, signed and scanned form on support@obancea.com. Also, the right of contacting the authorities is legalized. If some of your personal data is incorrect, we please ask you to inform us as soon as possible.*

Warning:

- *It is prohibited to install this application without clear consent from the user / owner of the device. DroidMonitor is not responsible for installing, traffic and data visualization on other devices than the ones used by the owner.*
- *Every person has the right to oppose, out of legitimate reasons, personal data processing. This right of opposition can be excluded on regulation terms provided by the law (for ex.: regulations performed by financial and fiscal services, police department, justice department, social security). Consequently, this specification will not be activated, if data processing has a mandatory feature;*
- *Every person has the right to freely oppose, without any justification, personal data processing that may hold direct marketing objectives.*
- *We do not have a refund policy, so before buying a PRO version please make sure that you are satisfied with your free account. "*

After dealing with the problems raised by the agency, the notice was modified and posted online, so that its validity was maintained available ever since.

3 Product publication

One of the main objectives of the project was to turn an idea into a set of applications and finally developing a real product. The application had to be released online, so that it becomes a certified product. Consequently, there existed two different stages of publication, due to the multiple product´s composition:

- Publishing web application: a minimal one year investment of approximately 20 Euro was paid for activating www.droidmonitor.com . After first completing the web application, and second obtaining the required approvals of national legislation, the acquisition was linked to the web application via a personal computer.
- Publishing Android application: adapting *Droid Monitor* to various media was a chief priority, so this is why the Android application was made accessible to a larger public. Initially, the app was only available on the web application, and ended reaching the largest market for Android applications, namely *Google Play.* It was not an easy way to accomplish, requiring a period of approximately 20 days to purchase the proper publishing rights, and to obtain confirming evidence from *Google.*

4 Economic

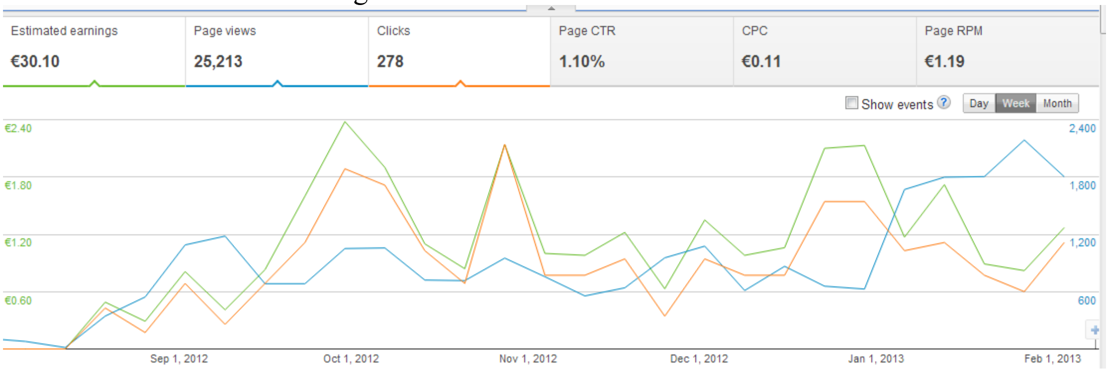
Even if the main objective of the project was developing a long lasting product, on long term, the system cannot last without proper funding. This is why, another important objective was to reach a point when *Droid Monitor* will be able to support some or all of the necessary costs, so as to *"produce more than it consumes* " or in other words to draw profits. [2]

Droid Monitor has provided two manners of obtaining financing support, both addressed to the users:

- Through Android Application: Two different approaches have been made:
  - By adding advertisings: This method remained unreachable, because the app was designed to run in the background, which means that users do not have direct contact with it, and so leading to inefficiency.

- By publishing a PRO Paid version: This attempt failed, after discovering that Google does not offer Romanian developers the possibility of charging money for downloading from *Google Play*. The only way to circumvent such rule was to open an bank account in one of the countries listed by Google, which has been considered useless in the initial stage of the product.

- Through the Web application: Two different approaches have been made:
  - Disposal of Free and Paid user accounts: The difference between the two account types was regarding their time validity, because the available tools were identical, and the 30 days free offer as well, whereas in order to prolong the usage of the product a tax extension is necessary. When publishing web application, such method was implemented, so that payment was safely delivered by using *PayPal* tools. Certain techniques were also added to warn users whose accounts were to expire by sending e-mails and advertising messages, and so contributing to provide a complete and safe module. Unfortunately, after about a month, tests were made on the basis of statistics, and it was decided to eliminate additional accounts, and so providing 100 years validity for each free account. However, payment module remains in place and can be activated at any time, depending on product performance and development on the market.
  - Adding paid advertisements: For this purpose it was decided to use *Google AdSense* tools by creating an account and send a request to verify the fulfillment of the rules by the web application. Having received the approval, advertising space was exclusively offered by *Google* on the sides and bottom section of each page. During this process a set of rules were settled regarding formatting and content advertising, which can change every client browser application. This method was the only one that survived ever since the releasing date and it proved itself useful, as illustrated in figure 2.
  - Remarks:
    - Earnings fluctuations are determined by various factors which will be presented in the following sections.
    - The graphic was generated using tools provided by *Google AdSense* earnings monitoring.



Estimated earnings	Page views	Clicks	Page CTR	CPC	Page RPM
€30.10	25,213	278	1.10%	€0.11	€1.19

□ Show events

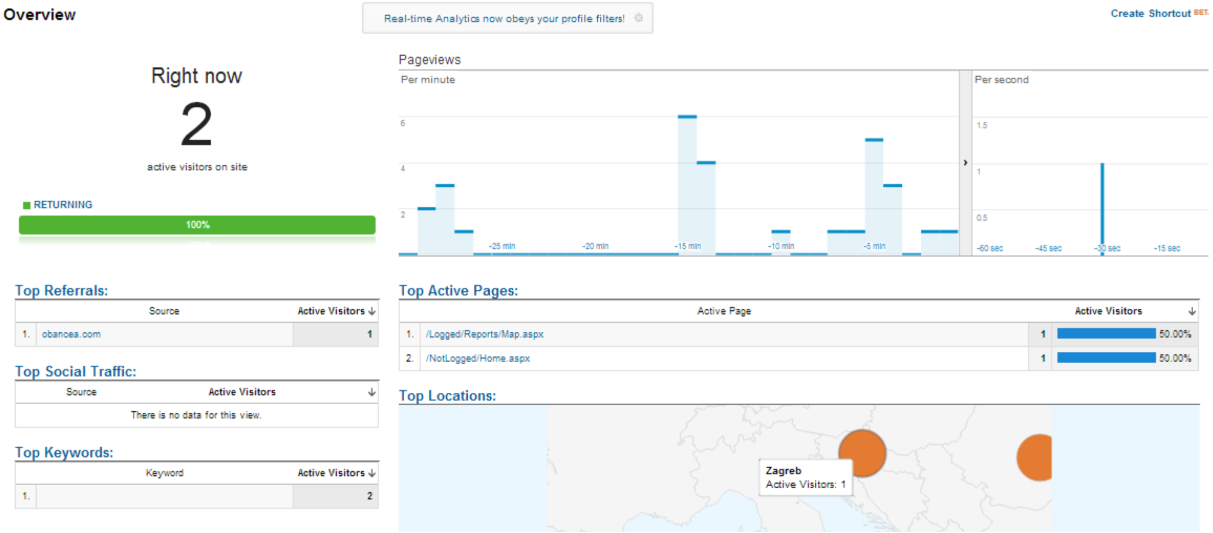
Fig. 2 Example of graphic advertising revenue from web application

5 Monitoring traffic

Monitor the traffic of web application is the most efficient way for analyzing the success of the product and may provide important clues regarding further development and optimization needs. [3]

In addition to the graphics and statistics obtained from *Google AdSense, Droid Monitor* also benefits from monitoring and tracking tools supplied by *Google Analytics.* These tools are disposing a range of statistical information, where the maximum interest rests on:

- *Real Time:* It shows online users and real-time details about their sessions. This information is used at times when the server requires an interruption due to restarting the system, updating or reconfiguring it, and so allows the administrator to know when the web application is accessed by someone, or is the least accessed.



Overview

Real-time Analytics now obeys your profile filters!

Create Shortcut BETA

Right now

2

active visitors on site

RETURNING 100%

Top Referrals:

	Source	Active Visitors ↓
1.	obancea.com	1

Top Social Traffic:

Source	Active Visitors ↓
There is no data for this view.	

Top Keywords:

	Keyword	Active Visitors ↓
1.		2

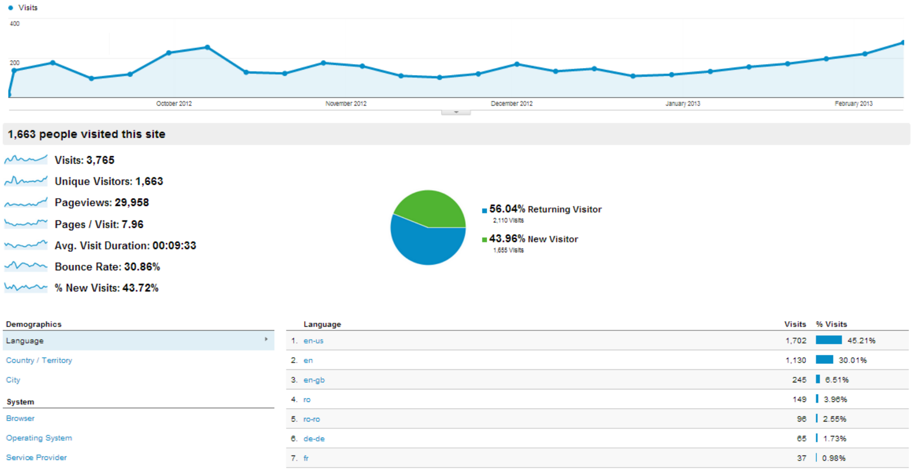
Top Active Pages:

	Active Page	Active Visitors ↓	
1.	/Logged/Reports/Map.aspx	1	50.00%
2.	/NotLogged/Home.aspx	1	50.00%

Top Locations:

Fig. 3 Real time traffic pannel example

- *Audience & Traffic sources:* Provides important information about the number of visits to the web application, the number of page views per session, information about browsers, operating systems, user location, etc.
  Obtaining this information was crucial in order to establish improvements and changes in the web application. The positive effects were numerous, such as:
  - Monitoring the number of visitors during the use of promotional methods to establish their effectiveness.
  - Information on traffic sources led to optimization of the web application for search engines.
  - Information on the type browsers led to optimization the application to access it from mobile devices.



1,663 people visited this site

Visits: 3,765
Unique Visitors: 1,663
Pageviews: 29,958
Pages / Visit: 7.96
Avg. Visit Duration: 00:09:33
Bounce Rate: 30.86%
% New Visits: 43.72%

56.04% Returning Visitor
2,110 Visits

43.96% New Visitor
1,655 Visits

Demographics
Language
Country / Territory
City

System
Browser
Operating System
Service Provider

	Language	Visits	% Visits
1.	en-us	1,702	45.21%
2.	en	1,130	30.01%
3.	en-gb	245	6.51%
4.	ro	149	3.96%
5.	ro-ro	96	2.55%
6.	de-de	65	1.73%
7.	fr	37	0.98%

Fig. 4 Audience overview panel example

- *Visitors flow and Content*: Shows the most frequently traveled routes used by users to access the web application. This information has inspired major changes in the web application in terms of linking the pages and the information they provide.

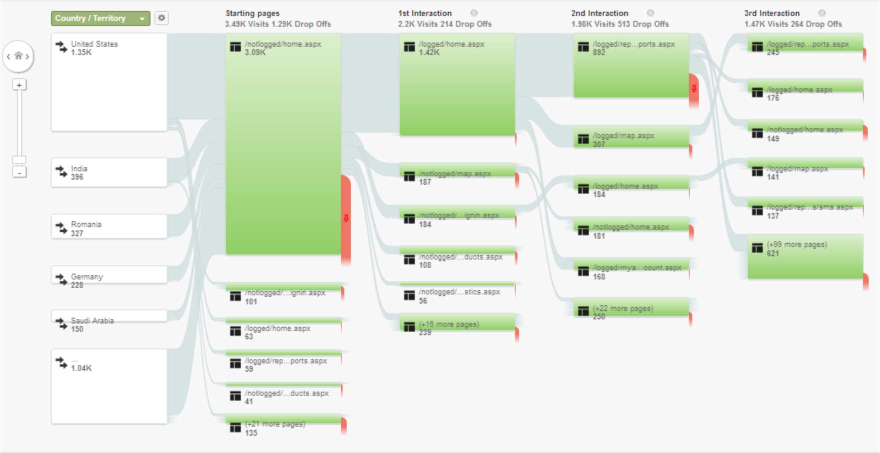


Fig. 5 Visitors flow panel example

6 Support

After the product release, one of the most important aspects that can influence its further development is ensuring a proper contact with the users. Either by direct contact or e-mail messages, the user must be able to address questions, complaints, ideas or simply opinions. To facilitate communication with users and potential clients, the web application provides both a page dedicated to sending messages to the administrators as well as an e-mail address for support: suport@obancea.com .

The easy accessing way to the contact page form, associated with simplified messaging led to a significant number of registrations from users. Message diversity was surprising because they have ranged from questions about product functionality for tracking calls on to stolen phones and small business proposals. Due the insight routine of daily checking the e-mail box, all users receive a reply within 48 hours. The following are representative type of text received on support e-mail address:

" When I go to the report page it says nothing found like there has been no activity. What is the problem? Please help."

" I asking if in this application is true that we can devices or can search my family and friends..because i like yo try this cause i have problem to my husband i want to know where he always going wen he been in the outside i hope this application help me so much..thankyou"

" On your Statistics Page, what is the Max GPS Speed calculation measuring? Thanks"

" HELLO I'm satisfied to use the Droid Monitor because I can follow all the events and receive the report of spotted places of my actual device. On november 1st my fist droid was stolen in a bus in, Madagascar. I declared it but I can't stay without searching and find your application with this android and I think that it's possibble TO LOCATE MY LOST ANDROID, isn'it?"

The diversity of income messages has led to new ideas, including the most desired feature of finding stolen devices, even when there was no previous installing of *Droid Monitor* application. At first glance this idea may seem impossible, but Android allows you to use certain "tricks" for remote installation of applications.

7 Aggressive advertising

The first real attempt of advertising which was addressed to a significant number of people was to automatically send emails. This decision was prompted by the development of another product called *WebCrawler* that automatically accesses different web pages according to predetermined rules and picks contacts, saving them in a database. Another feature of this application was sending emails to the addresses that were initially obtained.

After running the *WebCrawler* for about 10 days, approximately 60,000 e-mail addresses were obtained, out of which about 20,000 belonged to Android application developers. Consequently, the procedure of sending emails was delivered to the 60,000 contacts at a speed of 90 e-mails per minute. The messages were designed to promote *Droid Monitor*, whereas the subject box was filled in with the name of the page where contacts were initially found, in order to induce the recipient a certain curiosity and leave a first impression that the text is not promotional.

Unfortunately, this activity may be considered spam and therefore the server email sending was blocked after about a week. On account of such positive results and growing number of new users, a personal server was set up to resume sending electronic messages at an even higher speed. After about 2 weeks, ISP blocked the ports required for sending these messages, and so ended this kind of promotion.

The consequences of this type of promotion initially seemed positive, as it was indicating a doubling of the number of visits to the web application as shown in Figure 6. However, statistics have proved misleading because, at a closer look, it was observed that in the time aggressive advertising was made , the percentage of accident visits was located at around 60%, which means that 60% of the users were abandoning it after visiting a single page of the web application. Thus, taking into account all these aspects, aggressive advertising actually led to a decrease of about 10% in the number of real users of the product *Droid Monitor*.

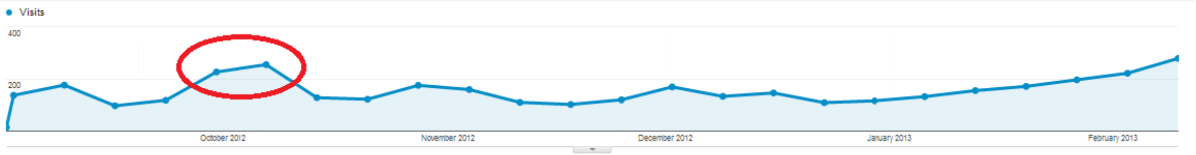


Fig. 6 Number of visits of the web application after aggressive promotions

8 Side effects

In the real world, most products on the market, regardless of their type, handle with periods of regression in terms of number of customers or users. *Droid Monitor* is no exception to this *unwritten* rule, because ever since its publication to the moment of speaking it has undergone two major setback times:

- Prohibition on *Google Play* application: the first *'kick'* received by Android *Droid Monitor* was suspension from *Google Play.* This was signaled by an email received from *Google Play* team, the motivation consists in the fact that the application processes personal data without properly notify the user, as follows:
  " This is a notification that your application, Droid Monitor - Tracking App, with package ID rotvaiv.droid.monitor, has been removed from the Google Play Store. REASON FOR REMOVAL: Violation of the dangerous products provision of the Content Policy…"

- In response to this warning, a clear messages was sent, explaining why the application should not be removed from *Google Play*. Actually, a list with 5 reasons was delivered, that represent 5 stages where the user is explicitly notified give his consent on continuing to install the application. In other words, before using this product, the user is warned 5 times in regard to the retrieved and processed data. Unfortunately, after two weeks of insistence *Google Play* team failed showing any courtesy to respond to the arguments presented, therefore being forced to release the application under a different package. Consequently, these events led to the first regression in terms of number of users, illustrated by the red ellipse on the left of Figure 7.
- The blocking of required ports for the web application: Due to the manifested aggressive advertising by sending e-mails from the PC that also hosted the web server, ISP decided to block several ports including port 80, which was required for proper functioning of web application. Consequently, the web application was unavailable for about 7 days to remedy the situation. Fortunately, the internet provider team comprehended the mistake made by sending messages, and unlocked all ports provided restraining from repeating actions which have caused problems. For this reason, it consumed the second period of regression, illustrated by the second red ellipse in Figure 7.

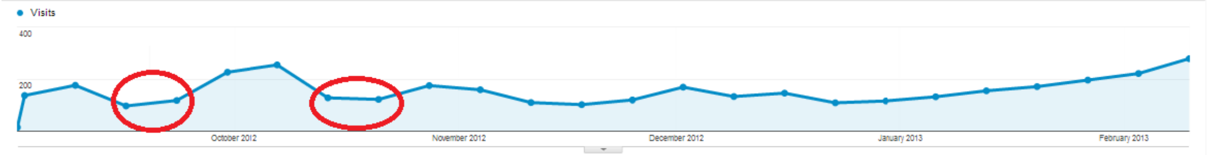


Fig. 7 Number of visits of the web application after side effects

9 Paid advertisements

As a consequence to the negative side effects caused by aggressive advertising, it was decided upon an investment of 50 Euro in promoting the product. After a thoroughly analysis it was determined to promote Android application in detriment of the web application, due to the risk of concomitant access to the web application by a large number of users, which could have led to excess hardware´s capacity, software and data traffic of server.

The value of 50 Euro was the minimum amount required by the largest "pawn" advertising market for Android applications, namely *Google AdMob*. Consequently, this value was divided by 5 days, with a budget of 10 Euro per day, with the price of 1 cent per access resulting in 1000 hits a day, where access is regarding the web application, but on page Android application on *Google Play*. Advertising space was distributed on Android applications that were registered on *Google AdMob*. The effect determined by this advertising enterprise was slightly positive one, but below expectations, as highlighted in the area circled by a red ellipse in Figure 8.

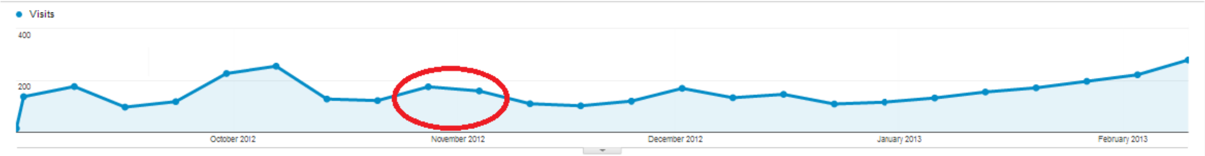


Fig. 8 Number of visits of the web application after paid advertisements

10 QR codes

A modern way in terms of promoting a product is QR codes. In fact, they are images composed of black and white, similar to barcodes that when scanned with a specific application for mobile devices makes the connection to different websites. *Droid Monitor* also provides a QR code which links to a *Google Play* section dedicated to the Android application. In order to promote it, images of QR codes were placed on the pages of social applications and on the personal vehicle, stirring curiosity of some people, and son leading to slightly positive effect on the number of views of the web application.

Fig. 9 Droid Monitor QR Code

11 Search engines

A short time after the web application release, the major search engines would return pages of the *droidmonitor.com* only by entries like *"Droid Monitor"* or *"DroidMonitor"*. Due to these issues plus *Google Analytics* statistics were showing that over 75% accessing came from direct traffic, in other words only the links *Google Play,* meaning that the product could only be found via the Android. Consequently, it was decided to optimize the web application in order to find the product by means of search engines.

After readjusting the messages from different pages of the web application, and metadata fields, *Droid Monitor* can be found in the first search results that contain combinations of keywords: *free, droid, android, monitor, tracking, monitoring, application,* and *app.* As a result, the percentage representing direct access to web application source declined to around 50%, the other half being fulfilled due to search engines.

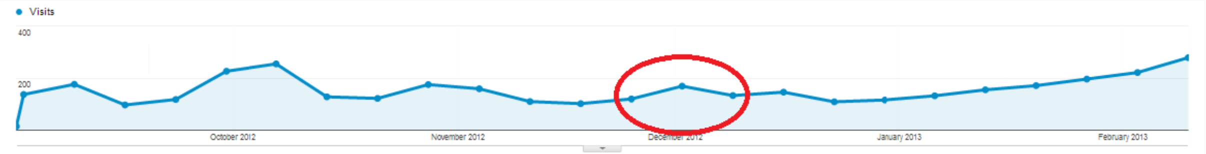


Fig. 10 Number of visits of the web application after search engines optimization

12 Social networks

Another new trend in promoting products, especially computer software, is the publication of accounts or pages on major social networks. In the case of this project two pages of the most popular social networks were created:

- *Facebook:* https://www.facebook.com/DroidMonitor
- *Google Plus:* https://plus.google.com/u/0/106713907860104013681/posts

Configuring social networking pages can be useless if they are not managed and updated. Therefore, it is recommended for increasing the efficiency to add regular updates of events, photos, news or distributions. Furthermore, two secondary menu buttons were added and an appreciation of the web application, one for each social network.

Impact on the number of users was slightly positive in the short term, but it is estimated that it has and will have a more pronounced impact on medium and long term.

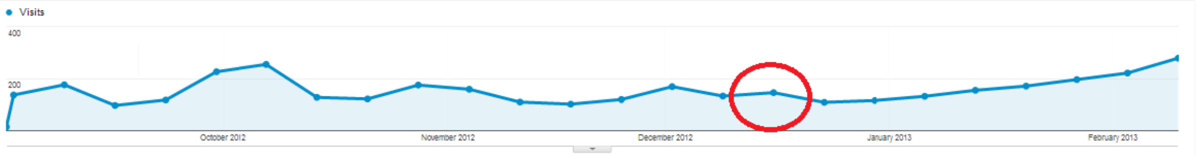


Fig. 11 Number of visits of the web application after social networks sign-ups

13 Presentations

Another way to promote *Droid Monitor* was to present the development of the project at various IT conferences. Unlike promotional methods mentioned in this work, presentations are not intended to attract new users but to attract people interested in any product-related collaboration. The main conferences where the project was presented were:

- International Scientific Session of Sibiu, ICDD, 2012, attainment: special award;
- Scientific Session of the Faculty of Mathematics and Informatics, University of Braşov, 2012, attainment: 1st prize;
- IMSMART Bucharest, 2012, where they spent two presentations, one at the initial stage of the competition and one in the finals, attainment: special award;

14 Advertising tricks

Besides the popular promotion methods presented in this paper, there were some less conventional actions with small effects in the short term but combined, they brought a significant number of new visitors on long term. Among the methods that have been used, the ones with impact were considered as follows:

- Forums: There are many websites that deal by means of discussions issues related to Android applications and more specifically on monitoring and tracking of mobile devices. Consequently, various comments were introduced in various forums, which contained recommendations and links to *Droid Monitor* application.
- Comments on Google Play: Some Android applications with similar features with the current product were installed and analyzed in order to shape new ideas for functionality. Depending on the quality of these applications comments were made on *Google Play* that contained, among other things, links to *Droid Monitor*.
- Application sale on Ebay: *Droid Monitor* was listed on Ebay not for sale purposes, but for promoting it, because the presentation page has the potential to collect a large number of views.

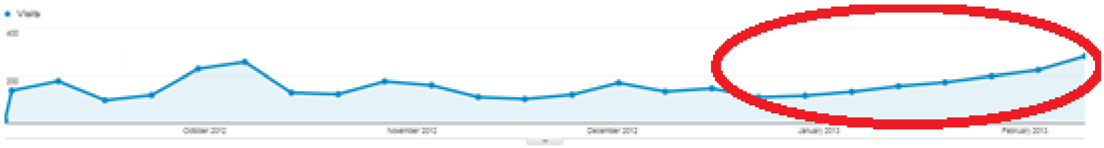


Fig. 12 Number of visits of the web application on long term

15 Regular updates

All the three systems of *Droid Monitor* are in continuous development. Each newly created functionality needs the necessary updates. If the web application and the application server updates can be made without the questioning the customers, Android application requires each user to grant permission on the mobile device.

Although at first glance this may seem to be an inconvenience, updates and promotional purposes are used to remind users of the Android application, because it runs in the background, so there is a risk to be forgotten.

16 External help

Another currently used approach designed to facilitate the product´s development is communication with different organizations or individuals that might be interested in investing time or money in this product. There is also the possibility of obtaining grants from the European Union, but steps in this direction are undergoing with difficulty. For the moment being, responses are expected from people who have shown minimal interest in this project, so that the situation seems to be moving in an upward direction.

17 Conclusions and future developments

The two main reasons for the existence of this project were educational and productive. The teaching objectives have been achieved by experimenting with a wide range of management activities and promoting the product, whereas the end result will be provided by success on the market. Therefore only the end user can give a realistic conclusion of the project, conclusion can be drawn out only from many years of activity and usage.

Bibliography

[1] ANSPDCP, *Home. ANSPDCP*, http://dataprotection.ro
[2] Obancea Dragos Iulian, *Primii pasi cu C#,* Editura Noua, 2009.
[3] Dorin Bocu, *Ingineria Softului*, Course, 2008

DRAGOŞ IULIAN OBANCEA
Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Computer Science specializaition
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
obancea_dragos@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Dynamic Data Structures – Modular design, implementation and particular application

Istvan Benimain Pallo

Teacher Coordinator: Dana Simian

Abstract

The purpose of this paper is to present the design and implementation of a modular application for working with dynamic data structures, double chained lists. The module is implemented in C ++ and can be used in practical applications in various fields. The paper presents the way how this application can be integrated in an e-learning system and in an on-line medical office application.

1. Introduction

In application's design, one of the most important elements consists in the proper choice of algorithms and data structures used. Dynamic data structures have high utilities because their advantages: efficiency in memory usage, ease of implementation in a modular way, easy modeling of problems in various fields. High schools and universities have in computer science curricula, many topics related to dynamic data structures.

The goal of this paper is to design and implement a modular application for working with dynamic double chained lists and to study the possibility of integrating the application in larger applications from different fields. We present, in this paper, two particular cases: integratio n in an e-learning system and an online medical office application.

This module is designed using object-orientated programming. We used C ++ as object orientated language. One reason is that, for me, at this moment, it is the only object orientated language studied. But application can be implemented using any object orientated language because the application logic does not change.

We chose dynamic double chained lists because it lends itself to easy modeling of a large number of problems form various fields.

User interaction is achieved through graphical window. For implementation in an e-learning system we designed graphical visualization of results.

There are many e-learning platforms (Frog, RM Learning Platform – [2]) and one of the purposes is to build an application that can be used for learning dynamic double chained lists. Our application can be integrated into an e-learning platform for computer science (such a platform can be found at [3]). To fully integrating this application into an e-learning is needed, that beside the application, to exist some documentation.

This documentation needs to include theory about dynamic structures in general and, in particular, theory about dynamic double chained lists.

There are many medical offices online sites (an example can be found in [4]) which use more complex applications but the purpose of this paper is to illustrate how dynamic double chained lists can be used into the design of a medical office online and not to develop a system more complex than the existing ones. Now, because we use this created module to a practical problem, we are forced to change some fields in our dynamic double chained list. We want to make a list with patient consultations in an ordinary day. For that we need a dynamic chained list, but also we can use a dynamic double chained list. The information contained in an element of the list are different from the information that our list, at this moment, can store so we need do redefine a small part from class declaration.

Pupils and students can use those dynamic structures of data in various ways because of their easily modeling.

The rest of the paper is organized as follows: in section two is presented the design and implementation of the application, in section three we present application interface, in section four we discuss about main feature due to which we can integrate this application into an e-learning system, in fifth section we see how this application can be integrated into a medical office online and last section include some conclusions.

2. Design and implementation

The application contains a basic part, which includes elementary operations with dynamic double chained lists and a part that varies depending of problem requirements. For example, application interface will differ. For e-learning application we will have a graphical display of the operations. Other problems may not require this visualization but only some normal operations on dynamic double chained lists.

The structure of a list element differs from one application to another. To achieve a greater flexibility we used a function that deals with loading data and this function can be modified from a problem to another without modify the rest of application.

The definition of the class lista is presented below.

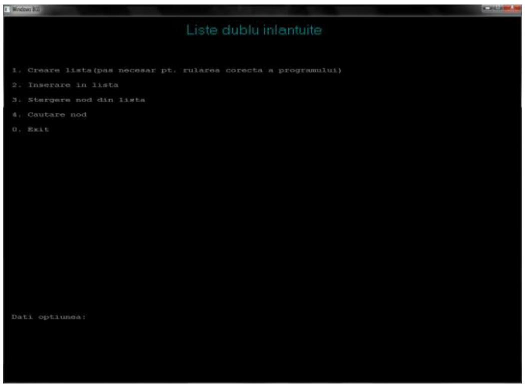
```
class lista_a
```

```
{
        public:
                lista_a(){}
                virtual ~lista_a(){}
                virtual int push1(int)=0;
                virtual int push2(int)=0;
                virtual int push3(int)=0;
                virtual int push4(int)=0;
                virtual int pop1(int&)=0;
                virtual int pop2(int&)=0;
                virtual int pop3(int&)=0;
                virtual int pop4(int&)=0;
                virtual int search(int)=0;
};


class lista;

class nod
{
        private:
                friend lista;
                int info;
                nod *ant;
                nod*urm;
                nod(int v,nod* p)
                {
                        info=v;
                        urm=p;
                        p->ant=this;
                        ant=NULL;
                        p->urm=NULL;
                }
 };

class lista:public lista_a
{
        private:
                nod* first;
                nod* last;
        public:
                lista_a(){first=NILL;last=NILL;}
                virtual ~lista_a(){}
                virtual int push1(int)=0;
                virtual int push2(int)=0;
                virtual int push3(int)=0;
                virtual int push4(int)=0;
                virtual int pop1(int&)=0;
                virtual int pop2(int&)=0;
                virtual int pop3(int&)=0;
                virtual int pop4(int&)=0;
                virtual int search(int)=0;
}
```

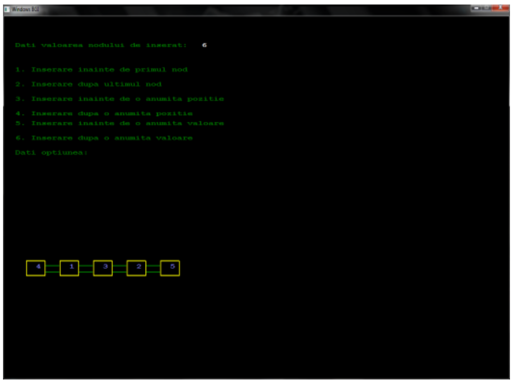
3. Application interface

The application interface enables data input and visualization of results using a graphic interface. The language of our interface is for the moment Romanian but it can be changed into English or other language with no great effort.

The main interface contains a five items menu: list initialization, add a node in the list, delete a node from the list, searching a node using a specified criterion, exit from the menu (Fig. 1).



```
                    Liste dublu inlantuite

1. Creare lista(pas necesar pt. rularea corecta a programului)
2. Inserare in lista
3. Stergere nod din lista
4. Cautare nod
0. Exit

Dati optiunea:
```

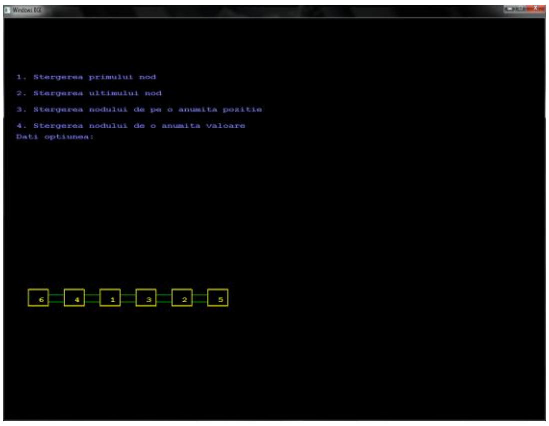


```
Dati valoarea nodului de inserat:   6

1. Inserare inainte de primul nod
2. Inserare dupa ultimul nod
3. Inserare inainte de o anumita pozitie
4. Inserare dupa o anumita pozitie
5. Inserare inainte de o anumita valoare
6. Inserare dupa o anumita valoare
Dati optiunea:
```

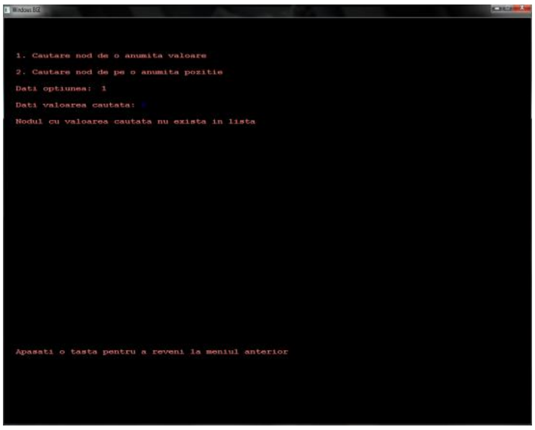
Fig. 1 Main menu

Fig. 2 Insert a node

A node can be added in different position in the list: before the first element, after the last one, before and after a node specified by address or by information (Fig. 2).



```
1. Stergerea primului nod
2. Stergerea ultimului nod
3. Stergerea nodului de pe o anumita pozitie
4. Stergerea nodului de o anumita valoare
Dati optiunea:
```



```
1. Cautare nod de o anumita valoare
2. Cautare nod de pe o anumita pozitie
Dati optiunea:  1
Dati valoarea cautata:
Nodul cu valoarea cautata nu exista in lista




Apasati o tasta pentru a reveni la meniul anterior
```

Fig. 3 – Delete Menu

Fig. 4 - Search Menu

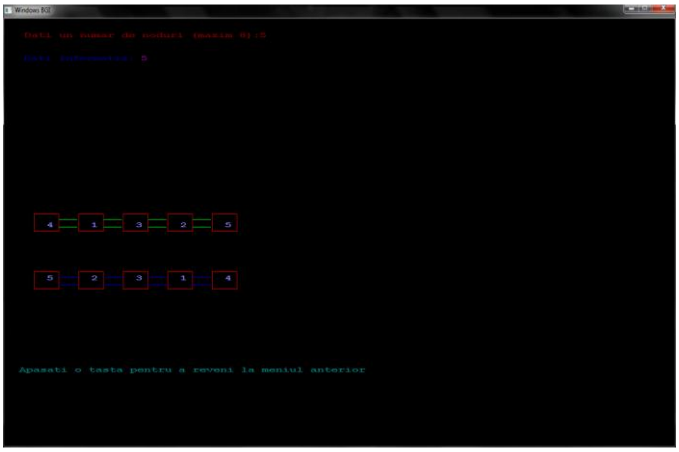
For deleting a node we also use a menu similar with this for adding a node (Fig. 3). In order to accomplish all the variants for add and delete a node we need a search a node by address or by information (Fig. 4).

4. Integration into an e-learning system

This application was designed to help learning of dynamic double chained lists. Through this application we intend to determinate the involvement of students and teachers in learning. For students it is easier to understand abstract concepts when they can interact with an application and see the results of interaction.

Because this application it was built for an e-learning system it was necessary to include into it graphic representation of the operations. It is easier for pupils and students to understand operations mechanism when they see what happens when they choose to do different operation with lists.

For e-learning application a fixed number (eight) of elements are used in the dynamic double chained list. This restriction is imposed by graphic window size and also in order to easily track operation that was made. The insertion and deletion of a node has a graphic visualization (Fig. 5)



```
Dati un numar de noduri (maxim 8):5

Dati informatia: 5
```

```
Apasati o tasta pentru a reveni la meniul anterior
```

Fig. 5 – Graphic representation of the operations made

5. Integration into a medical office online

Every medical office needs to schedule their patients. Nowadays it is a trend to automatically do this using a computer program. In Romanian medical offices, at this moment the maximum number of scheduled patients in a day is twenty. In our application we made the hypothesis that the schedule is made for an entire month.

First of all we need to know the name of patient, the date and hour of consultation. So we need to modify our list. One element should contain now a name and two numbers, one for date and one for hour.

A patient wants to make an appointment in a particular day, within the program of the medical for this day at an available hour. The queue structure would be proper for this operation, if we do not take into account the possibility of the choice of the hour by the patient. In order to use our module and to enlarge the possibility of our choice, we use a double list ascended sorted by the field hour. The sorting process is based on the direct insertion algorithm. Next, all operations will be made on an ordered list. It is allowed for the patient to choose the day and the hour. The program will return a warning message if the day is already full or/and the hour is not available

If one or many patients canceled their appointments, the application display a message and the patients will be deleted from the actual list.

6. Conclusions

This is an original application illustrating the vision of a student about how an e-learning application should look like. This application is not meant to be the best application, but it can be improved more and more.

There are many object orientated languages (a list can be found in [1]) and a future purpose is to study their performances in implementation process of our application.

References

[1] www.projectsofashok.blogspot.ro/2011/01/list-of-object-oriented-programming.html

[2] www.chip.ro/stiri/18356-top-10-platforme-e-learning

[3] www.informatica.upg-ploiesti.ro/moodle/

[4] www.romedic.ro/cabinete.php

PALLO Istvan Beniamin
Faculty of Sciences
University "Lucian Blaga" of SIBIU
Str. Dr. Ion Ratiu nr.5-7
550012 Sibiu
Romania
Email: eu.stefy@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Statistical analysis and models for trend of precipitation in Dobrogea

Andreea-Oana Petac
Teacher Coordinator: Alina Bărbulescu

Abstract

The aim of this article is twofold. Firstly, considering the annual precipitation series collected for a period of 41 years in the region of Dobrogea, Romania, we make their statistical analysis, consisting of normality, homoscedasticity, autocorrelation and break point tests. Secondly, we detect the trend evolution for the series in Dobrogea region using the wavelets approach.

1 Introduction

Management of water resources is of big importance for the human life and activity. Well-fit models of such time series can shed light upon past weather related phenomena and can help to explain future events. Annual precipitation is an essential component in the water budget [Bărbulescu and Băutu, 2009], especially for a region where the drought period is of 4-6 month per year.

We study the series of cumulated annual precipitation records registered in the period 1965 - 2005 at 41 secondary meteorological stations, situated in Dobrudja region, in the South - East of Romania. Our work comes to complete the studies performed on only 10 main precipitation series [Bărbulescu and Deguenon, 2011, 2012], in order de create a model for the trend evolution of precipitation in entire region of Dobrogea.

The paper is structured as follows: in the first section, some considerations regarding the studied series are provided. We continue with a presentation of the methodologies used to analyze the series and briefly present the mathematical background surrounding the wavelets methods. The next section contains the results of statistical analysis and the model for the precipitation trend. The final section contains the conclusions of the paper and the discussions of possible directions of future research.

Keywords: Annual precipitation, autocorrelation tests, time series, break tests, statistical analysis, wavelet

2 Methodology

The first step in our study was to perform statistical tests of normality, autocorrelation, homoscedasticity and break point existence in the time series.

We studied the series normality by means of the Jarque - Bera test [Jarque – Bera, 1980]. The null hypothesis of this test is:

H_0: The process is normally distributed,

and the opposite:

H_0: The process is not normally distributed.

The statistic Jarque – Bera test is defined as:

$$JB = \frac{n-p}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right)$$

where:

- *n* is the number of observations (or degrees of freedom in general);
- *S* is the sample skewness;
- *K* is the sample kurtosis.

This test has an asymptotic chi - square distribution with two degrees of freedom. If at a significance level α=0.05, JB > $\chi^2(2)$ from the table of χ^2 test, than the null hypothesis is rejected.

The autocorrelation of the time series is investigated analyzing the autocorrelation function.

The autocorrelation function at lag *h*, (*h* ∈ N*), associated to a time series $(X_t)_{t\in\overline{1,n}}$ is defined by:

$$\rho(h) = \frac{\sum_{t=1}^{n-h}(X_t - \overline{X})(X_{t+h} - \overline{X})}{\sqrt{D^2(X_t)}\sqrt{D^2(X_{t+h})}}$$

where $D^2(X_t)$ is the variance of X_t.

For a sample data $\{x_1, x_2, ..., x_n\}$ and a given confidence level, the values of the autocorrelation function (ACF) are calculated together with the deterministic confidence interval. If the values of ACF are inside this interval, then the hypothesis that the series is autocorrelated can be rejected [Brockwell and Davies, 2002].

For the break tests the null hypothesis that must be tested is:

H_0: There is no break in the series $(X_t)_{t\in\overline{1,n}}$,

against the opposite:

H_1: There is a break in the series.

The series homoscedasticity has been tested by Levene test [Levene, 1960]. The null hypothesis was:

H_0: The series is homoskedastic, i.e

H_0: $\sigma_1 = \sigma_2 = ... = \sigma_k$,

where *k* the the number of groups in which it is divided.

The alternative hypothesis is:

H_1: The series is heteroskedastic, i.e: at least a group variance differs from the others.

The test statistics is:

$$W = \frac{(n-k)}{(k-1)} \frac{\sum_{i=1}^{k} n_i \left(\overline{Z_{i\bullet}} - \overline{Z_{\bullet\bullet}}\right)^2}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(Z_{ij} - \overline{Z_{i\bullet}}\right)^2},$$

where:

$Z_{ij} = \left| x_{ij} - \overline{x_{i\bullet}} \right|$,

$\overline{x_{i\bullet}}$ - the mean of the elements in group *i*,

$\overline{Z_{i\bullet}}$ - the mean of the elements Z_{ij} in group *i*,

$\overline{Z_{\bullet\bullet}}$ - the general average of Z_{ij}.

At the significance level $\alpha = 0.05, H_0$ is rejected if $W > F_{k-1,n-k}$, where $F_{k-1,n-k}$ is the upper critical value from the table of Fisher test with *k*-1 and *n*-*k* degrees of freedom, at the chosen significance level, α.

In order to detect the existence of change points, we use within the investigation the following statistical methods.

The *Pettitt test* is a non - parametric one [Pettitt, 1979].

Hubert test [Hubert, 2000] is used to detect the multiple breaks in time series. The basis of this test is to reduce the series into at least 2 segments provided that the calculated means of the neighboring sub-series differ with a significantly value.

Both Pettitt and Hubert tests work even if the series are not normally distributed.

Lee & Heghinian test is applied under the assumption that the studied series is normally distributed. This method starts with the hypothesis that the precipitation series $(X_i)_{i \in \overline{1,n}}$ is one of independent variables, having a constant variation. Supposing that the break time follows a uniform distribution, this method determines the a posteriori probability distribution function of the parameters μ and δ, considering their a priori distributions.

Buishand [Buishand, 1984], as well as the Lee & Heghinian [Lee & Heghinian, 1977] methods, are based on the following model:

$$X_i = \begin{cases} \mu + \varepsilon_i \quad , & i = \overline{1,m} \\ \mu + \varepsilon_i + \delta, & i = \overline{m+1,n} \end{cases},$$

where *m* represents the break point, *m*, μ and δ being unknown, ε_i are random variables, independent and normally distributed, with null expectance.

In order to study the variability of the annual rainfall time series at different meteorological stations in Dobrogea region, wavelet analysis is applied.

The wavelet transform is a recent advance in signal processing that has attracted much attention since its theoretical development by Grossman and Morlet [Grossman and Morlet, 1984]. Its use has increased rapidly as an alternative to the Fourier Transform (FT) in preserving local, non-periodic, multiscaled phenomena.

It has advantage over classical spectral analysis, because it allows analyzing different scales of temporal variability and it does not need a stationary series. Thus, it is appropriate to analyze unregular distributed events and time series that contain nonstationary power at many different frequencies. Then, it is becoming a common tool for analyzing localized variations of power within a time series.

Wavelet is a special kind of waveform which must satisfy certain mathematical ”admissibility conditions” as:

$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \qquad (1)$$

Based on the upper equation, we recall the following one:

$$C_{\psi} = \int_{-\infty}^{\infty} \psi(t) dt = 0\,. \qquad (2)$$

The Mother Wavelet function is defined as below:

$$\psi_{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right) \qquad (3)$$

where a, b $\in R$, and a $\neq 0$. In (3) a is the scaling factor and b is the time position factor.

When dealing with hydrologic series, is always a discrete one, and for decreasing redundancy Discrete Wavelet Transform (DWT) is used in (4):

$$W_{j,k} = a_0^{-j/2} \int_{-\infty}^{+\infty} f(t)\overline{\psi}(a_0^{-j}t - kb_0)dt \qquad (4)$$

$W_{j,k}$ are the WT's output of time series *f*(*t*). When a_0 has a small value, it has a low resolution in frequency domain, but high resolution in time domain [Sang and Wang, 2008]

4 Results and Discutions

After performing the normality test, at the significance level $\alpha = 0.05$, the null hypothesis was rejected for only 3 series, it fails to be rejected for 21 series initial series, 9 series after a square root transformation, and 7 series after a logarithm transformation and a series after a Box – Cox transformation, with the parameter λ= -0.05 (Table 1).

The analysis of autocorrelation functions reveals the autocorrelation only for six series.

In Fig.1 the autocorrelation function of Posta series is presented (the bars represent the function values and the dotted lines, the limits of the confidence level, at the significance level of 0.05. We remark the autocorrelation of first order in the series.

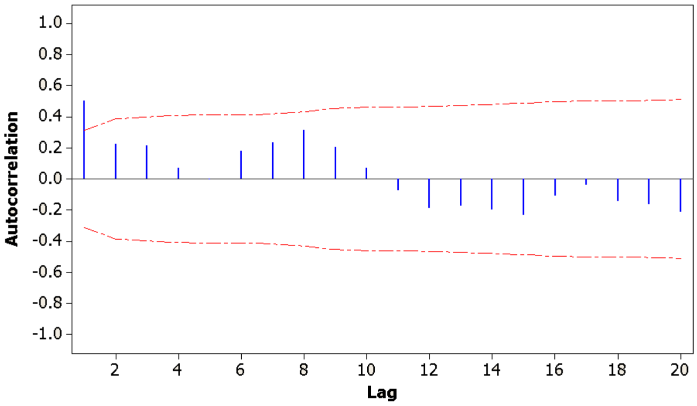


Fig.1. Autocorrelation function for Posta series

The break tests have been performed at the significance levels $\alpha = 0.1, 0.05$ and 0.01. the results are presented in Table 2, where the change point is also given. For a number of 17 stations the Buishard, Pettitt, Lee&Heghinian tests lead to the acceptance of the null hypothesis, at different significance levels, while the segmentation procedure of Hubert rejected it.

For nine series, the Buishand (Fig.2) and Pettitt tests rejected the null hypothesis, at different significance levels and Hubert test highlights at least one break point. For $\alpha = 0.05$, Both Pettitt and Buishard tests give the same result.

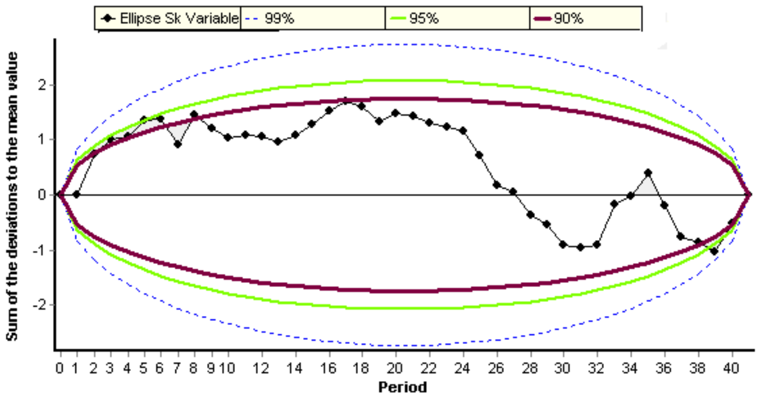


Fig. 2. Bois ellipse, associated to the Buishand test for Dăieni series (the break point existence rejected at all the confidence levels: 99%, 95%, 90%)

Table 1. Results of normality test

	Series	Normality test (α = 0.05)
1	Agigea	no
2	Albesti	yes after a square root transformation
3	Altan Tepe	yes
4	Amzacea	no
5	Baia	yes
6	Baltagesti	yes
7	Biruinta	yes after a square root transformation
8	Casian	yes
9	Casimcea	yes after a logarithm transformation
10	Ceamurlia	yes
11	Cerna	yes
12	Cheia	yes after a square root transformation
13	Cobadin	yes
14	Corbu	yes after a square root transformation
15	Crucea	yes
16	Cuza voda	yes
17	Daieni	yes after a logarithm transformation
18	Dobromir	yes
19	Dorobantu	yes after a logarithm transformation
20	Greci	yes
21	Hamcearca	yes after a square root transformation
22	Independenta	yes
23	Lipnita	yes after a square root transformation
24	Lumina	no
25	Mihai Viteazu	yes
26	Negru Voda	yes after a square root transformation
27	Negureni	yes
28	Niculitel	yes

	Series	Normality test (α = 0.05)
29	Nuntasi	yes
30	Pantelimon	yes after a logarithm transformation
31	Peceneaga	yes after a logarithm transformation
32	Pecineaga	yes after a square root transformation
33	Pestera	yes after a logarithm transformation
34	Pietreni	yes
35	Posta	yes after a square root transformation
36	Sacele	yes
37	Saraiu	yes after a Box and Cox transformation with λ= -0.05
38	Satu Nou	yes
39	Silistea	yes
40	Topolog	yes after a logarithm transformation
41	Zebil	yes

yes means that the null hypothesis is accepted.

Lee & Heghinian test and Hubert procedure detected break points (Fig.3).

Analyzing the results of the Hubert procedure we are able to observe that the majority of the break points is in 2003 and 2004. The series that presents break points in the period 2002 – 2004 are preponderant, fact that signifies a change in precipitation evolution at a regional level.

The results of Levene tests didn't reveal the series heteroskedasticity.

Table 2. Results of break tests

	Buishand			Pettitt				L&H	Hubert
Series/ α	0.01	0.05	0.1	0.01	0.05	0.1	year	year	year
Agigea	-	-	-	yes	yes	yes		-	2004
Albesti	no	no	no	yes	no	no	1990	1994	1994
Altan Tepe	yes	yes	yes	yes	yes	yes		2003	yes
Amzacea	-	-	-	yes	no	no	1994	-	1996
Baia	yes	yes	yes	yes	yes	yes		1973	yes
Baltagesti	no	no	no	no	no	no	1982	1982	1982
Biruinta	yes	yes	no	yes	yes	yes		2004	2004
Casian	yes	yes	yes	yes	yes	yes		2003	1972, 2003
Casimcea	yes	no	no	yes	no	no	1987	2003	2003
Ceamurlia	yes	yes	yes	yes	yes	yes		1972	yes
Cerna	yes	yes	yes	yes	yes	yes		1966	yes
Cheia	yes	yes	yes	yes	yes	yes		2003	2003
Cobadin	yes	yes	yes	yes	yes	yes		2004	yes
Corbu	yes	yes	yes	yes	yes	yes		1994	yes
Crucea	yes	no	no	yes	no	no	1981	1975	1975
Cuza voda	yes	yes	yes	yes	yes	yes		2003	2003
Daieni	yes	yes	yes	yes	yes	yes		2003	yes
Dobromir	yes	no	no	yes	no	no	1987	2003	1969, 1974, 2003
Dorobantu	yes	yes	yes	yes	yes	yes		2003	2003
Greci	yes	yes	yes	yes	yes	yes		2003	yes
Hamcearca	yes	no	no	yes	yes	no	1987	2003	2003
Independenta	yes	yes	no	yes	yes	no	1979	2003	yes
Lipnita	yes	no	no	yes	yes	yes		2001	2001

	Buishand			Pettitt				L&H	Hubert
Series/ α	0.01	0.05	0.1	0.01	0.05	0.1	year	year	year
Lumina	-	-	-	yes	yes	yes	-	-	yes
Mihai Viteazu	yes	yes	yes	yes	yes	yes		2004	yes
Negru Voda	yes	no	no	yes	no	no	1994	1994	1994
Negureni	yes	yes	yes	yes	yes	yes		2004	2004
Niculitel	yes	yes	yes	yes	yes	yes		2003	yes
Nuntasi	yes	yes	yes	yes	yes	yes		1973	yes
Pantelimon	yes	no	no	yes	no	no	1994	2002	1972, 1994, 2003
Peceneaga	yes	yes	yes	yes	yes	yes		2003	2003
Pecineaga	yes	yes	yes	yes	yes	yes		2004	yes
Pestera	yes	yes	yes	yes	yes	yes		2003	1967, 2003
Pietreni	no	no	no	no	no	no	1992	1992	1992
Posta	no	no	no	no	no	no	1994	1994	1994
Sacele	yes	yes	no	yes	yes	yes		2003	yes
Saraiu	yes	yes	yes	yes	yes	yes		2003	2003
Satu Nou	yes	yes	yes	yes	yes	yes		2004	yes
Silistea	yes	yes	yes	yes	yes	yes		2003	yes
Topolog	yes	yes	yes	yes	yes	yes		2003	1996, 1997, 2003
Zebil	yes	no	no	yes	yes	no	1981	1972	1972

yes means that there is no break point. The year gives the break point moment.

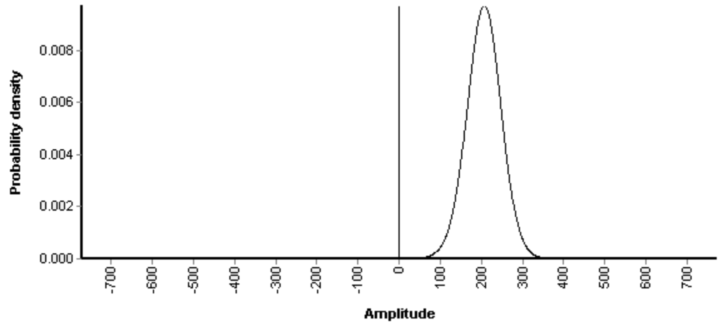


Fig. 3. Lee and Heghinian test. A posteriori probability density of a break amplitude for Pietreni series The break point is 1972

The models realized for some stations are presented in Figs. 4 and 6 together with the squared residual values.

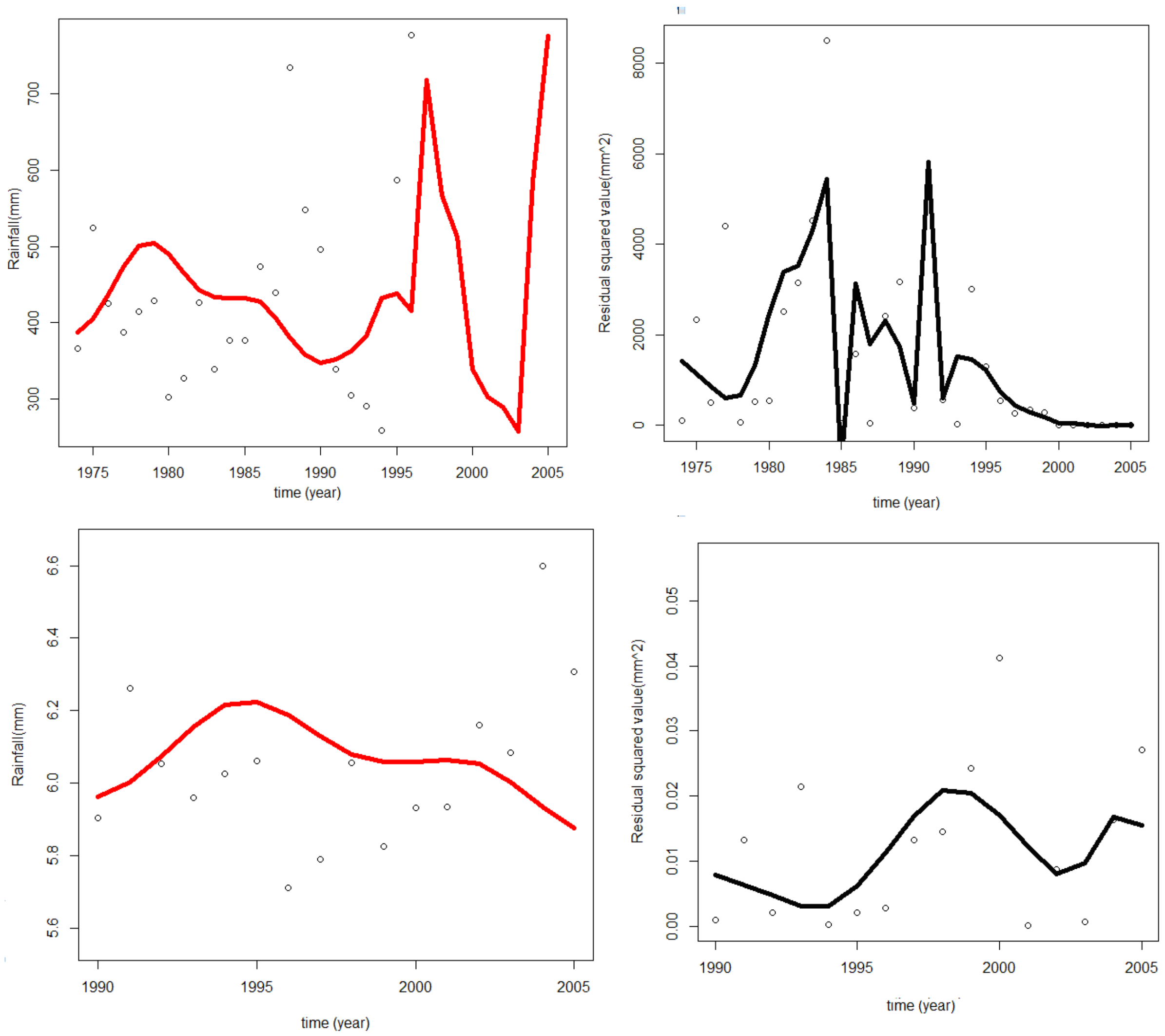
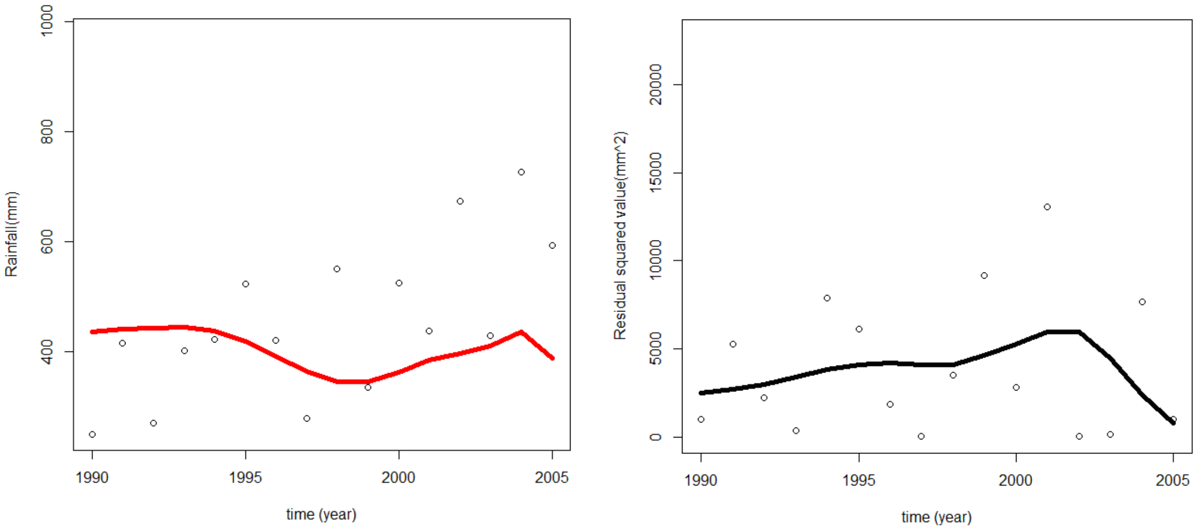


Fig.4. Model and residual squared values for Dorobantu series (the upper part) and Dorobantu series transformed by taking logarithm



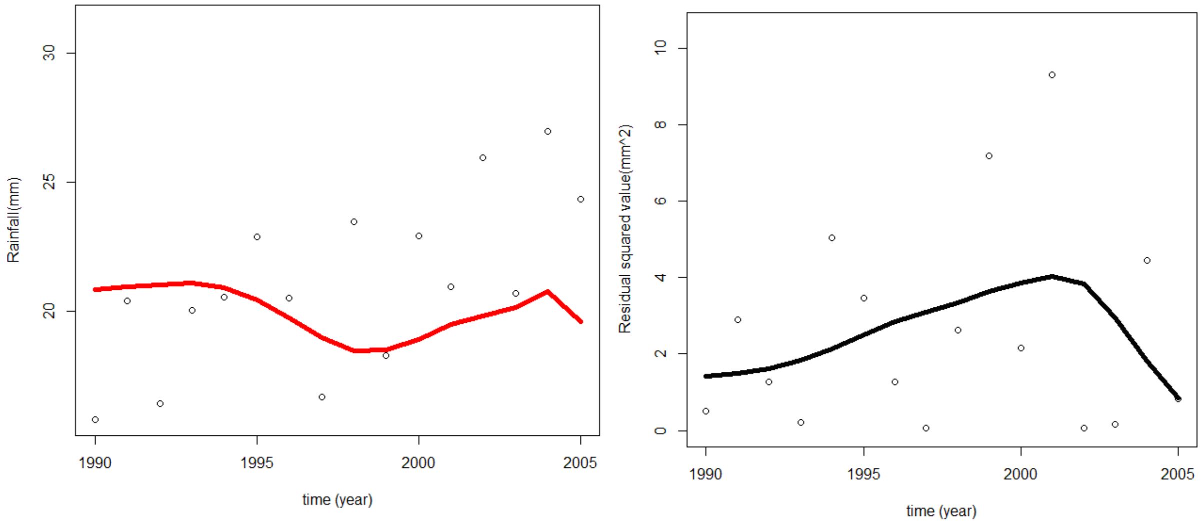


Fig.5. Model and residual squared values for Lipnița series (the upper part) and Lipnița series transformed by taking square root

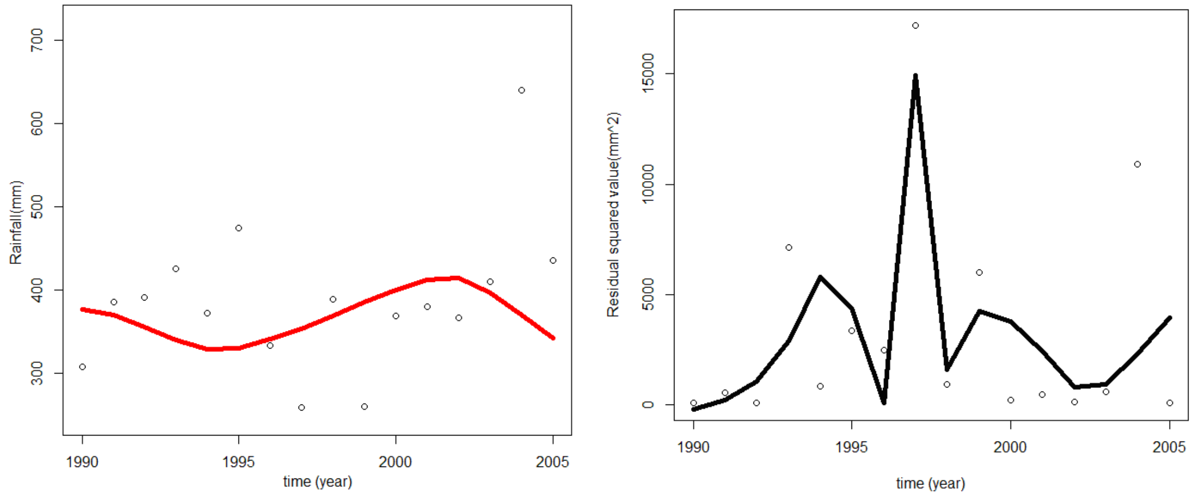


Fig.6. Model for Saraiu series

In order to decide if the models and the data have the same distribution (H_0 hypothesis), the goodness of fit tests (Kolmogorov – Smirnov and chi – square) have been performed. H_0 failed to be accepted for all the series.

The models' quality has been checked by calculating the standard errors.

For Dorobanțu series, the errors' standard deviation is 37.99 and for the transformed one, 0.078 for Lipnița series, the errors' standard deviation is 52.61 and for Lipnița, after the square root transformation, 1.49. For Saraiu series, the errors' standard deviation is 52.76.

Better models have been obtained for the transformed series.

Since the shape of the fitted curves are not uniform for all the series, in a future study we have to detect subregions with the same trend o precipitation evolution and to fit models for the specified areas.

3 Program

The program has been realized using the R software.

In order to realize our models, below are shown the steps that have been followed:

➔ Building a vector named X, that contains the time (year);

- ➔ Building another vector, Y1, with the precipitation values at a given station;
- ➔ Activate the package wavethresh;
- ➔ Generate another vector c, from ,Y1, containing a number of consecutive elements represented by a power of 2
- ➔ Fit the model using wavelets (waveletrain and hardthreshain)
- ➔ Plot the charts of the rainfall model and the residual squared values

References

[1] A. Bărbulescu, E. Băutu. Alternative models in Precipitation Analysis. An. Şt. Univ. Ovidius Math, vol.17(3): 45 – 68, 2009.

[2] A. Bărbulescu, J. Deguenon. Nonparametric methods for fitting the precipitation variability applied to Dobrudja region. International Journal of Mathematical Models and Methods in Applied Sciences, Issue 4, vol. 6: 608 – 615, 2012.

[3] A. Bărbulescu, J. Deguenon, D. Teodorescu. Study on water resources in the Black Sea region. Nova Publishers, USA, 2011

[4] A. Bărbulescu, J. Deguenon. Modeling the annual precipitation evolution in the region of Dobrudja. International Journal of Mathematical Models and Methods in Applied Sciences, Issue 5, vol. 6: 617 – 624, 2012.

[5] P. Brockwell, R. Davies. Introduction to time series, Springer, New York, 2002.

[6] A. Buishand. Tests for detecting a shift in the mean of hydrological time series, Journal of Hydrology, Vol.73: 51–69, 1984.

[7] Grossman, A. and Morlet, J.: Decomposition of Hardy functions into square integrable wavelets of constant shape, *SIAM J. Math. Anal.*, **15**, 723-736, 1984.

[8] C. M. Jarque, A.K. Bera. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Economics Letters 6(3): 255–259, 1980.

[9] P. Hubert. The segmentation procedure as a tool for discrete modeling of hydro-meteorological regimes. Stochastic Environmental Research and Risk Assessment, Vol.14: 297–304, 2000.

[10] A. F. S. Lee, M.S. Heghinian. A Shift of the Mean Level in a Sequence of Independent Normal Random Variables - A Bayesian Approach. Technometrics, 19(4): 503-506, 1977.

[11] H. Levene, In Olkin et al. (eds), Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling. Stanford University Press, pp. 278–292, 1960.

[12] A. N. Pettitt, A non-parametric approach to the change-point problem. Applied Statistics, Vol. 28, No. 2: 126–135, 1979

[13] Y-F. Sang, D. Wang, New Method for Estimating Periods in Hydrologic Series Data. Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008.

State University Press, 1989.

ANDREEA PETAC
Ovidius University of Constanța
Department of Mathematics and Computer Science
124, Mamaia Bd., 900527, Constanta
ROMANIA
E-mail: andreea.petac@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Graphs and permutations based applications for planning a sports competition

Andreea Petre, Petre Nistor, Ionut Dragomir
Coordinating Teacher: Gabriela Moise

Abstract

As the technology evolves, scientists try to adapt to the new requirements with regards to software. Nowadays, there are numerous programs on the market that are supposed to do humans' jobs or, at least, make them easier. These are used in all fields of activity: accountancy, industry, arts, security, and even medicine. Taking into consideration all the devices and hardware that have recently been invented, there are more and more requirements to be fulfilled by their operating systems. Current computer programs are mostly oriented towards offering acceptable solutions that are not always the most effective ones. Flexibility should be a keyword to obtain an appreciated result.

The number of programmers who are aiming to achieve success by conceiving brilliant applications is rapidly increasing. All fields of activity have awakened their interest, including sports competitions. Such competitions have always been interesting for the spectators and, at the same time, considered a nightmare for the people who manage them. In order to organize them, a strict algorithm needs to be implemented.

In this paper we are going to analyze two different ways of planning matches between teams without any repetitions or mistakes that could interfere with the course of the games. We propose programs based on graphs and permutations that can be easily translated into all programming languages.

1. Introduction

Planning a sports competition can be carried out manually, case in which the instructional decisions are to be taken by the organizers directly. This means that every match should be thoughtfully scheduled, avoiding any repetitions or inconsistencies.

There are different ways of planning a match, depending on the sport – every team plays a single match with each team or by organizing a championship. In the last case, the winner A of a match should play with another winner B of another match and so on. Each of these possibilities is applicable using graphs and even permutations. The algorithms presented in this paper generate random solutions, thus there will not be any suspicion of fraud or arrangements. The most common example of sports competitions might be football matches, which seem to be more and more interesting from some peoples' point of view. Furthermore, we should not forget about

tennis, basketball, handball or chess. The way of organizing them is similar to planning football competitions, thus it is sufficient to implement a single, portable algorithm.
The rest of the paper is structured as follows:

- in section 2 we present the algorithms used for planning sports competitions that were already implemented;
- in section 3 we describe how sports competitions could be organized;
- in section 4 we emphasize the way our implementations actually work;
- in section 5 we state the importance of our work and present some other future directions of study.

2. Related work

In the last three decades or so, the automated construction of sports competition schedules has received considerable attention. Among the approaches that have been proposed in the last decades, there are exact algorithms (including methods based on combinatorial design theory) and heuristics, as presented in the article 'Applying Heuristic Methods to Schedule Sports Competitions on Multiple Venues', written by E.K. Burke, D. de Werra, J.D. Landa Silva and C. Raess[1].

There is a big interest in the problem in which a set of N teams must compete on a set of S neutral venues over T timeslots. A schedule should be constructed so that the competition is as balanced and as fair as possible. A feasible organization must satisfy some constraints: only one match can take place on a given venue at a given timeslot and each team can compete in exactly one venue in a given timeslot.

A particular case is when the number of competitors (N) equals the number of timeslots (T) and it is twice the number of venues (S). Then, each team competes N times and therefore, each team competes against one of the other teams exactly twice (this can be called the repeated match). There are three constraints that should be satisfied. "First, each team must play against each of the other teams at least once. Second, each team must compete in each venue exactly twice. Finally, any pair of teams should not compete against each other more than one time on the same venue."[1] The actual problem is to find a schedule that satisfies all the constraints.

Urban and Russell tackled the above scheduling problem using integer goal programming. They reported optimal results (perfectly balanced schedules) for the cases in which N equals 4, 6, 8 and 10. They also reported best-known results for problems with N equal to 12, 14 and 16.

They observed that the algorithm could easily find solutions for some cases (N=8 or N=10). However, there are some other cases in which the solution is not feasible (N= 6). They also noted that this approach consumes too much time when it comes to larger problems (N>10). For those cases, they proposed to run the algorithm for a fixed number of iterations followed by a post-processing phase that seeks to improve the quality of the schedule.

Recently, de Werra et al. designed constructive algorithms to solve some specific cases of this problem.

Another possible solution to the problem of organizing sports competitions could be the genetic algorithms, as mentioned in the thesis of Andrew B. Goldberg [2]. Genetic algorithms use ideas of natural evolution, specifically Darwin's notion of the "survival of the fittest" to solve optimization problems. Beginning with an initial random population of feasible solutions called *chromosomes*, a genetic algorithm generates progressively more desirable solutions over a series of generations by mimicking the processes occurring on DNA in real chromosomes. A solution's fitness is determined by applying an evaluation function, which incorporates a human scheduler's knowledge in distinguishing better and worse solutions. In an indirect encoding, the algorithm consists of using two permutations of team identifiers in the range [0, N-1], where N is the number of teams. The search space thus contains N! x N! solutions. A schedule-building routine decodes the chromosome to form a schedule in two steps. First, it generates the game match-ups per round. Then it applies a set of constraint-based rules to choose the home team for each game. The following algorithm, adapted from Leonard [3], generates a set of games for each week in a season:

1. Use the first permutation in the chromosome to obtain an ordered list of teams, indexed from 0 to N-1. If the N number is odd, then one 'virtual' team is added at the start of the list.
2. Create one round of games by pairing up teams: team 0 vs. team N-1, team 1 vs. team N-2, etc.
3. Assign the current round to week w, where w is the position of team 1 in the second permutation.
4. Rotate teams between positions 1 and N-1, leaving team 0 in place. That is, team 1 becomes team 2, team 2 becomes team 3 … and team N-1 becomes team 1.
5. Repeat steps 2 through 4 until the original team in position 1 returns to position 1.

3. Background of sports competitions

While some sports (such as fishing or hiking) have been viewed as primarily recreational, most sports are considered competitive. The majority involve competition between two or more people (sometimes using horses or cars). For example, in a basketball game, two teams compete against one another to determine who can score the most points. When there is no set reward for the winning team, many players gain a sense of pride. In addition, important rewards may also be given. A regularly scheduled (for instance annual) competition meant to determine the 'best' competitor of that cycle is called a championship.

Sports competitions are generally broken down into three categories: individual sports, such as archery, dual sports, such as tennis and team sports competition, such as cricket or football. While most sports competitions are recreation, there are several major and minor professional sports leagues throughout the world. The Olympic Games, held every four years, is usually regarded as the international pinnacle of sports competition.

Competitive sports are governed by codified rules agreed upon by the participants. Violating these rules is considered to be unfair competition. Thus, sports provide artificial (not natural) competition; for example, competing for control of a ball, or defending territory on a playing field is not an innate biological factor in humans. Athletes in sports such as gymnastics and competitive diving compete against each other in order to come closest to a conceptual ideal of a perfect performance, which incorporates measurable criteria and standards which are translated into numerical ratings and scores by appointed judges. However, in sports like football, a strict set of rules must be established. There might be different ways of organizing them. One of these is that each team plays with each of the other teams once. A second possibility could be that every two teams play a match, the loser leaves the competition and the championship continues until there is only one team left.

4. Description of the applications

In mathematics and computer science, graph theory is the study of graphs. These are mathematical structures used to model pair wise relations between objects from a certain collection. A *graph* in this context is a collection of *vertices* or *nodes* and a collection of edges that connect pairs of vertices [4]. Graph coloring is a special case of graph labeling. It is an assignment of labels traditionally called *colors* to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph so that any two adjacent vertices cannot share the same color. This is called a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that any two adjacent edges cannot share the same color. A face coloring of a planar graph assigns a color to each face or region so that any two faces that share a boundary are not of the same color [5]. Many problems that have to do with graph coloring can be solved through one of these techniques: The four-color theorem[6], The strong perfect graph theorem[7], The Erdős–Faber–Lovász conjecture[8], The total coloring conjecture[9], The list coloring conjecture[10], The Hadwiger conjecture (graph theory)[11]. Our problem, the football match schedule can be interpreted as a graph coloring one and treated as such. However, we also took this problem outside the graph boundaries and tried a different approach.

The first program we created is a graph-based application, a graph coloring program. Its goal is to make all the elements on columns and rows unique, like a Sudoku game. The functions used in this algorithm are:

procedure read_data; //reads 'n' which is the number of teams

procedure build_graph;

/* In this procedure we initialize the matrix with 0 and afterwards we insert numbers from 1 to n-1 on the first line, starting from the second column. The matrix satisfies the condition that on the main diagonal all numbers are equal to 0.*/

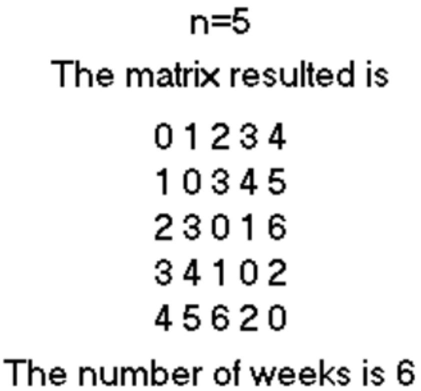
procedure coloring_graph;

/* After we have completed inserting elements into the second line, starting from the third column we check if the element 1 is on the same column or line, using two functions, one for lines and one for rows. If there is not, we add 1 to the matrix. If there is, we check again until we find a number to add. This sequence is repeated until there are no more lines to complete in the matrix.*/

procedure write_results;

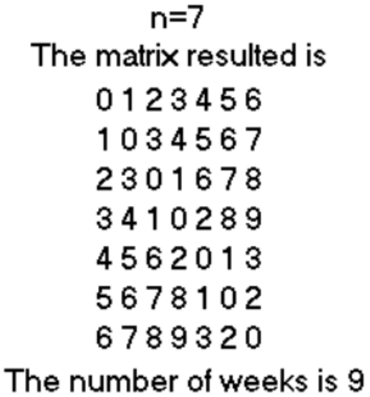
/*The final matrix, after the sequences above were executed, is written in a text file. This matrix contains elements that are unique on each line and column. The week when a certain match will take place can be found in the position determined by i and j in the matrix, the indexes i and j being the teams that participate in that match. Also, the minimum number of weeks required to play all the matches in the competition is the highest number in the matrix. This number is printed on the screen.*/

The results obtained for n=5 teams and n=7 teams are shown in Fig. 1 and Fig. 2.



```
n=5
The matrix resulted is
0 1 2 3 4
1 0 3 4 5
2 3 0 1 6
3 4 1 0 2
4 5 6 2 0
The number of weeks is 6
```

Fig. 1



```
n=7
The matrix resulted is
0 1 2 3 4 5 6
1 0 3 4 5 6 7
2 3 0 1 6 7 8
3 4 1 0 2 8 9
4 5 6 2 0 1 3
5 6 7 8 1 0 2
6 7 8 9 3 2 0
The number of weeks is 9
```

Fig. 2

Our second algorithm uses a vector to generate all the possible encounters, a list which memorizes all the games that took place and a number of counters and conditions to select the valid ones from the possible matches. The schema of this program is presented in Fig. 3.

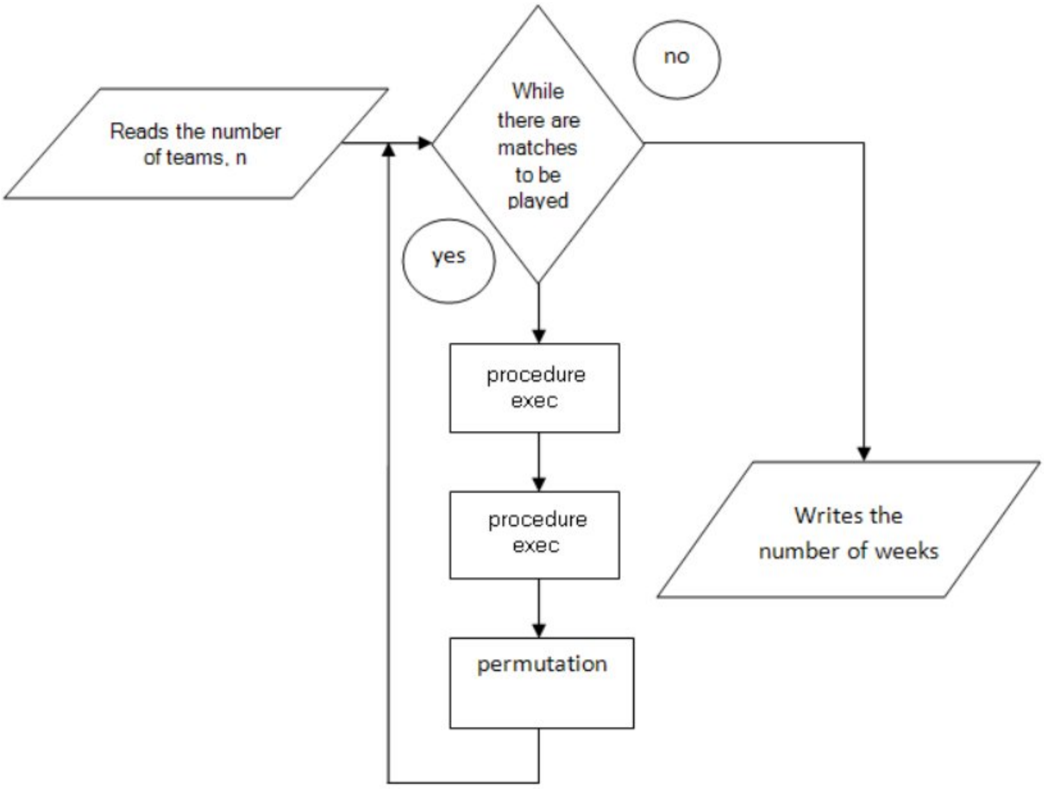

Reads the number of teams, n
While there are matches to be played
no
yes
procedure exec
procedure exec
permutation
Writes the number of weeks

Fig. 3

The variables that are used in our program are:

- *n* = the number of teams
- *k1* and *k2* = the teams to be tested if they have already played a match
- *l* = the list in which we put the teams that played a match
- *c* = a copy of 'l'
- *i1* and *i2* = the fields in the 'l' list
- *a* = the vector which memorizes the teams that are playing in the current week
- *b* = the vector which is being permuted
- *y* = the total number of matches to be played
- *i* = counter for going through vector 'b'
- *nr* = the number of weeks necessary for playing all the matches
- *o* = the number of teams that have played each week
- *s* = the number of matches that have been played at a specific time

The program reads the number of teams that are supposed to play and then performs the initial permutation. Next, it initializes all the counters and also the vector which memorizes the teams that are playing in the current week, in our program variable 'a'. After this, the number of games to be played, kept in variable 'y', is set. It uses a 'while' structure to repeat the program sequence until that number of games has been reached, creating a loop. In this loop, we make two calls of the 'exec' procedure, used to generate all the possible team combinations for the current permutation. We have to make two calls to the same function because two weeks are needed to go through the combinations offered by a single permutation, before a new one is needed. The 'exec' procedure mainly consists in a 'for' structure which crosses the vector 'b' and verifies each team combination, obtained from the numbers 'i', showing the position in the vector and b[i], a number between 0 and n-1. These combinations are verified using the 'ok' procedure.

```
procedure ok()
   If l<>NULL
   return 0
   else
        c<-l
        ok1<-0
        repeat
           if (c->i1=k1 and c->i2 = k2)
           or (c->i1 = k2 and c->i2=k1) then
                ok1=1
           else
                c=c->next
        until c=NULL and ok1=1
   return ok1
```

/*This procedure returns 1 if the match has already taken place or 0 if not. */

If an eligible pair is found (the match has not already been played and none of the teams are playing in the current week), then a match is created by adding it in the 'l' list. Then, the match is printed on the screen, the counter 's' is incremented and the value in the position from the vector 'a' corresponding to both teams is changed to '1', marking them as 'played' in the current week.

The 'exec' procedure was implemented as follows:

```
procedure ok()
o=0
for i<-1,n
        if ok(i,b[i],l)=0 and a[i]=0 and a[b[i]]=0
                a[i]<-1
                a[b[i]]<-1
                s<-s+1
                o<-o+1
                write i,b[i]
                //teams are added to the list
```

A decisional structure is used in this sequence to mark the end of a week, by increasing the 'nr' counter, adding a sign on the screen (a number of underlines in our case) and resetting the number of matches played in the current week, variable 'o', when the latter reaches a value equal to the number of teams or half of it. The variable 'o' was included to allow the program to track how many teams have been played each week, knowing that there cannot be played more matches than a number equal to half of the number of teams in the competition each week, thus eliminating useless repetitions.

```
if o=n or o=n/2
        nr<-nr+1
        o=0
        break //stops the 'for' loop
```

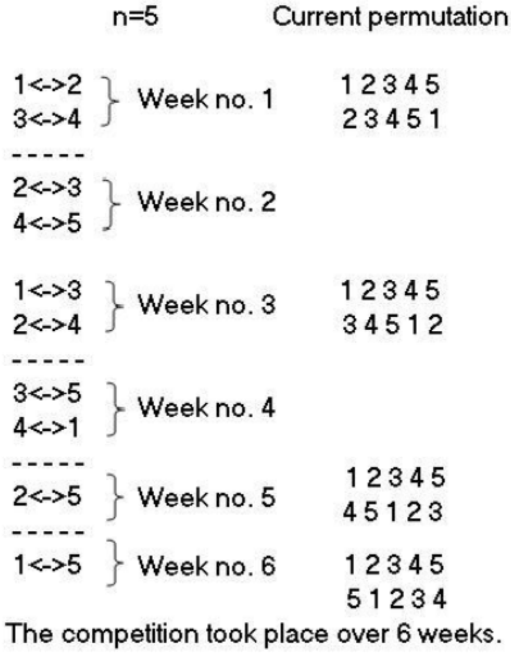
If the 'for' loop ends and the 'o' variable has a value different from '0', it means that the week did not end in the repetition and there are no other possible pairs for the current week and permutation, so the program forces the week to end anyway. After the loop and week have ended, the vector 'a' is reset, to allow new pair combinations. When the two calls of the 'exec' procedure are over, a new permutation is made, creating new possible team combinations and crossing again the 'while' loop.

The main program is structured as follows:

```
while s<y
        exec
        //the week ends
        nr<-nr+1
        //vector ‘a’ is reset to 0
exec
//the week ends
nr<-nr+1
//vector ‘a’ is reset to 0
//vector ‘b’ is permutated
```

When the program exits the ‘while’ loop, the number of weeks is memorized in the variable ‘nr’ and it has had all the encounters written on the screen.

For n=5 teams, the number of weeks necessary for playing all the matches is equal to 6. The actual schedule of the matches is shown on the screen. They are separated by hyphens which mark the end of a week. Also, the permutations that have been used are written in a text file so that the way the program works could be easier to understand. The result is presented in Fig. 4.



```
          n=5          Current permutation:

1<->2 }                 1 2 3 4 5
3<->4 }  Week no. 1     2 3 4 5 1
-----
2<->3 }
4<->5 }  Week no. 2

1<->3 }                 1 2 3 4 5
2<->4 }  Week no. 3     3 4 5 1 2
-----
3<->5 }
4<->1 }  Week no. 4
-----                   1 2 3 4 5
2<->5 }  Week no. 5     4 5 1 2 3
-----
1<->5 }  Week no. 6     1 2 3 4 5
                        5 1 2 3 4
The competition took place over 6 weeks.
```

Fig. 4

5. Discussions and conclusions

The problem of planning sports competitions may be approached through different techniques that are not apparently connected. Despite the fact that the algorithms presented in this paper are plausible, these are not the only solutions. The programs above were created as a response to the world’s continuous concern regarding software by using two basic methods: graphs and permutations. Both methods are easy to use and understand, thus the person who wants to organize a competition might choose to implement the algorithm which is considered to be the most explicit. However, there might be some future directions regarding this field of activity, as the technology evolves and the society becomes more and more interested in computer science. The algorithms we presented can be improved in order to make the execution faster and to minimize the storage space occupied by data. In the future, these programs might be optimized or adapted to a visual language, being easier to use. Furthermore, we should take into account the problems that can cause the cancelation of a game, like weather or some other restrictions (each team cannot play more than two games with any other team in one season, etc). In time, sports competitions might change and there will be some other rules and these algorithms will have to be adapted.

As for our algorithms, the goal was to find the number of weeks that would take for a competition with ‘n’ teams to be completed. The first program we implemented is more efficient for this purpose alone, but it cannot be modified in order to satisfy any type of conditions, such as the cancelation of a match and it would increase the number of instructions to be executed. In this way, efficiency is reduced in order to display all the matches that are supposed to be played. Since the second algorithm was created to print the matches on the screen, its inefficiency, but also the changes that might be made on the code could be reduced, making a more flexible program.

6. References

[1] E.K. Burke, D. de Werra, J.D. Landa Silva and C. Raess, *Applying Heuristic Methods to Schedule Sports Competitions on Multiple Venues*, University of Nottingham, UK, Chaire de Recherche de Operationelle EPFL Lausanne, Switzerland, 2009

[2] Andrew Brian Goldberg, *Highly Constrained Sports Scheduling With Genetic Algorithms*, Department of Mathematics and Computer Science of Amherst College, 2003

[3] Jason Leonard, *Interactive Game Scheduling with Genetic Algorithms*, Master’s thesis, Royal Melbourne Institute of Technology University, Melbourne, Australia, 1998

[4] http://en.wikipedia.org/wiki/Graph_theory accessed at March 29, 2013

[5] http://en.wikipedia.org/wiki/Graph_coloring accessed at March 29, 2013

[6] P. J. Heawood, *Map-Colour Theorem*, Quarterly Journal of Mathematics, Oxford, 1890

[7] C. Berge, *Stochastic graphs and strongly perfect graphs - a survey*, Southeast Asian Bull, 1983

[8] Paul Erdős, *On the combinatorial problems which I would most like to see solved*, Combinatorica, 1981

[9] Hugh Hind, Michael Molloy, Bruce Reed, *Total coloring with Δ + poly(logΔ) colors*, SIAM J. Comput, 1998

[10] Tommy R. Jensen, Bjarne Toft, *Graph Coloring Problems*, New York: Wiley-Interscience, 1995

[11] Hugo Hadwiger, *Über eine Klassifikation der Streckenkomplexe*, Vierteljschr. Naturforsch. Ges. Zürich, 1943

Andreea PETRE
PG University
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
Romania
E-mail: andreea_petre92@yahoo.com

Petre NISTOR
PG University
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
Romania
E-mail: nistor_petre@ymail.com

Ionut DRAGOMIR
PG University
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
Romania
E-mail: dragomir.ionutz92@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Online Banking System with CORBA, Hibernate and Struts

Petrescu Octavian Florin
Teacher Coordinator: Tudor Balanescu

Abstract

This project is an online banking system which provides its user a desktop ATM application which interacts with its own server, a web application through the use of which he can manage his own account and an application with more privileges which can manage, create and delete accounts. The project is created in Java programming language, based on Model-View-Controller architecture using the Struts 2 framework. JSPs are used for web applications and the Swing package for ATM. The server is created using the CORBA technology (The Common Object Request Broker Architecture). Hibernate is used to store data from the system and Tomcat 7 as a server. As innovation, the project combines Java portability with CORBA standard, the efficient way to interact with databases provided by Hibernate and advantages of the Model 2 MVC pattern implemented by Struts 2.

1 Introduction

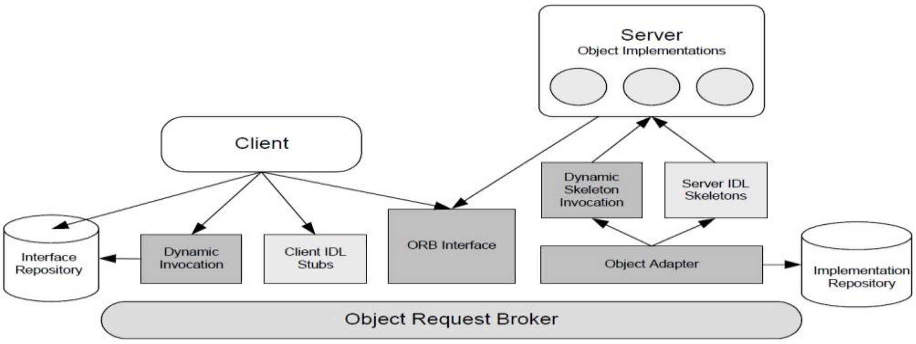
The notion of Internet Banking is the possibility of interaction, using the internet, for a client (under the generic name of user) with the bank where he usually possesses a bank account. More specifically, a system of Internet Banking offers customers password-based access to the information related to their accounts through a computer with internet access. The system provides the possibility to query the balance, to view transactions for a certain period of time, to transfer money from one account to another. Other works are written on this topic, the most relevant of these use JSPs and SQL as database [6] and [7]. At the moment, there are several applications that integrate Struts and Hibernate technologies, but this paper introduces the concept of integrating Struts and Hibernate to the world of internet banking software using the benefits of distributed systems such as: scalability, wideness, heterogeneity and fault tolerance.

2. Project structure

The project is implemented in Java using as IDE Eclipse Juno. It consists of 3 parts:

- a desktop application representing an ATM
- a server that manages the requests of the ATM having also database access
- two web applications, one in which the user can manage his account without the possibility of closing the account and another one that has not only the privileges of administering accounts, but also of creating and closing accounts.

The first component of a distributed application is, as in our case, the IDL interface. It is compiled using the Java IDL compiler and there are generated the stub files and skeletons files.
Classes generated in stubs are called proxies and they are responsible for accepting requests from the client and for sending messages through the orb to the suitable object implementation.
Skeleton files are required in the object implementation which are instantiated in the server. The full structure of the application is shown below [3]:



Central Components of CORBA-Based Applications

2.1 Implementation of ATM

The desktop application is created using the Java Swing package and enables balance query and money withdrawal. It is used the architectural pattern Model-View-Controller. The model, the logic part of the application, has the responsibility of operations on the data, the user's authentication and integration of classes that allow information processing from the database of the system. It also interacts with the server by sending requests and waiting for results. The view is the interface seen by the user and which displays the results received from the model. The controller connects the model and the view and calls certain functions to be used by the model to process (retrieve, update) data, after which the resulting information will be sent to the view for display.

2.2 Creating the Server

The server extends the specific POA class (Portable Object Adapter) to call methods through the skeleton and implements the methods specified in the IDL. In our case, the server will receive the request from the client, it will access the database in order to query the balance or carry cash extraction operation, and, after that, it sends the response to the client. Accessing the database will be performed using the open-source framework Hibernate, that offers persistence to system's data. Queries are performed using HQL (Hibernate Query Language), a powerful query language that apparently is similar to SQL.

2.3 Implementation of web applications

Web applications are designed to meet the demands of MVC Model 2 used by the Struts framework [2], [5]. The latter has ActionServlets that control the navigation and Action type classes that are used for the business-logics. When the ActionServlet receives a request from the container it uses the URI (or the path) to determine what type of Action class has to be used to satisfy the demand. This Action class can access the input using getters and setters [1], can validate the data and perform specific actions for this Action class. The next page will be determined using the execute method(). This method will return a String; the ActionSupport class has some constants such as: SUCCESS, ERROR, INPUT, LOGIN and NONE, which should be returned. Also, in the execute() method we access the database using hibernate [4] and perform all actions we need. In struts.xml we have all our Action classes with their results and for each result we set next view(JSP). For each specific action we have a specific JSP which will be shown. The first application is designed to give the customer the chance to view the balance and to view transactions carried out over a period of time. Account access is performed using an ID and a password. The other application is designed for the operator at the counter because this has the privilege of opening and closing accounts, and the account access is also performed using an ID and a password. The applications are created using Eclipse Juno and run on Tomcat 7.

3. Conclusions

Presently, there are several programming languages that can create projects at the same level as the project outlined above, the most used are PHP and ASP. However, Java remains one of the most powerful languages because it is independent from the platform and it has many frameworks that can be integrated to increase the application performances. Using Java and the Struts framework ,a project has been created, being independent from the platform and which manages a database in an efficient way with the help of Hibernate and which interacts with the user through a desktop interface and also through web sites. Using CORBA standards, the application has the advantages of distributed systems and it is independent from the platform

4 Further work

As further improvements, one target is to add the possibility of paying some bills and of making money transfers; after creating your account, an e-mail with a message will be automatically sent to the holder, and the other aim is to show the exchange rate which will automatically update after the BNR rate. I also plan to add a Model Checking module for the system specifications.

5 References

[1] Bruce Eckel, *Thinking in Java 4Th Edition*, 2007
[2] Srikanth Shenoy, *Struts Survival Guide*
[3] Markus Aleksy, Axel Korthaus, Martin Schader, *Implementing Distributed Systems with Java and CORBA*
[4] http://www.hibernate.org/
[5] http://struts.apache.org/
[6] http://www.techzoo.org/projects/online-bank-management-system-project-in-java.html
[7] http://www.scribd.com/doc/18028736/Online-Banking-System

Octavian Florin PETRESCU
University of Pitesti
Department of Mathematics and Informatics
Str. Targul din Vale nr. 1, Pitesti, Arges
petrescuflrn@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Hand Gesture Interpretation Unit

Adrian Popa
Coordinator: Dana Simian

Abstract

This paper presents an original project which aim is to design a glove equipped with an array of sensors and to implement the source-code by which a user, using the glove, can interact with a computer. These interactions consist in mouse - move and click events, as well as sign language interpretation and keyboard event simulations.

1 Introduction

The topic of this paper is human – computer interaction. This interaction is achieved by means of a glove that sends messages to the computer. These messages are interpreted by a program that simulates mouse and keyboard events.

In recent years I have worked on a number of electronics and software projects all of which had in common the need of human interaction. As a result I have developed a glove by which I can manipulate features of several different projects.

The project presents a glove and a Java application, written by myself. The glove consists of several bend sensors and a voltage divider circuit, all of which made by myself, an Arduino microcontroller and an inertial measurement unit.

The aim of this paper is to explain the method I used in making the bend sensors, the concepts behind sign language interpretation and the structure of the software I have written, used to interpret the signals sent from the glove.

2 The glove

2.1 Bend sensors

The bend sensors are the main components of the glove. They can be compared to a potentiometer, in the sense that, when bent, they give out a different electronic resistance value. This change is detected by the microcontroller and therefor can determine how much a certain finger has been bent.

The sensors' main components are strips of plastic [1]. This plastic is made out of PET and is slightly conductive. Antistatic bags are made using this kind of plastic. The way it works is that when pressure is applied to the plastic strip, it changes its electronic resistance. This pressure can also be achieved by bending.

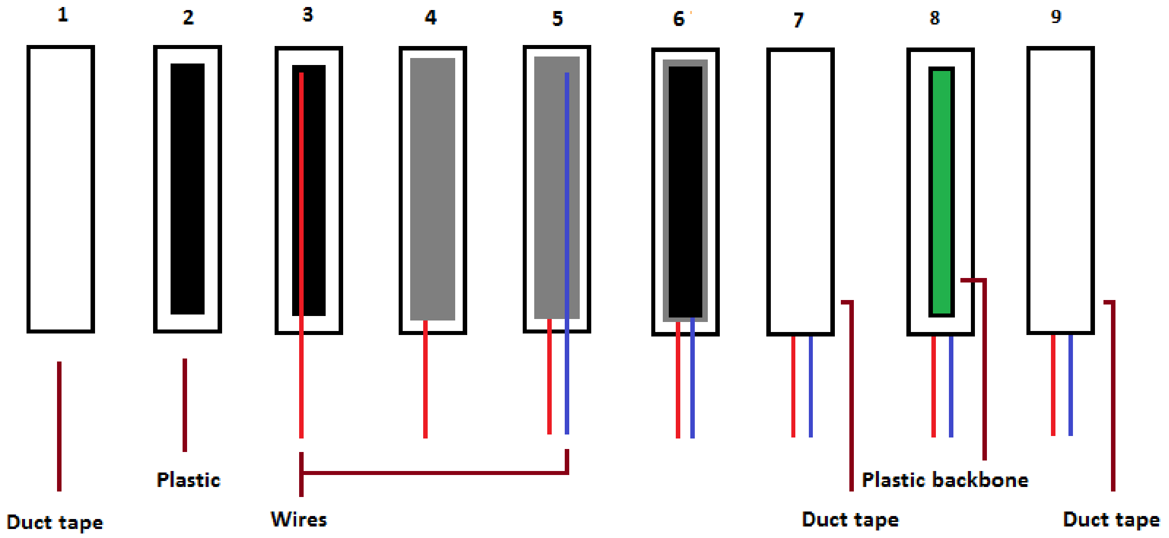


Fig. 1 – Bend sensor producing process

For this project I have made 5 bend sensors to determine each fingers bend-value. The sensors contain 2 wires (Fig. 1, step 3 and 5) separated from each other by a strip of plastic bent width-wise in half (Fig. 1, step 4). Steps 1 to 9 represent the full process used in making the sensors. At step 9 we have a fully functioning bend sensor. By measuring the electronic resistance between the 2 wires when fully bent and fully stretched out, we can obtain its maximum and minimum resistance values, with which we can determine how far a sensor is being bent at any given time.

2.2 Voltage divider circuit

A voltage divider [2] is a circuit that divides the voltage from an input to different outputs. For the glove I have made a circuit board that divides the incoming 5 volts from the microcontroller equally for each bend sensor. In this way we can accurately determine each of the bend sensors values.

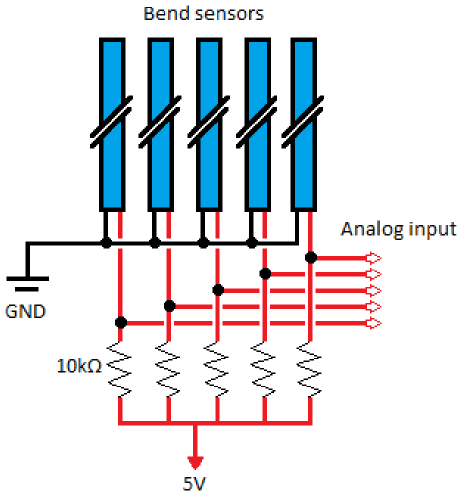


Fig. 2 – Voltage divider circuit

As seen in Fig. 2, 5 resistors of 10kΩ have been set in parallel to equally distribute the voltage across all bend sensors.

2.3 Inertial measurement unit

An inertial measurement unit, also called IMU, is an electronic device that responds to gravitational forces, orientation and velocity changes. It is typically used in unmanned aircraft. For this project, I have bought a Razor - 9 Degrees of Freedom IMU (Fig. 3) which has a 3 axes accelerometer, 3 axes gyroscope and a 3 axes magnetometer on it. By using its own firmware we can determine the gloves yaw, pitch and roll (Fig. 4).

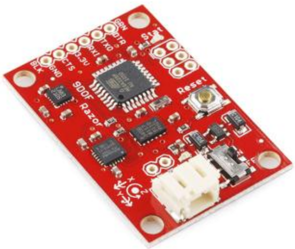


Fig. 3 – IMU [3]

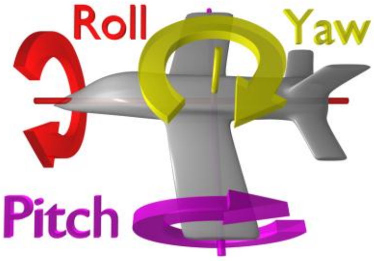


Fig. 4 – Yaw, pitch, roll [4]

I have placed this IMU on the back of the glove so that I can measure in which direction the hand is inclined. This way, by tilting the glove, I can move the mouse cursor on the computer the glove is connected to in the direction the hand is being tilted. The movement is achieved by the application I have written.

2.4 Arduino microcontroller

To measure all the values from the bend sensors and the IMU, I have to use a microcontroller that has the capability to read analogue inputs and is capable of serial communication with two different devices simultaneously, the IMU and a computer. I have used and Arduino Mega 2560 (Fig. 5) because it provides all the necessary features.

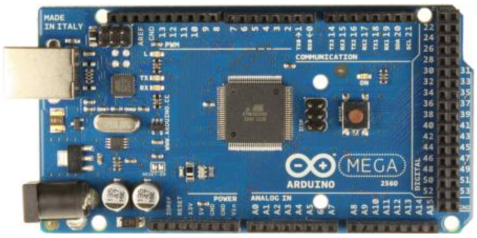


Fig. 5 – Arduino Mega 2560 [5]

This microcontroller is programed using a language similar to C/C++. The code I have programed the microcontroller with, reads the bend sensor values and stores them in a usable format. Afterwards they are sent together with the yaw, pitch and roll angles to the program on the computer.

2.5 Final assembly

Using all the components mentioned above, I have assembled a glove like device that measures the bending of each finger and the attitude of the hand (Fig. 6).

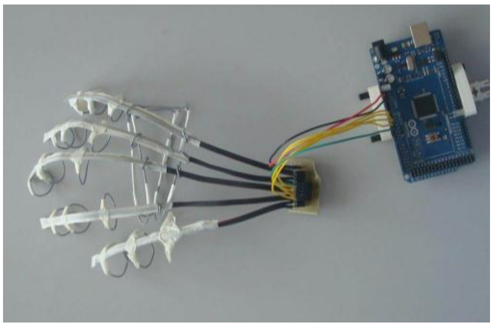


Fig. 6 – The glove (without the IMU)

3 The application

3.1 General

I have written a Java application which listens for incoming messages on a USB Port. These messages are sent by the microcontroller on the glove and consist of bend values for each finger and values of the yaw, pitch and roll angles.

These messages are interpreted by the program and sent into a stack of instructions where the program verifies if the last instruction can be executed on its own or if it needs to be in a combination of instructions. For example if I assign a 3-gesture sequence to the windows shortcut CTRL + ALT + DELETE, then, when the stack contains all 3 gestures representing CTRL, ALT and DELETE in that sequence, it takes them out of the stack and simulates these key press events.

The program can be set to different modes. These are standby, mouse, sign language and shortcuts. When the program is in mouse mode, it simulates mouse click and move events. In sign language mode, the program interprets the gestures made by the hand and simulates the key pressed event of the letter that the gesture represents. In shortcut mode, the program listens for a range of pre-set gesture and movement combinations and executes the events assigned to them.

All modes, except standby, can be simultaneously set. In the case that gestures overlap, the priority order is mouse - 1, shortcuts - 2, sign language - 3.

3.2 Bend sensor calibration

Due to the fact that I have made the bend sensors by hand, they generate a certain amount of noise and unusable values. The top 40% of the returned values are completely unusable. When a sensor is bent, it doesn’t return an exact value, but a range of values in the general vicinity of the correct bend value. To be able to obtain exactly how far the sensor has been bent, I have written a function (Fig. 8) that calibrates and trims the sensor values. The bend sensor array of the glove is represented in the microcontroller by the following structure (Fig. 7).

```
typedef struct{
    int pin;
    int max_v;
    int min_v;
    int val;
    int proc;
}flex_type;
flex_type flex[5];
```

Fig. 7 – Bend sensor structure

Each bend sensor has a pin thru which it is connected to the microcontroller and its own maximum and minimum values, as well as the currently read bend value and a calibrated-trimmed value (proc).

```
void read_flex(){
  for(int i=0;i<5;i++){
    flex[i].val = analogRead(flex[i].pin);
    if(flex[i].val > flex[i].max_v){
      flex[i].max_v = flex[i].val;
    }
    if(flex[i].val < flex[i].min_v){
      flex[i].min_v = flex[i].val;
    }
     flex[i].proc = map(flex[i].val, flex[i].min_v, flex[i].max_v, 100, 1);
     flex[i].proc = map(constrain(flex[i].proc, 40, 100), 40, 100, 0, 9);
  }
}
```

Fig. 8 – Sensor calibration function

I determine the minimum and maximum values returned by the sensors and map them to values between 100 and 1. After that, I cut of the top 40% of the range and map the remaining values between 0 and 9. This way I can have a final value representing the amount of bending: 0 – sensor stretched out completely; 9 – sensor bent completely.

4 Conclusions and future development

I conclude that even if the main interacting methods with a computer consist of mouse and keyboard, they can be combined into a single device. The glove presented in this paper provides an intuitive way to interact with a computer.

I intend to make the glove capable of wireless communication. In this way I do not need to be directly connected by wire to a device destined to be controlled.

The glove can be used, not only for human-computer interactions, but also for other gadgets and apparatus such as: simulating a TV remote control, remote manipulation of existing wireless devices, smart-house controllers.

My main future objective is to change the way that the glove simulates mouse movement events. Currently this movement is achieved by tilting the hand in the direction the cursor is desired to be moved. The best way to simulate the movement would be to determine the hands X, Y and Z coordinates relative to where the glove has been turned on and by measuring the distance the starting point and the point to which the hand has been displaced on the XoZ plane. This way the mouse could be moved in a much more intuitive manner.

The glove itself needs to be made to look more like a normal glove and the wires need to be hidden from view. This is a general design objective.

In the future, after all functionalities of the glove have been completely developed, I will change the Arduino Mega 2560 microcontroller with a much smaller chip. Currently I am using the Arduino because it is not yet clear how many serial communications, input and output pins I will use in the final version of the glove.

5 References

[1] http://en.wikipedia.org/wiki/Antistatic_bag

[2] http://en.wikipedia.org/wiki/Voltage_divider

[3] https://www.sparkfun.com/products/10736
[4] http://en.wikipedia.org/wiki/Inertial_measurement_unit
[5] http://arduino.cc/en/Main/ArduinoBoardMega2560

Adrian POPA
University of Lucian Blaga
Department of Science
Sibiu, Str. Dr. Ion Ratiu nr.5-7
Romania
E-mail: ad_robotics@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Gesture based smart human-computer interaction

Dorin Adrian Rusu, Daniel Grigore Rosca, Alin Rosu
Teacher Coordinator: Adrian Gligor

Abstract

As computers evolved towards a new age, so did the way humans interacted with it. So far this evolution was mostly centered on the software and the way it interprets the same peripherals, insuring better interpretation of the end results. Technological progression was always oriented toward achieving a “natural” interaction, meaning methods that human being consider natural, suggestive or ones that require little to none adaption time.

This paper addresses the modern approach toward controlling and interacting with computers by the means of gestures, postures and human movement as nothing is considered more natural, than human movement. Our work describes the current problems in this field, current technology that surpasses them and a practical application as a result of our study and research.

We aim to create viable and practical software that can be used worldwide by a big number of people, with extreme ease of use and minimal to no training required.

1 Introduction

Gesture recognition is a domain in computer science and engineering, oriented on detecting and interpreting human gestures, achieved with mathematical and data mining algorithms that analyse the data usually originating from video cameras. Since image processing existed long before, many of the specific algorithms can be used for gestures, since a video stream is a sequence of images.

Gesture recognition technology is a fairly new domain, being researched by institutions for the first years of its existence, but receiving a wider attention since it was introduced in gaming and mobile device.

The first types of gestures were interpreted and recognized by touch screens and gyroscopic sensor, identifying movements of the hand in three dimensions, the speed of the movement, the act of shaking the device and others.

More advance gyroscopic gesture detection was implemented in Nintendo’s gaming platform Wii, where different actions in games could be realised by the movements that were induced in the console [1].

Gesture recognition obtained a wider attention with the introduction of Project Natal, or Kinect as it came to be named, a module associated with Microsoft Xbox [2]. It has shown that real-time interpretation was possible and opened a new door for gaming development.

Currently extensive research is being conducted this topic towards developing smarter but easier interfaces, helping people with disabilities, education, virtual interaction with environments, eye tracking, facial recognition and emotion detection [3], [4], [5].

2 Overview of the approach

2.1 The scope of the application

The application is oriented towards achieving a practical and useful product that can make use of the current technology of human gesture recognition. A secondary scope would be the desire of including a high level of user-friendliness, insuring ease of use, fast adaptability and suggestive control.

Our application has applicability in the entertainment industry, clothing industry or many household applications.

Let consider the case of the clothing industry, where a big percentage of this industry is centred on human interaction with the clothes they buy and wear, therefor human gesture and posture recognition is a technological domain with big potential.

2.2 Technology used

The technological choice for human gesture recognition is the Microsoft Kinect module, part of the Xbox gaming platform. It is a device specialized towards human interaction, having a SDK well integrated and documented with Visual Studio, and the C# programming language.

2.3 Software structure

The application itself is an interface that shows the output of one of the video cameras on the Kinect on which different types of clothing are applied when a human being is detected. The clothes follow human movements and are scaled according to each individual person. As an aid to the potential buyer, an algorithm that detects the height was implemented

2.4 Software functionality

Although as a device Kinect is very sophisticated, Microsoft has developed a very well documented SDK that implement a high level of transparency between the programmer and the functions of the Kinect.

The application begins with the sensors initialization procedure, where the two video cameras, the infrared sensor and two microphones are started. Afterwards the sensor specific algorithms are initialized to required settings.

The most relevant to our application is the skeleton tracking which overlays a 20 point skeleton over the detected human body. Also the speech recognition algorithm was used to implement a more natural control over the functions.

To change between different types of clothing articles, a number of gestures are being detected that insure this. For a natural and comfortable feel all the gesture are hand related, and are identified with the X and Y axis of each point by applying trigonometry and analytical geometry algorithms.

An extra feature is the height calculator algorithm which using trigonometry algorithms calculate the distance between a series of point on the skeleton that make up about 95% of the full height of the detected human body.

3 Comment and experimental results

Experimental results indicate that the application works at its 100% in a good ambient light environment increasing the response rate of the application.

Since the models used in our application are 2D, we have a restriction for scaling; the recommended is reported at approximately 1.5 m and 3 m in front of the device. The experiments have responded with better performance rates at 2.5 m distance from the sensor.

Furthermore, we have a similar restriction when it comes to detecting the person's height and cloth dimensions. The person in front of the device has to be at least 1.40 m tall so that the position of the clothing is accurate and the detection is approximately correct. With this restriction in vision, it is clear that the software reacts faulty around children.

Other experimental results indicate that movement gestures have to be executed in a period of time, fast gesture movements or incorrect gesture movements could result in slower response rate or no response at all. Each and every movement has to be executed in a perfect line, horizontally or vertically, so that the result can be registered.

An issue which must be resolved is the random change between detected human bodies when more than one person is standing in the Kinect's view area. Kinect chooses the one it considers, shifting the clothes to that person. Issues have been met where although one person is testing the clothes; Kinect recognises the gestures of the other one, therefor only one person must use the software at any given moment.

Also to be noted, the height calculating algorithm functions with a 90-95% efficiency only when the person is standing with the back straight, legs together and head at a straight line with the back. When these conations are not met the efficiency reduces drastically.

4 Conclusions

We have implemented a user interface application based on gestures that can act as a virtual store at home. The interface is relatively satisfactory and practical for many uses.

The application has been created using the Kinect Sdk, we chose this device because of its advantages: Kinect is easy to use, our movements become the movements in our application, it is entertaining and an ultimate source of fun. Like any other device or technology it has its disadvantages like hand mapping, face recognition, catching all the body movements without limitations.

Other improvements in our project would place the virtual store in xbox live and integration with social networks so every person could share their experience and enjoy every moment when using it.

Finally, another approach would also make it possible to implement a physics engine so that the person in front of the device could see and live a real shopping experience.

We also hope of implementing a higly accurate methode of detecting the size of the clothes the possible clinent should buy. The first steps were taken with the height calculator algorithm, but we intend to include a waist measure calculator and other determine other measure that must be taken into consideration when chosing the size to wear.

References

[1] R. Colin Johnson, *3-D Gesture Control Breaks Out of the Game Box*, https://www.eetimes.com/electronics-news/4212294/3-D-gesture-control-breaks-out-of-the-game-box-semiconductor, 1/19/2011, retrieved April, 2013.

[2] Jarrett Webb, James Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*, Apress, ∨ Full Description ISBN13: 978-1-4302-4104-1, 2012, pp. 3-5.

[3] Miguel Sales Dias, Sylvie Gibet, Marcelo M.Wanderley, *Gesture-Based Human-Computer interaction and Simulation*, 7th International Gesture Workshop, Lisbon, Portugal, 2007.

[4] Matthias Rehm, Nikolaus Bee, Elisabeth André, *Wave Like an Egyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions*, British Computer Society, 2007, http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/199/wave_like_an_egyptian_final.pdf.

[5] S. Mitra, T. Acharya, *Gesture Recognition: A Survey*, *Systems, Man, and Cybernetics*, Part C: Applications and Reviews, vol.37, Issue 3, pages 311 – 324, May 2007

[6] Oliver Pauly, Loren Schwarz, Diana Mateus, *Introduction Machine Learning Kinect Lab Course*, Computer Aided Medical Procedures, Technische Universitat Munchen

[7] Zhengyou Zhank, *Microsoft Kinect Sensor and Its Effect*, MultiMedia, IEEE, vol.19, Issue 2, pages 4-10, February 2012

[8] K. Khoshelham, *Accuracy analysis of Kinect Depth Data*, International Archives of the Pthotogrammetry Remote Sensing and Spacial Information Sciences, August 2011

RUSU Dorin Adrian
“Petru Maior University”
Computer Engineering
Str.Nicolae Iorga, nr.1
Romania
E-mail: rusu.dorin.adrian@stud.upm.ro

ROSCA Daniel Grigore
“Petru Maior University”
Computer Engineering
Str.Nicolae Iorga, nr.1
Romania
E-mail: rosca.daniel.grigore@stud.upm.ro

ROSU Alin
“Petru Maior University”
Computer Engineering
Str.Nicolae Iorga, nr.1
Romania
E-mail: rosu.alin@stud.upm.ro

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

uTorrent Backup

Dragoș Sebeștin
Teacher Coordinator: Cătălin Stoean

Abstract

This software is intended to simplify the process of backing up uTorrent settings in case of a system reinstall is needed and intended. Most users do not know how to do this manually or simply forget it, so there is a need for such software because uTorrent lacks native support. Backing up the settings is useful in case the user wants to continue seeding files on his computer or if the download of a torrent file is not finished, so it may resume after the reinstall of the operating system.

This application is written in Visual Studio 2010 using the C++ language, starting from a win32 template project, and using UNICODE character set.

1 Introduction

BitTorrent clients represent peer-to-peer file sharing programs based on the BitTorrent protocol.
A **peer-to-peer** (**P2P**) computer network can act as a client or server for the other computers in the network and allow shared access to selected files without the existence of a central server [5]. Each such P2P network that is set up over Internet requires all computers in the network to use the same or a compatible program to connect to each other and access files that are shared on the involved computers. P2P networks are usually used for sharing any type of digital format content such as audio, video or data. They have become one of the most used ways for people to share files. Peer-to-peer file sharing strongly depends on the number of peers which are sharing a certain file. The more peers (participants in the application that share files) there are, the greater the download speed of another peer that downloads that multi-shared file. Thus, it is important that many users keep seeding torrents with files they intend to share for as long as possible.
uTorrent is a popular BitTorrent client and it is the one on which this work is focused on. Settings files are located in a folder in %appdata% location on the windows partition. Data is stored using bEncode algorithm [3].
The development of such a backup solution intends to answer to the following necessities:

- avoiding the manual re-download of the individual torrents
- eliminating the time consuming hash check process which uTorrent applies to make sure the files are downloaded and are not corrupted.
- on some tracker type sites, seeding torrents is important for keeping a high upload to download ratio.

Backing up requires copying the settings files and all of the .torrent files. But uTorrent offers an option to store .torrent files on another folder chosen by the user. So in order to copy them a process of decoding the settings files occurs and the path to that particular folder is thus obtained.
The restore process is opposite to the backup one. The software must be able to successfully put back the files in uTorrent's folders. So if the user chose to store the files on some other folder rather than the default one, the application must remember that path and restore the files accordingly.
uTorrent has an option to delete only the data from the computer and keep the .torrent file. Inadequate use of this feature can lead to useless disk space usage by keeping those residual files. This problem requires a cleaning tool. The proposed software offers a cleaning solution by decoding the settings files, making a list of active torrents and then comparing them to the contents of the folder and deleting files which are not on the list.

2 uTorrent Backup application

2.1 The application structure

The application features a simplistic, easy-to-use user interface. The user if prompted to enter a backup folder path and then choose the actions: backup or restore. The UI also features commands to open the root folder, or the backup folder and a button for the clean function.

2.1.1 Backup option

The backup process is fairly easy. The user has to select a path to a backup folder using the UI and then the backup button needs to be pressed. The application then proceeds to copying the settings files along with the .torrent files to that folder. A progress bar shows when the operation is completed and the users is notified if there are any errors.

2.1.2 Restore option

The restore process is opposite to the backup one. The user is prompted for the backup folder location and then the restore button has to be clicked. Again, the user is notified if the operation completes successfully or if there are any errors.

2.1.3 Clean option

The clean option is one of the most powerful features of this application. It works by decoding the "resume.dat" file which contains a list of all the currently active torrents. Being part of uTorrent's settings files, this one is also organized using the bEncode algorithm. From here, a list with all the active torrents is obtained. Now the application processes every .torrent file in the settings folder by checking if it is part of the list and, if not, the file gets deleted.

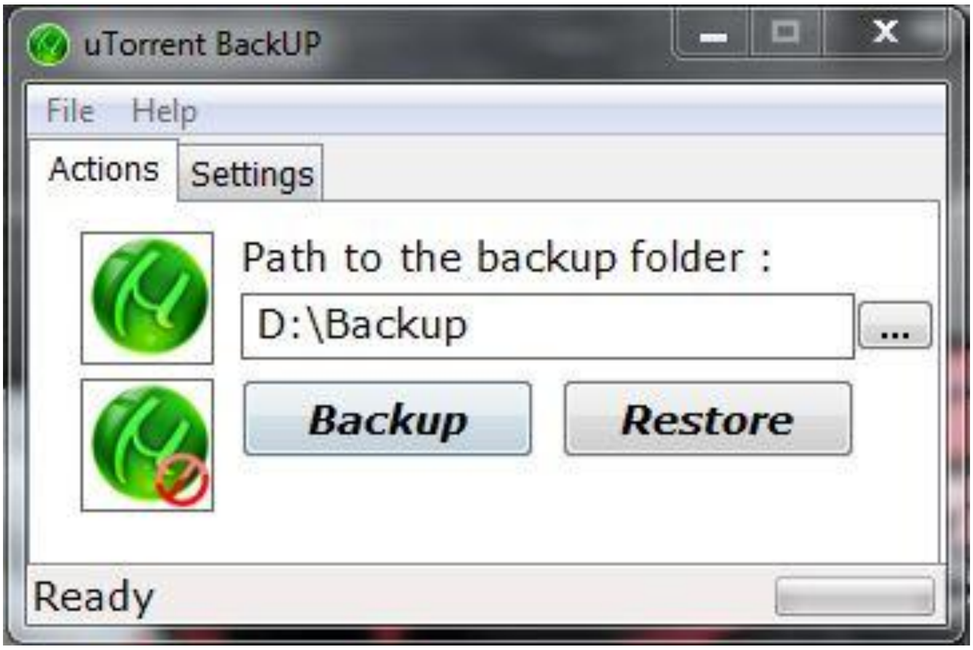
When the process is completed the users is notified regarding the number of files that were cleaned.

2.2 The User Interface

The interface (Fig. 1 and 2) was built using Visual Studio, starting from a win32 application template [2] and contains the following components [1]:

- Edit box used for selecting the backup folder path
- Buttons and bitmap buttons used for various user actions

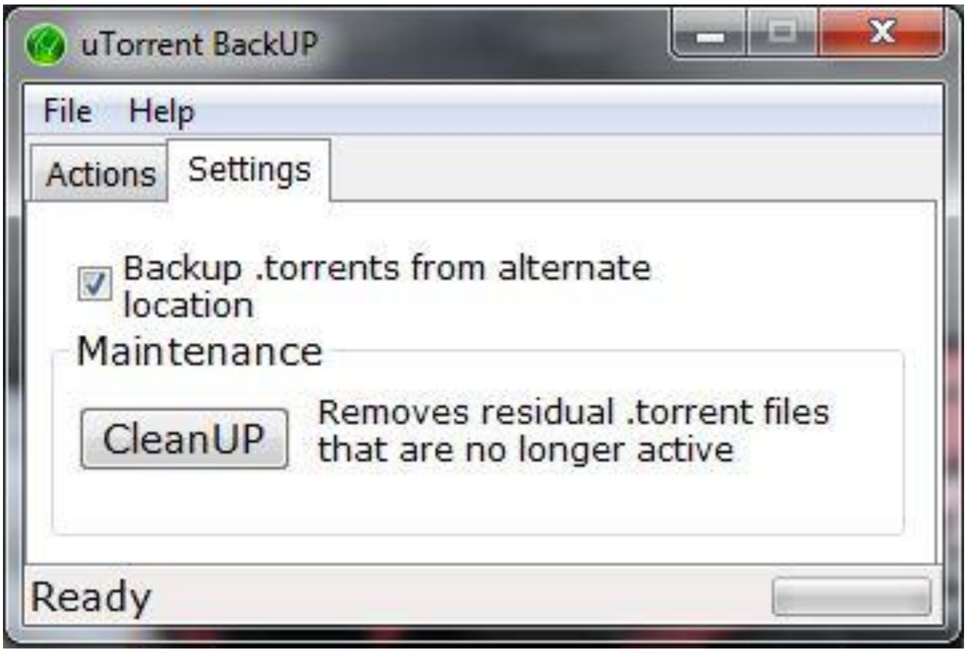
- Tab Control used for creating 2 sections: actions and settings
- Status bar used for displaying success or error messages
- Checkbox used for opting for whether the alternate folder should be included
- GroupBox used for aesthetics



Path to the backup folder : D:\Backup

Ready

Fig. 1. The path to the folder to be backed up or restored can also be inserted via navigation through folders. The status bar also contains a progress bar.



☑ Backup .torrents from alternate location

Removes residual .torrent files that are no longer active

Ready

Fig. 2. The checkbox gives the user the option to include the alternate folder in the backup process.

3 Program Code

The backup and restore functions use the `SHFileOperation` function to copy all of the files by initially finding the first file and consequently, all of the others:

```
SHFILEOPSTRUCT sf;
memset(&sf,0,sizeof(sf));
sf.hwnd = NULL;
sf.wFunc = FO_COPY;
sf.pFrom = (LPCTSTR)from;
sf.pTo = to;
sf.fFlags =  FOF_SIMPLEPROGRESS | FOF_NOCONFIRMMKDIR;
sf.lpszProgressTitle = _T("Copying Files ...");
nret = SHFileOperation(&sf)
```

The cleanup function loads the “resume.dat” file into memory and decodes its contents. It creates a simple linked list and it creates a new node for every .torrent entry.

```
struct torrentList{
    LPTSTR name;
    bool visited;
    torrentList* next;
};


while(q=decodeText(data+index, index)){

    if(_tcscmp(q, _T("rec")) == 0)
        break;
    index += _tcslen(q);

    newItem = new torrentList;
    newItem->name = q;
    newItem->visited = 0;
    newItem->next = list->next;
    list->next = newItem;

    q = findSubStr(data+index, _T("webseeds"), fileSize);
    q+=11;
    index += q-(data+index);

    torrentCount++;

}
```

Subsequently, it checks the content of the settings folder and iterates through the list to find out if the current file is active or not. If it is not on the list, it gets deleted.

```
do{
    count = 1;
    while(itemCrt != NULL){
        count++;
        if(itemCrt->visited == false){
            if(_tcscmp(itemCrt->name, wd.cFileName) == 0){
                itemCrt->visited = true;
                deleteFile = false;
                break;
            }
        }
        itemCrt = itemCrt->next;
```

```
                }
                if(deleteFile){
                        torrentName = (LPTSTR)malloc((len + _tcslen(wd.cFileName)) * sizeof(LPTSTR));
                        ZeroMemory(torrentName, (len + _tcslen(wd.cFileName)) * sizeof(LPTSTR));
                        _tcscpy_s(torrentName, len + 1, rootFolder);
                        torrentName[len] = '\\';
                        _tcscat_s(torrentName, len + _tcslen(wd.cFileName) + 2, wd.cFileName);
                        DeleteFile(torrentName);
                        deletedFiles++;
                }
                deleteFile = true;
                itemCrt = list;
        }while(FindNextFile(hFile, &wd));
```

4 Conclusion and further developments

The software comes in aid to users who intend to backup their data in case they need to reinstall their operating system. Since uTorrent lacks native support for this issue, there was a great need for such software. Therefore, since the development of the software and its upload on code.google.com [4], it has been highly appreciated and has been downloaded by over 1500 users.

In the future there is already undergoing personal work for adding the option to backup the files online through file hosting solution so users who use multiple computers could share their files on all of the devices and also be able to sync them.

References

[1] Bruce Eckel, *Thinking in c++ 2nd Edition*, Free Electronic Book, 2003.

[2] theForger's Win32 API Programming Tutorial, http://www.winprog.org/tutorial/.

[3] bEncode algorithm, http://en.wikipedia.org/wiki/Bencode.

[4] Project website, https://code.google.com/p/utorrent-backup/

[5] Rüdiger Schollmeier, A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, *Proceedings of the First International Conference on Peer-to-Peer Computing*, IEEE (2002).

Dragoş SEBEŞTIN
University of Craiova
Department of Informatics
A.I. Cuza nr. 13, CRAIOVA
ROMANIA
E-mail: sebestin.dragos@gmail.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Aida Studio - IDE for rapid application development and universal compiler

Mihai Stancu
Teacher Coordinator: Prof. PhD. Dana Simian

Abstract

The aim of this paper is to explore the requirements and principles for creating rich user interface design tools. To achieve this goal, we will evaluate existing approaches and implementations and based on this analysis, we will design, develop and test an abstracted framework for creating user interface design tools for the .NET platform.

1 Introduction

The purpose of this paper is to understand and explore the requirements and principles of creating tools for developing rich graphical user interfaces.

In this paper we present the design and implementation of an architecture that allows building user interfaces for .NET applications. The implementation is compatible with Microsoft. .NET, but one of our goals is for the application to be included in the Novell sponsored Mono proj ect, which provides an open-source implementation of the platform.

1.1 Toolkits and designing GUIs

Graphical user interfaces (GUI - Graphical User Interfaces) consist of a set of controls arranged in a container. The controls may be buttons, text boxes, menus, etc. The container may be a window, but it is often referred to as a form or web page.

Developers can create interfaces programmatically, directly accessing system functionality provided by the operating system graphics. In the case of Microsoft Windows, it is integrated in the core libraries and in the case of UNIX platforms, it is provided by the X Window System. In both cases, developers need to write low-level code, which makes the source code extremely long and difficult to maintain and extend, because of the complexity.

A tool for creating a graphical interface allows users to create GUIs in a visual way. Usually it provides a toolbox with a predefined set of components and a drawing area where the components may be positioned. A table of properties allows developers to modify or customize the characteristics of objects, such as size or location. The tool should also allow saving the designs and then resuming editing.

In addition, a good design tool should provide different options dynamically depending on the selected components, and automatic alignment of controls in a nonintrusive way. Another requirement is to expose an interface to customize the behavior of components during design time.

Below are some details of the platform. NET that make it ideal for implementing this type of application, and in the next section we describe the design and implementation of Aida Studio.

2 The .NET platform

.NET is a development platform that focuses on rapid application development and independence of the systems used.

All .NET programs are compiled to CIL byte code (Common Intermediate Language). The resulted code is called managed code. Programs are executed in a virtual machine - CLR (Common Language Runtime). Compiled code is then stored in executable files (. exe) or libraries (. dll).

The dominant .NET language is C #, but there are also many other languages that rely on CIL such as Visual Basic .NET, F #, IronPython, IronRuby, etc

One of the largest and most important components of the .NET platform is the class library - a huge collection of reusable object-oriented types. An overview of its components can be seen in the following figure:

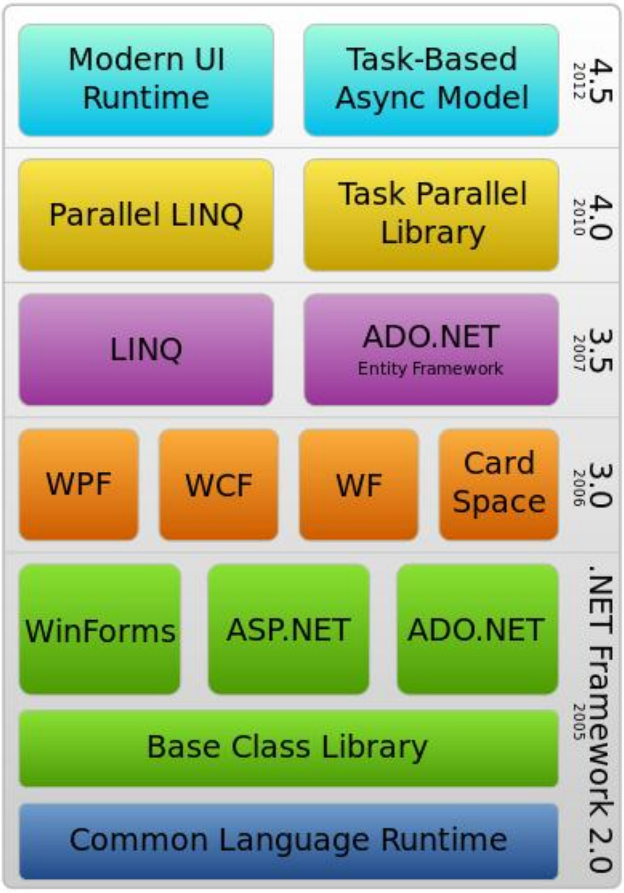


Fig. 1: .NET Framework components

There are two major implementations of the CLR: the one developed by Microsoft and the Mono project sponsored by Novell. The second is operating system independent and can run on Linux, FreeBSD and Apple OS X, unlike the first one, which runs only on Windows.

2.1 Metadata

Unlike other platforms and languages .NET brings the concept of metadata. In .NET the compiler automatically generates metadata during compilation and stores it in the file that contains the CIL code. Unlike RTTI in C++, CIL metadata contains a complete description of the assembly:

- Method names, variables returned type, the type parameters

- Property names and their types
- Names of the fields and their types
- Type inheritance chains, etc.

This allows a range of extremely powerful reflective programming techniques, allowing not only full description of the types, but also invocation of methods, dynamic instantiation of objects at runtime, and more.

In. NET, once an assembly is loaded into memory all the information on the types contained is also loaded and made available for reflection and inspection.

If a design tool must find all controls of a certain type and its derivatives, such as "Button", it can search all existing types in the loaded assemblies for classes derived from "Button". This is an extremely powerful technique that allows the use of arbitrary assembly components at runtime without the design tool having any prior information about those files. This is ideal for using custom components in designer. Graphical user interfaces (GUI - Graphical User Interfaces) consist of a set of controls arranged

2.2 Assemblies

Developers can add custom metadata code via attributes that basically sets a flag for the compiler to generate the respective metadata. An attribute is a class that inherits a special "Attribute" class that can be used with any method, property, class or the entire assembly.

Attributes are used even within the .NET platform. For example, ASP.NET uses special attributes to expose certain methods as web services. A forms designer could use special attributes to filter components that should be available at any given moment.

One of the many uses of attributes is the association of a particular type of data with a converter. For example, the "TypeConvertor" class in. NET is the class that can be inherited by possible converters of data types.

An important aspect regarding the transformation of types in .NET is type based editors. These editors can get an object of a certain type, and allow viewing and editing it. An example of such an editor is "Color picker" for the "Color" property. Editors based on data types are derived from the "UITypeEdit" class and are associated with a particular attribute. A table of properties can use this attribute to allow editing of certain properties of the selected object.

2.3 Attributes

The .NET platform contains classes that allow direct compilation of a piece of text that represents the source code written in a particular language. The reverse process of decompiling the assembly files is not as simple.

Decompiling is the process of converting an executable file containing binary code for a physical machine to understandable, high-level source code. Usually resulting from the compilation are the instructions to be executed by the CPU. Another possibility is to compile with a "virtual machine" which then either interprets each instruction one by one, or compiles the instructions to machine code through a process called JIT compiling (Just In Time). One of the earliest examples of this type of compilation is Machine-Cod-O developed by Martin Richards in the late 1960s. The best example of virtual machine in the present is JVM (Java Virtual Machine), created by Sun Microsystems, which used to run code written in the Java language.

When a program is compiled, it becomes extremely difficult for someone to understand the binary code. Compiling almost always results in loss of information such as the names of local variables and explicit identification of control structures. Also, many compilers provide code optimization, reducing similarity to the source page.

This is why compilation is a many-to-one process, in which different source code files can produce the same binary representation. A decompiler must reverse the process: source code must be chosen that produces results that are closest to the binary.

The .NET Platforms was created by Microsoft to directly compete with the Sun JVM product. It is very similar to the JVM, it comes with a full set of libraries, is platform independent and it is JIT compiling virtual machine on an intermediate code to improve performance. Optimization occurs in this stage, and not at the moment when high level code is compiled to intermediate language. This means that in standard binary format, .NET code is not optimized, so the code structure is very similar to the source from which it was compiled.

To decompile assembly files we used the source code of a number of open source projects such as Mono, SharpDevelop and ILSpy, adapted to our application requirements and optimized for maximum efficiency.

3 Aida Studio architecture

The application allows developers to create rich interfaces with minimal effort, and with high speed. Extensibility and modularity were two key elements that have been considered throughout all stages of development.

The .NET Platform has a service-oriented architecture, abstracted, extremely rich and complex, which allows easy connection of various components, containers and services.

The components have associated design-time behavior independently of the tool used to edit them. There is a list of design-time services that can communicate with different components, designers and among themselves. The DesignSurface component is composed of these services, also including the designers and edited components.

This architecture uses metadata attributes to describe the behavior of components and their members. This makes it possible to dynamically add, delete and modify metadata components.

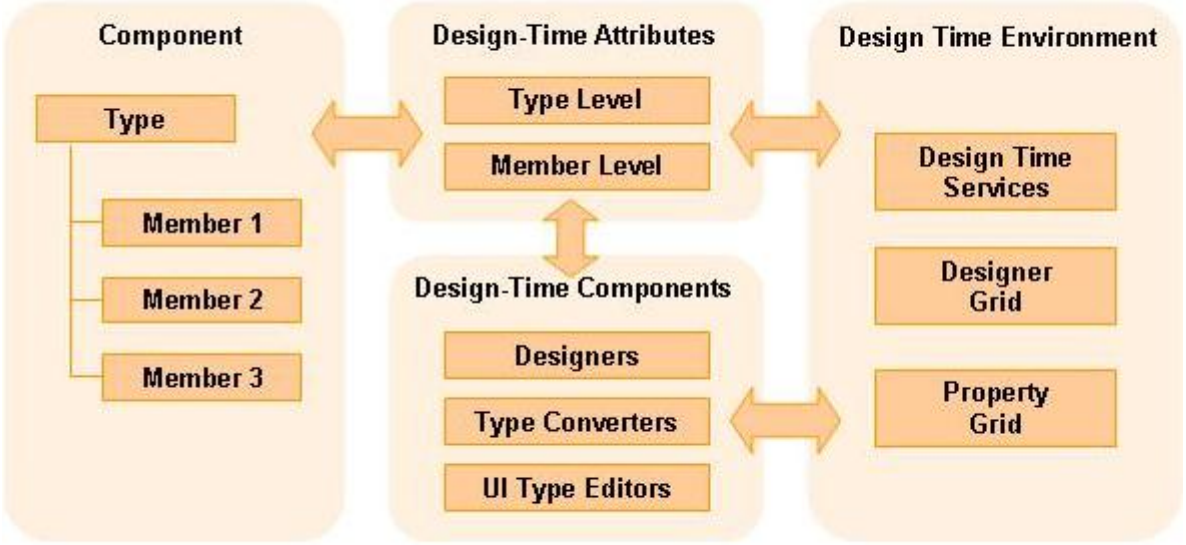


Fig. 2: Design-Time architecture in .NET

3.1 Components, Containers, Sites

The IComponent interface represents the core of this architecture. Any item that can implement this interface is called a component. Components have three important characteristics:

1. They can be contained / hosted by a container;
2. They can launch the service request through the Site;
3. They can be assigned a name.

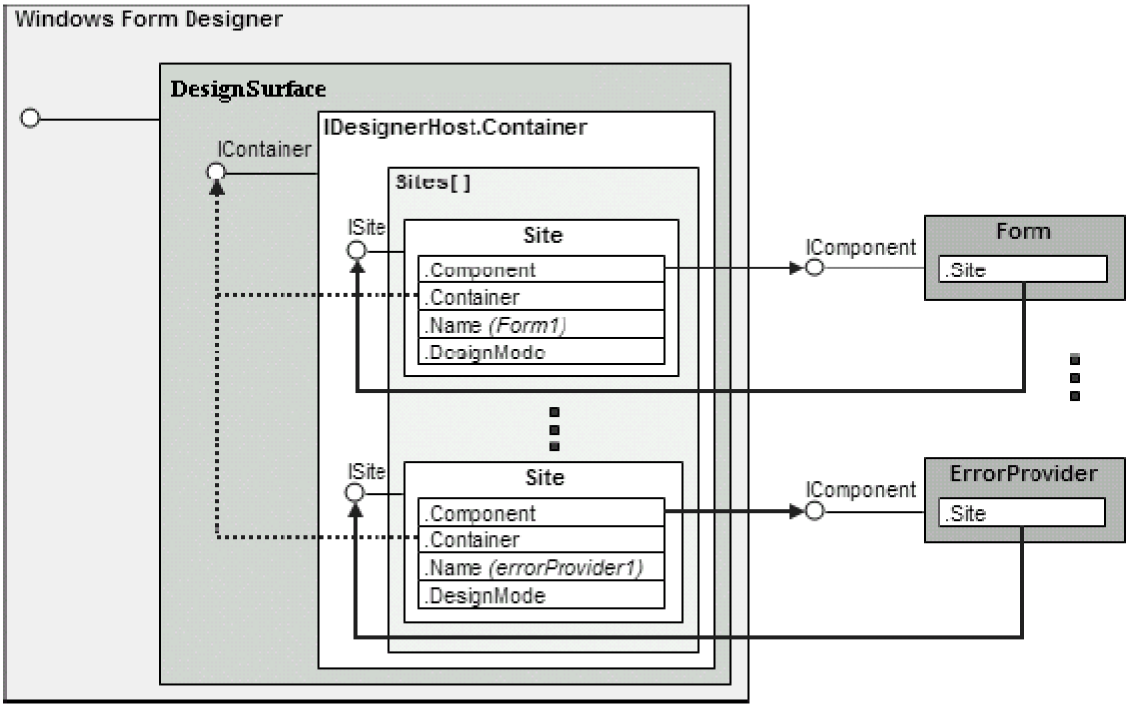


Figure 3: Hosting a DesignSurface

3.2 Services

A service is an instance of an object that implements a particular set of features. Services are stored in service containers and can be retrieved from providers such as the component Site or generally via IServiceProvider interface implementations.

The real power lies in how free they are from depending upon each other; an application publishes its interface or base class that defines a service but does not publish the class that implements the service. This development pattern allows components to seek and obtain a specific service without knowing the details of implementation, or where that service comes from.

3.3 Design-Time Attributes

Design-Time Attributes of a particular data type or of a member of a class extends the design-time functionality without creating a dependency for that data type or class. The implementation uses the following attributes:

Attribute	Description
DesignerAttribute	Associates a data type with a designer
TypeConverterAttribute	Associates a data type with a type convertor
EditorAttribute	Associates a data type with an image editor
DesignerSerializerAttribute	Associates a data type with a serializer

Table 1: Design-Time attributes

3.4 Designers

Designers are classes that are associated with a particular component type through the DesignerAttribute and which implement the IDesigner interface. The designers are initialized with an instance of the component that they manage.

The root component, such as the Windows Forms form or a web page must implement the IRootDesigner interface which is special in that it must provide a visual representation of the component – a View.

Designers are responsible for initializing the component in design mode, changing and extending behavior and appearance, adjusting attributes, events and properties exposed, adding menu items to the context menu of a component, etc.

3.5 Design-Time Services

The DesignSurface component contains a service container for the other components that populate it with the specialized implementations of the following Design-Time services:

Service	Description
INameCreationService	Generate and validate component names
ISelectionService	Defines the interface for component selection
IMenuCommandService	Allows designers to add new actions to the context menu of specific components when they are being edited
IServiceContainer	The container which holds the services provided by other objects
IComponentChangeService	Warns of adding/deleting/changing components

Table 2: The services implemented by the DesignSurface component

Other crucial services are provided by our implementation of a Toolbox that allows adding components located in the edit form and the PropertyGrid which allows you to change the properties of components selected at a specific time.

By discovering service types, the Toolbox is populated with all the available components, both in GAC (Global Assembly Cache) and local folder. The items shown in the Toolbox are filtered based on attributes that have been provided. Moreover, when the application detects at runtime a change in the structure of its files, it scans the assembly sites for new locations to detect new types of components that can be added to the Toolbox.

Last but not least, another very important service is the Undo / Redo engine, which records any changes (or series of changes inside a transaction) to be possibly canceled by the user. For this, we must be able to serialize all possible actions that are allowed in the designer, and for this we used the Memento design pattern.

3.6 Design-Time Serialization

Design-Time serialization is the process of converting an object graph into source code (C#, VB.NET, XML, etc.) which then can be read to recreate the graph of objects. The DesignSurface component has an associated DesignerLoader which performs loading of the form to be edited.

When creating a new file is generated, using CodeDom, the main body of the form is generated (including references, namespace declaration, the declaring class and its members, together with the InitializeComponent method containing class member initialization). It produces a CodeCompileUnit object which will then be modified by DesignSurface via user actions. At the end of editing, this object is compiled into an assembly that can be directly referenced in other .NET applications.

If the user wants to launch the editing of a form contained in a given assembly, using the ILSpy library, the assembly is decompiled into C# source code, and then using NRefactory, part of the Mono package, the source code is parsed (more specifically, only the InitializeComponent method) to generate a CodeCompileUnit object. At this point, the DesignSurface component is ready to display its content.

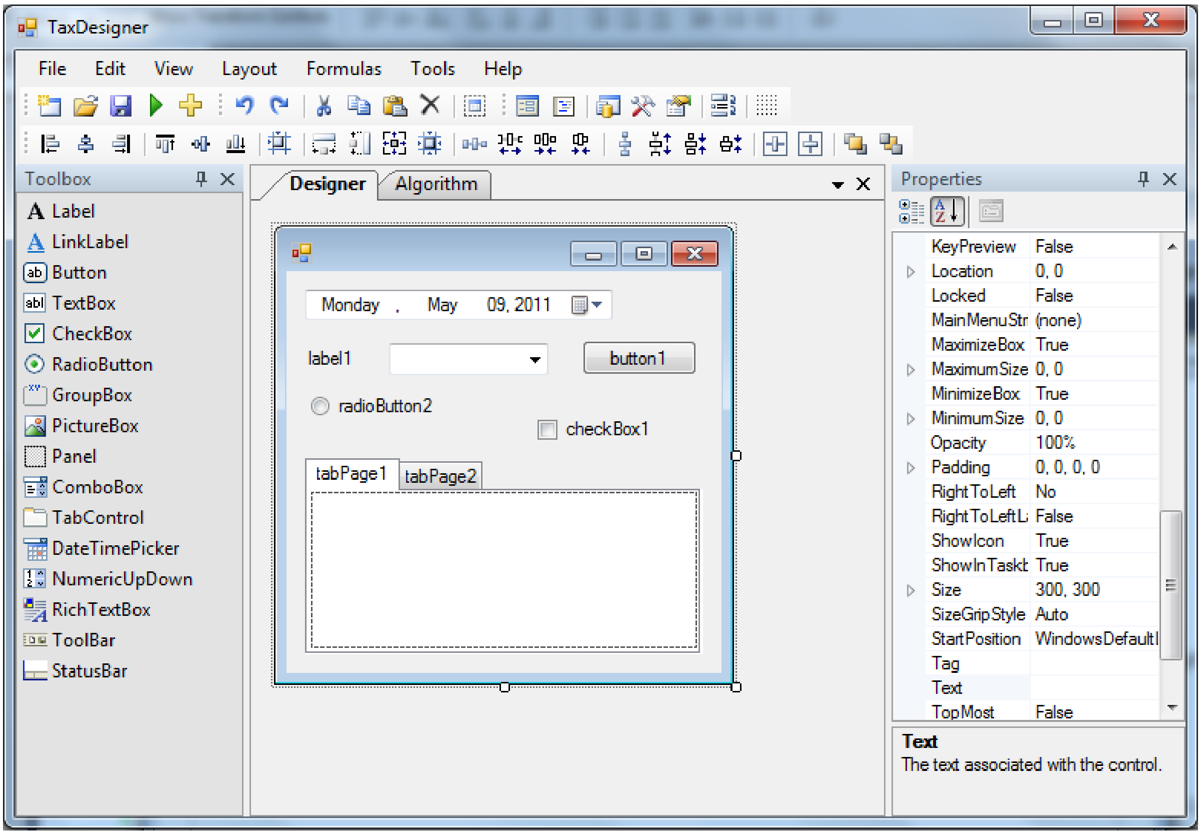
4 Plans for future development

Numerous efforts have been made towards increasing application performance and user friendliness, but there are many improvements that can still be made, which are planned for future versions. Here we mention only a few:

- Adding a designer component for Windows Presentation Foundation (WPF)
- Export output to other formats, not just assemblies (especially websites)
- Creating new components, customized to user specifications
- Porting to other platforms
- Importing different data sources (xml, xls, csv, etc.).
- Connecting to different database servers (SQL Server, MySQL, Oracle, etc.)
- Introducing modern interface elements (specifically the Microsoft Office Ribbon)

5 Conclusion

In the final two sections of this paper we present our conclusions regarding the application, a critical evaluation of the results, and the ways that the application can be extended in the future.



KeyPreview	False
Location	0, 0
Locked	False
MainMenuStr	(none)
MaximizeBox	True
MaximumSize	0, 0
MinimizeBox	True
MinimumSize	0, 0
Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftL	False
ShowIcon	True
ShowInTaskb	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultL
Tag	
Text	
TopMost	False

Text
The text associated with the control.

Figure 4: Aida Studio

5.1 Challenges

The implementation was intended to be elegant and efficient, but due to the complexity of the application, its development did encounter a series of difficulties.

The first challenge we met was the extremely poor documentation provided by Microsoft for the classes required to implement the designer.

The open-source projects such as Mono and ILSpy contained bugs which needed fixing before using the available source code.

Design-Time serialization proved to be difficult to implement as well. In particular, serialization of nested data structures posed serious problems.

Implementing transactions and assembly files decompilation were other issues for which we have allocated a considerable amount of time, especially for optimization.

5.2 Results

The project has successfully achieved the goals we have set, by designing and implementing a platform for creating visual interfaces with .NET technology. Written in C #, it represented a perfect way to improve knowledge about the language, by using the most "exotic" elements of it.

A key issue in developing the application was the creation of automated test suites, which significantly reduced development time and increased reliability results.

Last but not least, the application is very easy to use and the resulting assembly files can be added as a reference to any .NET project and can be seamlessly integrated into any .NET application.

References

[1] Tim Dawson, Hosting Windows Forms Designers http://www.divil.co.uk/net/articles/designers/hosting.asp, 2003

[2] Christian Holm, Mike Kruger, Bernhard Squida, Dissecting a C# Application, Apress, 2004

[3] Mono, Novell, http://www.mono-project.com/WinForms_Designer, 2011

[4] Stephen Horne, Decompilation of .NET Bytecode, Dissertation, Trinity Hall, 2004

[5] ILSpy, SharpDevelop Team, http://wiki.sharpdevelop.net/ilspy.ashx, 2011

[6] SharpDevelop, SharpDevelop Team, http://www.icsharpcode.net/OpenSource/SD/Default.aspx, 2011

[7] Shawn Burke, Writing Custom Designers for .NET Components, http://msdn.microsoft.com/en-us/library/ms973820, 2001

[8] Shawn Burke, Customizing Code Generation in the .NET Framework Visual Designers, http://msdn.microsoft.com/en-us/library/ms973818, 2002

[9] Shawn Burke, Make Your Components Really RAD with VS.NET Property Browser, http://msdn.microsoft.com/en-us/library/Aa302334, 2002

[10] Sayed Hashimi, Tailor Your Application by Building a Custom Forms Designer with .NET, MSDN Magazine, 2004

[11] Dinesh Chandnani, Create And Host Custom Designers With The .NET Framework 2.0, MSDN Magazine, 2006

Mihai STANCU
“Lucian Blaga” University of Sibiu
Computer Science Department
10, Victoriei Bd., Sibiu, 550024
Romania
E-mail: stancu_meehigh@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Breaking devices barriers by creating a continuous Windows 8 application using Modern UI and Windows Azure

Simona – Mihaela Tecșa
Teacher Coordinator: Daniel Hunyadi

Abstract

We are living in a digital world where people are used to do the same things on different devices (laptop, tablet, smartphone). Breaking devices barriers and creating a continuous user experience takes applications to a higher level, a level where you can use the same application anywhere, anytime, having access to the same data.

The paper presents the Modern UI principles in a Windows 8 application where users can create custom albums using different templates (templates are illustrating different events, e.g. first year of a baby life, and they can be customized by adding all the important details and captured moments following the timeline of the template, resulting a nice digital story), image effects and collage pictures. This application stores its data in Window Azure and gives users the possibility to access their albums and pictures from multiple devices, using authentication for restricting access to their cloud data.

1. Introduction

A common problem when using applications today is that a broken workflow occurs when you move from one device to another. For example as we switch among devices in a typical day, application data should naturally flow to all of them. That way, we'll spend less time on context switching and more time on actual work and play.

In this article, I'll describe how I built a photo album application that shares data over multiple Windows 8 devices. Building the application I used Modern UI principles to create a user friendly design and Windows Azure to handle the communication between the devices.

CloudAlbum is an application that helps you create custom albums by using templates provided in the application and access this albums from multiple devices.

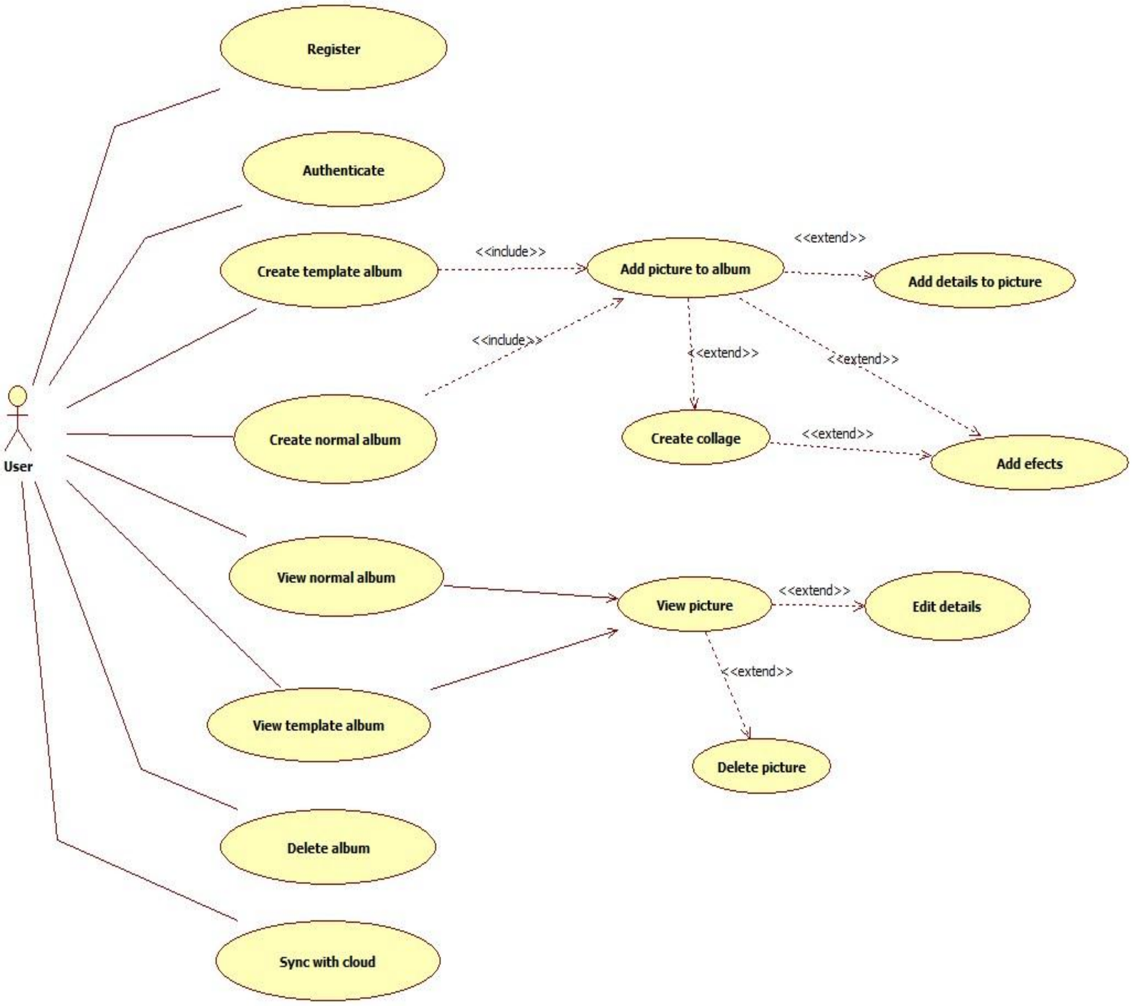
CloudAlbum application features:

- create an account for the user in order to access his photos on different devices;
- create/modify/delete/view albums;

- group pictures in albums;
- use given templates that can be customized for creating albums for different occasions (e.g. albums for the first year of the baby);
- create a collage, using a random or chosen template, from multiple photos and add it to an album;
- add details like description, date or location to the photos;
- add different effects to the photos (e.g. sepia, bakumatsu, threshold and others).

2. Architecture & Design

2.1 Use case scenarios



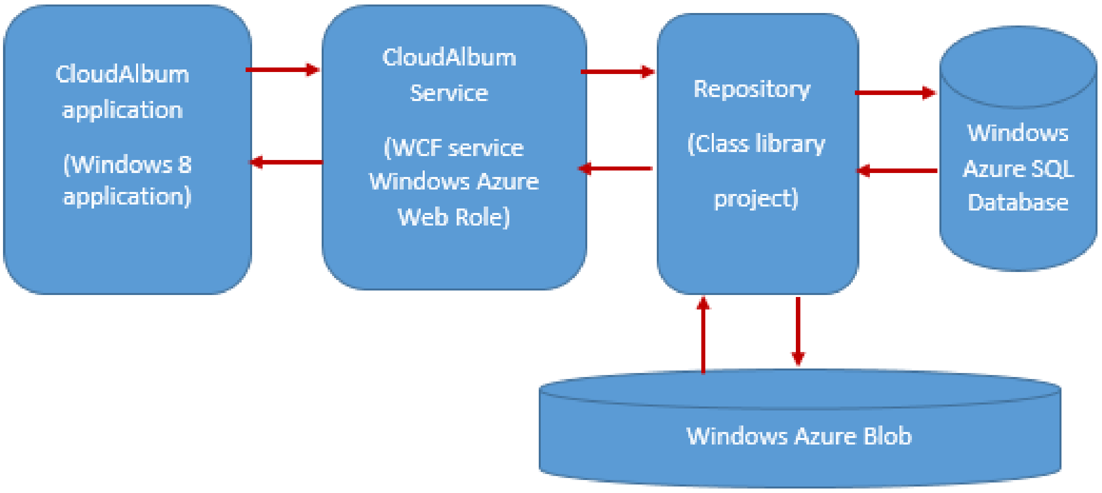
2.2 Architecture

CloudAlbum is a Windows 8 application created using C# and XAML, and following the Modern UI principles. The application consume a web service that is created using WCF.

The WCF web service is published in a Windows Azure Web role. The communication with the database is made using a Repository.

The Repository is a Class library that uses Entity Framework to store and retrieve data from the database and the Windows Azure Blob.

Data of the application is persisted in a Windows Azure SQL database and the pictures are stored in a Windows Azure blob.


CloudAlbum application (Windows 8 application)
CloudAlbum Service (WCF service Windows Azure Web Role)
Repository (Class library project)
Windows Azure SQL Database
Windows Azure Blob

I created the architecture of the application having in mind that the Windows 8 application represents the Presentation layer, the WCF service represents the Business layer and the Repository represents the Data Access Layer.

Using this layer separation I can easily reuse code and I can also extend or modify the application without having to make many changes.

2.3 User experience

I tried to create an elegant and efficient interface at every step of the development process, from computer sketch to graphics editing, even at coding and testing.

When creating a template album I use a selector to customize the look and feel of the album. This selector allows me to create different items for each templates, items that are composed from a different number of pictures and text sections. The user can easily add his pictures by clicking on the pictures placeholder. After this the user is took to a page where he can add details to the pictures and, if he clicks on the effects button he can apply effects on the picture.

The effects page is created using a clean UI, and the actions are brought to user attention only when needed. User can add one or more effects to a picture from a predefined list that is displayed when the user taps the effects button.

If the user wants to add something more special than a usual photo he can create a collage and add it to the album. When creating a collage the user has the possibility to create a random one, or select a type of the collage from a list, and after the type of the collage is chosen the user has to select the photos that will be used in the collage.

Normal albums can also be created in an easy and intuitive way, by adding pictures and details or applying effects to them as in a template album. This albums can be visualized as a slideshow.

The application creates a continuous user experience for the users, by allowing them to access their albums from different devices without feeling any difference. To protect their data, users are requested to authenticate in order to use the application on multiple devices.

3. Technologies, principles and patterns used in development

3.1 Modern UI

In order to provide a consistent, elegant and compelling user interface I tried to respect the design principles of the Modern UI Architecture.

3.1.1 Leverage Microsoft design style

Content is the heart of Windows Store apps, and putting content before chrome is fundamental to the design of Windows Store apps.
Everything else is accessory — or chrome — that helps present and enable interaction with the content.
The app's UI has to showcase the content, and make your UI accessible for users with disabilities. It also needs to have a clean and open layout in order to minimize distraction and help people get immersed in the content by leaving only the most relevant elements on screen.
Use typography in place of lines and boxes to help establish structure and hierarchy. Windows 8 applications need to have a familiar silhouette which helps people to instinctively recognize the app as a Windows Store app (as opposed to, for example, a webpage).
The most natural way for people to interact with content is directly, without intermediary chrome. Because some actions can't be accomplished with direct manipulation, we have to show commands contextually, only as needed, to avoid cluttering the app’s canvas. Leverage the screen or device edge so people can confidently find commands. [4]

3.1.2 Be fast and fluid

Respond to user actions quickly with matching energy. We can use purposeful animations - well designed animations bring apps to life and make the experience feel crafted and polished. Windows Store apps are designed to be touch first.
They leverage the hands and fingers for what they are great at, with comfort and ergonomics in mind, using the built-in controls to get fluid, beautiful, consistent animations and touch-optimized behaviours baked in. [4]

3.1.3 Snap and scale beautifully

The app has the opportunity to be used on hundreds of millions of PCs running Windows 8 that have a wide range of screen sizes and resolutions, ranging from <10" tablets to >27" screens.
Because people naturally multitask, they can run two apps side by side through snapping and, as a result, all apps must implement a snapped state. [4]

3.1.4 Use the right contracts

Contracts are the glue that binds Windows Store apps together and to the system UI. Two apps that have implemented the same contract can work together to complete a broad or complex scenario. The most common contracts are: share, search and file pickers. [4]

3.1.5 Invest in a great tile

A tile is the front door into an app. Sitting on the Start screen, it is an extension of the app and can provide much more personal and engaging information than a traditional icon. Invest in designing a great tile to draw people into your app. [4]

3.1.6 Feel connected and alive

Provide fresh content through live tiles and notifications to let people feel connected to your app. Both live tiles and notifications use the same infrastructure, and can be updated at any time using the Windows Push Notification Service, locally while the app is running, or at a pre-scheduled time.

Using live tiles you can draw people in continually with dynamic, relevant, personalized content on your app tile. Fresh tile content gives people a reason to place the tile in a prominent position on the Start screen, and to launch the app time again and again. Apps can use notifications to briefly show people time-sensitive messages that need to be interruptive from anywhere in Windows. [4]

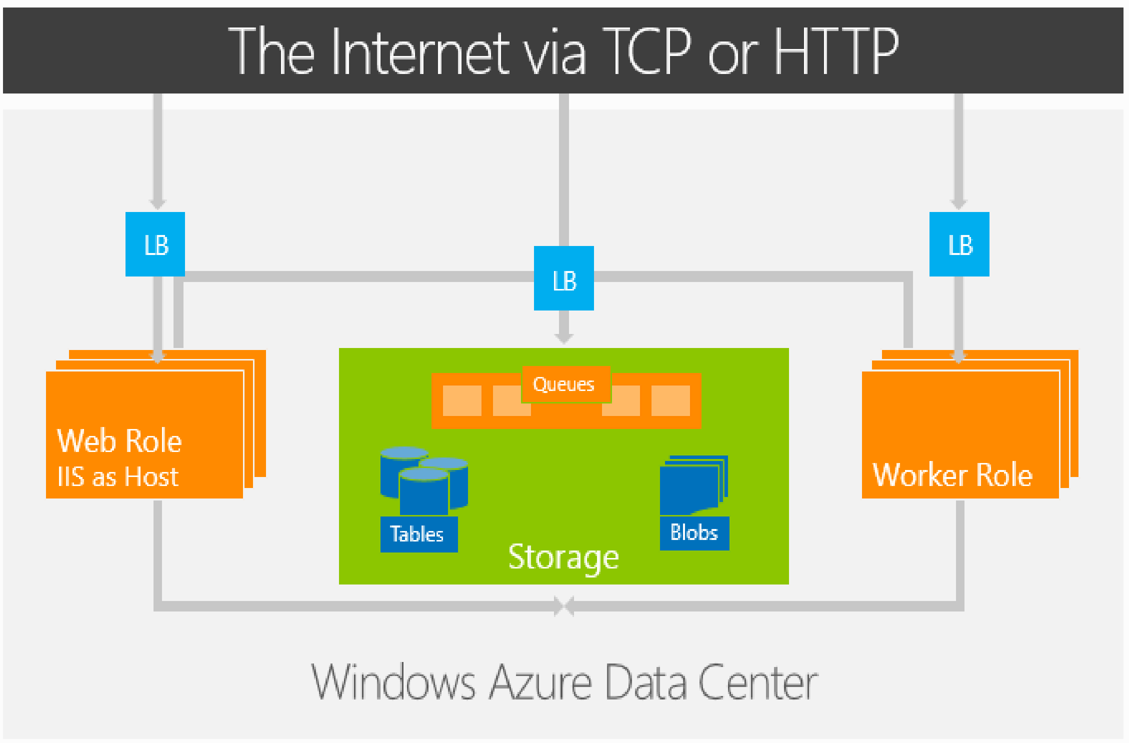
3.2 Windows Azure

Windows Azure is a Comprehensive set of services that enables you to build, host and scale applications in Microsoft data centers and it's ideal for applications needing: scalability, availability, fault tolerance, on-demand computing and performance.

Windows Azure core services are:

- Compute: worker role and web role;
- Storage: blobs, tables and queues;
- Database: SQL Azure.

Windows server architecture



Windows Azure Worker Role

General purpose host for executing code or an executable. It implements code in a Run method and it's similar to a Windows Service. It enables you to host your own web server, encoder, etc. Typically used for background processing.

Windows Azure Web Role

It is designed for web sites/services accessible using HTTP and provides the developers with all features of a worker role and IIS 7 or 7.5. Execute ASP.NET, WCF, PHP, etc. Can include multiple web sites in the same role.

Blobs, tables and queues

The Windows Azure storage services provides storage for binary and text data, messages, and structured data in Windows Azure.

The storage services include:

- The Blob service, for storing binary and text data
- The Queue service, for storing messages that may be accessed by a client
- The Table service, for structured storage for non-relational data

SQL Azure

SQL Azure is SQL Server database technology delivered as a service on the Windows Azure Platform and it is ideal for both simple and complex applications, being designed to scale out elastically with demand.

Supports popular languages: .NET Framework (C#, Visual Basic, F#) via ADO.NET, C / C++ via ODBC, Java via Microsoft JDBC provider, PHP via Microsoft PHP provider and popular frameworks: Entity Framework, WCF Data Services and others. It also has support for popular tools like SQL Server Management Studio (2008 R2 and later).[2]

The Load Balancer

Azure handles load-balancing for table storage and SQL Azure by distributing traffic across the instances you set up. For example if you have a web role that run on two instances and a new request come, the request passes through the load balancer and this decide which instance will handle the request (in our example if the first machine has the CPU at 75% and the second at 15% the request will be passed to the second instance).

3.3 Entity framework (EF)

EF is a data access framework from Microsoft that helps bridge the gap between data structures and objects in your applications.

What Does It Do?

- Develop against conceptual view of your data, instead of data store itself.
- Plumbing and mapping is done for you.
- Generates and maintains entity classes.
- Generates code for interacting with database, including connection.

3.4 WCF Services

Windows Communication Foundation (WCF) is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data. [5]

WCF uses three types of contracts:

- **Service Contract**: defines Operations, Behaviours and Communication Shape;
- **Data Contract**: defines Schema and Versioning Strategies;
- **Message Contract**: allows defining application-specific headers and unwrapped body content.

4. Future development

The market is in a continuous evolution nowadays and my application has to evolve along with it. To always satisfy the users you have to not only make the application stable and easy to use, but you also have to add new features, to add something fresh so that users will have the desire to always come back to it. That’s why, I intend to improve the application by extending it with the following features:

- users have the possibility to share the albums between them;

- use geolocation to add the location where the picture was taken;
- give users the possibility to create their own specific templates;
- support for multiple languages.

5. Conclusions

Even though there are applications for editing and manipulating pictures in the Windows Store, there is no application which allows you to create your own albums, from predefined templates, and sync it with Azure, in the same time.

Implementing CloudAlbum, I showed that it is possible to create modern applications that help user to use their data on multiple devices. This feature is possible because the data is stored in Windows Azure in a more safe and secure way.

Developers will try to take their applications to a higher level, to break the barriers between devices and to create new user experiences. Developing the CloudAlbum application I reached the conclusion that you need to have a well-designed architecture and good principles design to follow, in order to achieve this continuous user experience.

I can conclude saying that the CloudAlbum, an application that is designed based on Modern UI principles and using Windows Azure and WCF services, breaks the barriers of the devices and creates a continuous experience for the users.

References

[1] Adam Freeman, *Metro Revealed: Building Windows 8 apps with XAML and C#*, Apress, 2012
[2] Sriram-Krishnan, *Programming Windows Azure: Programming the Microsoft Cloud,* O'Reilly Media, 2010
[3] Juval Lowy, *Programming WCF Services,* O'Reilly Media, 2007
[4] http://msdn.microsoft.com/en-us/library/windows/apps/hh464920.aspx
[5] http://msdn.microsoft.com/en-us/library/ms731082.aspx

Simona – Mihaela TECŞA
"Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Raţiu Street No. 7-9, Sibiu, 550012
ROMANIA
E-mail: tecsa_simona@yahoo.com

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Hu moments based shape recognition algorithm

Ana Tuba
Teacher Coordinator: Milan Tuba

Abstract

Shape recognition in digital images is an active research area. In this paper we propose an algorithm for shape recognition based on Hu moments. Since single shape can be in different positions, any template based recognition is very difficult. Hu moments are invariant to many transformations like translation, rotation and scaling and proposed algorithm achieves acceptable recognition ratios even in the initial form, without any pre-processing which can significantly increase recognition success.

Keywords: Digital image, Image processing, Shape recognition, Hu moment

1 Introduction

Digital images are very popular and there is a lot of situations where they are used. Having images in digital form allows processing by many useful mathematical operations that increases their overall usability. One very much utilized area of digital image processing is image segmentation and shape recognition which are used in artificial intelligence systems for image analysis and understanding. When images are taken, requirements such as high contrast, even color saturation etc. that are essential for image recognition are often not taken into consideration. This is due to limitations of technologies used to obtain images (X-Ray, Radio, Infra-red images etc.) or the fact that they were not originally intended to be used for automatic software recognition. For these reasons there is usually a need to adjust images to make process of recognition easier and more precise. Unfortunately, there is no universal technique or algorithm that will work well with all types of images and it is usually based on trial and error to find the optimal algorithm that will work on a given set of images for particular purpose.

Images can include various deformations (e.g., quantization noise, illumination differences, or any number of other "natural" and instrumental sources of variability (e.g., mixed pixels). Because of that, the first step of a shape recognizer can involves pre-processing which include the size and aspect ratio of normalized image, the interpolation technique of pixel values, etc. The second step is feature extraction. The selection of appropriate feature extraction method is probably the single most important factor in achieving high recognition performance. In this paper a feature extraction method based on moment invariants was applied to shapes. Invariant features have the strength of keeping the same value despite geometric transformations: scaling (Fig.1 b), translation (Fig.1 c), rotation (Fig.1 d), etc. The third step is the classification process which needs features that contain enough information about the class.

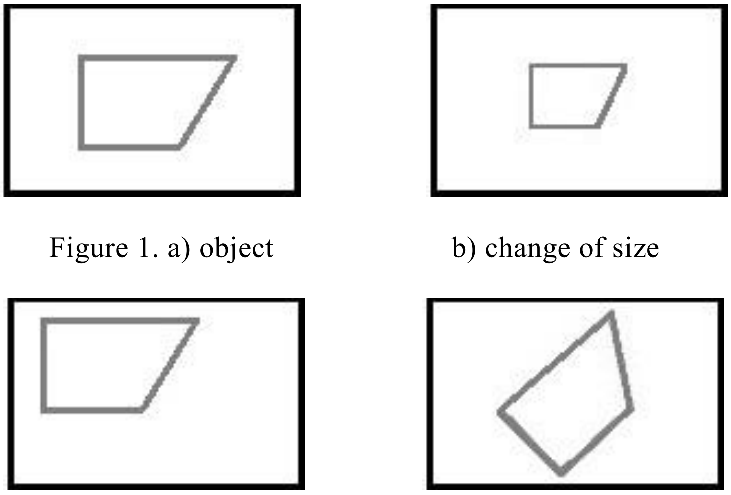


Figure 1. a) object b) change of size

c) change of position d) change of orientation

The rest of the paper is organized as follow: in Section 2 basics of shape recognition and template matching are introduced. In Section 3 Hu moments are described. Section 4 deals with our program and test results. Finally, conclusions are discussed in Section 5.

2 Digital image

Digital images are composed of pixels (short for picture elements) [1]. Each pixel in the picture is a point representing a color or gray level of black-and-white images, so a pixel is like a small dot of a particular color. One bit is needed for black and white images (0 - white, 1 - black), while for a number of shades of gray a larger number of bits is required (e.g. 256 shades of gray - 8 bits). The primary colors for mixing light are red, green, and blue (RGB) [2]. Each pixel on a color image contains three numbers corresponding to the red, green, and blue levels of the image at a particular location. Any color can be created by mixing the correct amounts of red, green, and blue. If we have 256 levels for each primary, each color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors. The images usually are stored row by row, from left to right and top to bottom. Rows of image elements are called raster scan lines. To describe the color of each image element (color depth), we can use length of raster-scan lines (the number of picture elements on horizontal axis), the number of raster-scan lines (the number of picture elements in vertical axis) and the number of bits. That data is necessary for the reconstruction of recorded lines.

2.1 BMP image file format

Popular and simple file format that is used in our software system is BMP without compression and 24 bit color depth since it is the easiest format for manipulation and in this experimental system we are not concerned with performance issues.

BMP file format is used today despite that it is historic [3]. It was made to work very easy with Windows application. Independently of a display device, this simple raster graphics image file format stores bitmap digital images. Sometimes it is called Device-Independent Bitmap (DIB) since the image is held in memory as a DIB structure. For storing 2D digital images we can use BMP file format. The images have width, height, resolution, and they can be either monochrome or color, in various color depths (1, 4, 8, 16, 24, 32 or 64 bits per pixel) and also, colors can be indexed. Also, there is specific lossless compression algorithms (fast to decompress) and alpha channels as optional features. Images 16bpp and 32bpp are always uncompressed, and also most BMP images are uncompressed in practice. BMP file structure has two headers, 14-byte BitmapFileHeader and the second known as a DIB header. The first allow a check for the file type and for damage. This header specifies the type of bitmap file, the size of the file and the position in the file where the pixel data begins. DIB header contains technical information (bits-per-pixel, image width and height in pixels, compression method) needed to render the image.

3 Shape recognition

Shape recognition can be used in several situations, for example, detection of similar shapes in the database, or parts that fit together. This is mostly automatic analysis of geometric shapes. Objects must be represented in digital form to a computer to automatically analyze and process geometric shapes. The shape is usually described using boundary representations, but can also be used other volume based representations (constructive solid geometry) or point based representations (point clouds).

In practice, we use shape recognition in numerous areas. In archeology it is used for finding similar objects, missing parts, in architecture, to find objects that are spatially fit into some space. Also in medicine, it is used for surgical planning, or in security applications for face recognition. There is a place for shape recognition in entertainment industry (the movies, games etc.). There is many more areas where we use it.

Objects can be given by computer-aided design, scanning, separating shape from 2D or 3D images. In any case, the objects must be simplified before comparison. Simplified representation is called shape descriptor. These simplified representation carry most of the important information and it can be used as long as it is easier to handle, store and allow comparison of the forms directly. It is possible to completely reconstruct the original object using the complete shape descriptor.

We can classify shape descriptors by invariance considering which of the transformations are allowed. Congruent shapes, the shapes which can be translated, rotated and mirrored, will always have same descriptor. Intrinsic shapes descriptor is another class of shape descriptors and is invariant to isometry. This means that different isometric embedding of the shape will not change the descriptor. This allows us to apply these descriptors nicely to deformable objects which do not have a lot of stretching. This shows that different shape descriptors target different aspects of shape and we can use them for specific application. In conclusion, we should always do a complete analysis of how a descriptor captures the relevant features.

Pattern recognition is the science of making inferences based on data [6]. Shape recognition is very popular topic of research for many years. It is an essential part of any high-level image analysis systems. Most of these systems share a general structure of four building blocks: Image acquisition, pre-processing of the images, feature extraction, and classification. To improve the classification results in various situations, many approaches have been studied including decision theory, feature selection, optimization, learning and so on. An effective shape descriptor is a key component of multimedia content description, since shape is a fundamental property of an object. There are two types of shape descriptors: contour-based shape descriptors and region-based shape descriptors. Regular moment invariants are one of the most popular and widely used contour-based shape descriptors.

In human's perception, recognition and understanding the shape of an object is extremely important factor. A challenging problem in computer vision and shape recognition is object shape comparability. In these applications the usual method is the method based on the geometric moment invariants. In computer vision, the analysis and discernment like geometry is very significance. The reason is that geometric shape is the main characteristic of an object. It has invariance with respect to translation, orientation or scale. Because of versatility of the shape of the object which results in numerous kind of expression ways, there is no way so far that is completely correspond to human's recognition and comprehension of shape. But there has a big success on the research of matching in shape characteristic of the object. Fourier descriptor, moment invariants and boundary energy function and much more is the expression ways of image contour. Having good invariance on translation, scale, orientation and anti-interference of moment invariants contribute to very valuable achievements in research. But problem is that these descriptors often give a poor discrimination between complicated objects. To describe shape of object, shape parameters are taken from the object. It is using to compare to the shape of template objects or to partition objects into classes of different shapes. But now we have the important question, how shape parameters can be made on certain transformation. We need to find shape parameters that are scale rotation invariant or that invariant under affine or perspective projection, because object can be viewed from distinct distances and point of view.

4 Moments invariants

Shape recognition depends on the variability in the feature values for shape in the same category, relative to the difference between feature values for shape in different categories. Features should be distinct for different shapes images, so that the computer can extract the correct shape. The features of moments are used in shape recognition system because they are remains invariant whenever the recognizable shapes are resizable drawn or orientation variant through sequence of images.

Moment invariants have become the most popular tool for object recognition and they are substantially shape descriptors in computer vision. They can been frequently used as features for shape recognition, image processing, remote sensing, etc. Characteristics of an object that absolutely represent its shape also can be seen from moment.

4.1 Hu moments

Invariant shape recognition can be performed as to classify in the multi-dimensional feature space of invariant moments. There are several techniques for shape recognition and representation, which is based on the invariants feature from moment. The techniques are differentiated by the definition of moments, such as the type of data, the method for obtaining the values of invariant image moments. Moment invariants are firstly introduced by Hu [7], [8]. Independent to translation, scale and orientation there are seven of these shape descriptor values, according to Hu. They are all computed from central moments through order three. By computing normalized moments with respect to the center of gravity w achieve translation invariance. That way the center of mass of the distribution stays at the origin. Simple size normalization can lead to size invariant moments. Independent of rotation set of seven invariant moments can be computed from the second and third order values of normalized central moments. The moment invariants have been proved to be the adequate measures for tracing image patterns regarding the images translation, scaling and rotation under the assumption of images with continuous functions and noise-free. Moment invariants have been extensively applied to image pattern recognition, image registration and image reconstruction. However, the digital images in practical application are not continuous and noise-free, because images are quantized by finite-precision pixels in a discrete coordinate. In addition, the noise may be introduced by various factors such as camera. In this respect, errors are inevitably introduced during the computation of moment invariants. In other words, the moment invariants may vary with image geometric transformation. Because of the implementation process moments used to construct the moment invariants are computed in the discrete form instead of the continuous form in which they are defined. The basic definition for moment invariants is as follows: two-dimensional moments of a digitally sampled M × M image that has gray function:

$$f(x,y),\ x,y=0,\dots,M\text{-}1 \qquad (1)$$

Given a function *f(x,y)*, these regular moments are defined by:

$$m_{p,q}=\iint x^p y^q f(x,y)dxdy \qquad (2)$$

The order of the moment is *(p+q)* where *p* and *q* are both natural number. For implementation in digital from this becomes:

$$m_{p,q}=\sum_{x=0}^{M-1}\sum_{y=0}^{M-1}x^{p}y^{q}f(x,y),\ p,q=0,1,... \quad (3)$$

The moment *f(x,y)* translated by an amount (a,b), are defined as,

$$\mu_{p,q}=\sum_{x}\sum_{y}(x+a)^{p}(y+b)^{q}f(x,y) \quad (4)$$

To normalize for translation in the image plane, the image centroids are used to define the central moments. The co-ordinates of the center of gravity of the image are calculated using equation (4) and are given by: $a=-\frac{m_{10}}{m_{00}}$, $b=-\frac{m_{01}}{m_{00}}$. The moments are further normalized for the effects of change of scale using the following formula:

$$\eta_{pq}=\mu_{pq}/m_{00}^{y} \quad (5)$$

where the normalization factor $y=[(p+q)/2]+1$.

In particular, Hu defines seven values, computed by normalizing central moments through order three, that are invariant to object scale, position, and orientation. In terms of the central moments, the seven moments are given as:

$$M_1=(\eta_{20}+\eta_{02}),$$
$$M_2=(\eta_{20}-\eta_{02})^2+4\eta_{11}^2,$$
$$M_3=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2,$$
$$M_4=(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2,$$
$$M_5=(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right]+(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$$
$$M_6=(\eta_{20}-\eta_{02})\left[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03}),$$
$$M_7=(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right]-(\eta_{30}+3\eta_{12})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right]$$

5 Program Code

Our program necessitate using images contain white shapes on black background. The shapes in test images may be seen disarranged, rotated, or in different sizes. The work branched into two streams: training and recognition. The training stream goes to enroll the database of the considered shapes in different situations of size, pose, and orientation. Image of shapes must be in binary mod.

5.1 Binarization

A digital image with only two possible values for each pixel is a binary image, also known as bi-level or two-level image. Black and white are two colors which usually used for a binary image. The foreground color is the color that is used for the object and everything else on the image is the background color. Bi-level or two-level means each pixel is stored as a single bit and for that concept are used names such as black-and-white, B&W, monochrome or monochromatic. But any images that have one sample per pixel also is define by that names (for example grayscale images). As the result of the operations (such as segmentation, thresholding) or in digital image processing as mask we often get binary images.

Binarization is the process of converting the image into binary mode. The goal of the binarization is to separate the shape from the background in the gray image, also to reduce the image levels into black and white. A gray level image may be converted to a binary mode by threshold process. If a pixel's value is higher than the global threshold, it is assigned to 255 otherwise it is assigned to 0.

5.2 Program

Our program is made of two parts. The first is called training, and second is shapes recognition. The program trains by the loaded image, in which shapes are drawn in white on a black background, for example circles. Loading many images different circles, of different sizes, shapes (ellipse is recognize like circle), at various points on the background. Images from our test are shown on Figure 2.

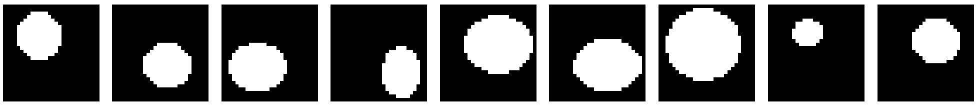


Figure 2. Images of circles

Then, program calculated seven Hu moments. Our test is loading ten images of circles and Hu moments are shown in Table 1. After that we calculate the average of the first moments, then second etc. (Table 1). That represents features of circle. Then do the same thing for other shapes: squares, triangles, and so on.

	M1	M2	M3	M4	M5	M6	M7
Image 1	0.797456	3.92311	5.29459	8.11605	14.8273	10.2596	15.6072
Image 2	0.793865	3.9312	5.29679	8.11607	14.8496	10.1279	15.6073
Image 3	0.79152	3.066	6.13875	7.93616	14.9736	9.46916	20.1482
Image 4	0.780557	2.65243	5.98104	7.5916	14.384	9.03981	15.1586
Image 5	0.792149	3.1242	6.34578	8.3471	15.6937	9.91603	17.2338
Image 6	0.783651	2.75005	6.21118	7.9804	15.6005	10.12	15.0965
Image 7	0.797874	4.29798	6.32845	9.46038	17.3746	11.7436	17.8845
Image 8	0.77641	2.56714	5.95043	7.41185	14.093	8.69542	19.2226
Image 9	0.797456	3.92311	5.29459	8.11607	14.8273	10.2596	15.6073
Image 10	0.796832	3.38352	5.48385	7.5598	15.9898	9.60466	14.0817
Average	**0.790777**	**3.361874**	**5.832545**	**8.063548**	**15.26134**	**9.923578**	**16.56477**

Table 1. Hu moments for ten images of circles and average of each moment

The second part of program is shape recognition. Program load images of unknown shape (Figure 3) and calculate their seven Hu moments (Table 2).

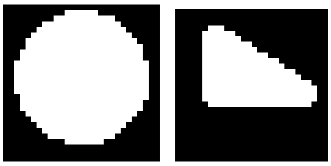


Figure 3. Images of two unknown shapes

	M1	M2	M3	M4	M5	M6	M7
Unknown shape 1	0.798225	3.86325	6.18908	7.8754	16.2473	10.0191	18.024
Unknown shape 2	0.677591	1.89752	2.56182	3.65205	6.85677	4.77847	6.97928

Table 2. Hu moments for unknown shapes

Then comparing with characteristics of shape which we already known and looking which shape is the most similar. We calculate distance of Hu moments known shape (Table 3).

	M1	M2	M3	M4	M5	M6	M7
Unknown shape 1	0.007448	0.501376	0.356535	0.188148	0.98596	0.095522	1.45923
Unknown shape 2	0.113186	1.464354	3.270725	4.411498	8.40457	5.145108	9.58549

Table 3. Distance of Hu moments for unknown and known shapes

Unknown shape 1 is circle, because distance of every Hu moments is very small. Unknown shape 2 is not circle. That we can't see from first and second Hu moments, but from another four are clear that this is not the same shape.

6 Conclusion

Software system is developed that facilitates, through user-friendly graphical interface. There is a lot of research in the area of shape recognition that are appropriate for specialized types of images and applications. Further development of this software system will include neural network which make it easier to recognize shape.

References

[1] Wilhelm Burger, Mark J. Burge, *Principles of Digital Image Processing*, Springer-Verlag London, p. 261, 2009.

[2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Prentice Hall, p. 793, 2002

[3] Wim Wouters, *Windows Bitmap File Format Specifications*, Clean Coding Company, 2008

[4] S. Belongie, J. Malik, J. Puzicha. *Shape matching and object recognition using shape contexts*, PAMI, 24(4):509–522, 2002

[5] M. Mercimek, K. Gulez, T. V. Mumcu, *Real object recognition using moment invariants*, Sadhana, Vol. 30, Part 6, 2005, pp. 765-775

[6] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification,* 2nd edition, (New York: John Wiley & Sons, 2000

[7] Ming-Kuei Hu, *Visual pattern recognition by moment invariants*, IRE Transactions on Information Theory, Utah, 179-187, 1962.

[8] L. Keyes, A. Winstanley, *Using moment invariants for classifying shapes on large-scale maps*, Elsevier, Vol. 25, Issue 1, 2001, pp. 119-130

[9] A. Mukherjee, S. Kanrar, *Enhancement of Image Resolution by Binarization*, International Journal of Computer Applications, Vol. 10, No. 10, 2010, pp. 15-19

Ana Tuba
University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

Prof. Milan Tuba
University Megatrend Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Belgrade
SERBIA
E-mail: tuba@ieee.org

Third International Students Conference on Informatics
Imagination, Creativity, Design, Development
ICDD 2013, May 16-18
Sibiu, Romania

Graphical interface for Network On-Chip Simulators

Vilceanu Dumitru
Teacher Coordinator: Mircea Vladutiu

Abstract

According to the latest needs of bandwidth and higher integration, nowadays computers tend to increase by complexity, and the most significant part which should be taken into consideration is their structure which tends to become on-chip micronetworks. As every network, the architecture is made by nodes and the interconnections are done by edges, but the main subject is the feasibility and advantages of this architecture. We should consider that NoC provide a good scalability and performance concerning the designers, and also for the manufacturers, but the question is how much it will take until the structure fails, in other words, what should happen when the congestion takes place, but nevertheless the network should work at least partially.

Keywords: NoC, topology, flits, latency, throughput, bandwidth, congestion.

1 Introduction

From the invention of the transistor until the first computer came out didn't took too long. After this huge step, the problem of connecting two or more computers was solved using a network. But a bigger problem is that while the requirements regarding the performance have increased, the scalability became a problem of matter for every computer. At that moment the structure of a computer was very straightforward and the central processing unit was the component which was supposed to be the brain, which should be defined by a complex set of electronic circuitry capable of executing stored program instructions.

A significant step in computer's ascending was designing the system on chip which integrates on a single chip components of a computer and other electronic system. A SoC can be compared with a microcontroller, but the main difference is that SoCs provide more powerful processors with the option of connecting external memory chips. SoCs may include more than one processor so it will be called Multiprocessor System-on-Chip (MPSoc) [5] which is usually designed for embedded applications.

SoC's can be manufactured by several patterns like: full custom - where are specified the elements of the layout which are the transistors and the interconnections between them, standard cell - where an abstract logic representation (such as AND gate) is actually a low-level very-large-scale integration (VLSI) layout, but also FPGA - where the components are programmable logic, and using hierarchy of reconfigurable interconnects it is possible to obtain different configurations by wiring together components which are usually called "logic blocks".

While the number of cores within a SoC has increased, the communication in the subsystem of IP cores has been assigned to a new approach called Network-on-Chip. While such systems are implemented should be taken into consideration on-chip communication issues. As it is expected this new approach will overcome today's on-chip interconnection and communication challenges because of the integration of an overwhelming large number of store, logic, and computational blocks in a single chip. Power consumption is one key factor, which can be easily compared to the integration density, because both of them have been increasing similarly and the levels that has been reached are unacceptable. In the near future, even if at this moment it has just a small impact, it is presumed that power consumption value will be high influenced by the leakage and subthreshold currents, obtained through high integration. The system performance will be influenced by the wire delay as a result of inefficient scaling of global wires in higher metal layers. Hence, having a large chip size which uses high frequency will be very hard to have a synchronous design, so the design will have limited modules synchronized and these modules will be working globally asynchronous.

NoCs paradigm is facing almost the same problems like every micronetwork, so before the manufacturing process a great deal of time should be invested in analyzing which design fits better with the requirements.

This paper provides a description of NoC topologies, how they are implemented, but also making comparison between different results of NoC simulators nowadays. Things that are taken into consideration is the communication between nodes, routing of packets through several hops, via switches, parallelism, efficient sharing of wires. A user interface for the NoC simulators has been developed to be flexible and adaptive to every NoC simulator, to enhance the comparison of several simulations, to have a clear vision of simulation process, but more important is to determine the congestion [2] moment of the network. The simulation will also gather the number of clock cycle passed, the real time passed, the state of the machine that runs the simulation, the number of cores, the routing scheme, the number of packets generated [3], the number of packets received, but most important the packet injection rate.

2 Background

We provide a brief background on on-chip architectures and NoC simulators that are used to test NoC topologies using different case situations. Depending on the design of the architecture, the arbitration mechanism and other parameters we are trying to identify key factors that influence congestion through the network.

2.1 NoC architecture

NoC paradigm tries to cover the problem which long wires are facing regarding the latency and electrical behaviour, but also the disadvantage of on-chip busses regarding the speed because of connecting components which are spread across the chip. Main advantage of on-chip networks is that wire lengths between adjacent routers can be short and uni-directional. NoC while supporting better pipelining of data than the busses doesn't affect reducing the design complexity, that's why after a router is designed, it is just replicated very easy wherever is needed.

By borrowing concepts from computer network where communication is performed by sending and receiving packets, NoC implement the same principle of routing packets using the network and transport layer that any source module can send messages to any destination module via different intermediate nodes (several hops).

A major difference between off-chip networks and on-chip networks is the bandwidth which is lower for off-chip networks, and also while for off-chip networks the expense of each chip pin is the main constraint, on-chip networks are constrained by the number of metal layers and also the number of hops through wire routing. The optimum choice selecting best routing algorithm and network topology for NoCs is related to the bit-width, because a greater bit-width allows sending much shorter packets for the same amount of date compared to an off-chip network.

Network performance and energy-efficiency are determined mainly by the network topology which acts like a map for the connectivity among nodes. Mainly, a high-speed router is the one that connects neighbours by links, but also the cores from the nodes. So, the topology defines logical layout, while physical layout is defined by the floorplan. Compared with a traditional off-chip network, NoC's topology is statically known, but also very regular that most common topologies are mesh and torus which constitute over 60% of cases.

The most typical and one that has been used from early NoC designs is the two-dimensional (2D) Mesh shown in Fig. 1. In this topology, each router (switch) has 5 input and 5 output ports which connects each one from its four neighbours to the network interface (NI). In the best case scenario each packet spends between 1 cycle and 4 cycles at each router before being forwarded, and this depends on router architecture, arbitration but also routing schemes. This type of topology presumes a relatively small size network and because the topology is statically known can be implemented global coordination mechanism but also packet routing must be base on local decisions, because a packet spends only a few cycles through a hop. To overcome the large number of hops which are needed for mesh topology, it turned out that low-diameter NoC topology are better because of lower number of costly router accesses, so improving network latency and energy efficiency. An example of low-diameter NoC topology is butterfly which is the derivative of fat tree architecture.

Routing in NoC can be done using two kinds of algorithms: oblivious and adaptive. First type algorithm routes the packet/flit [1] according to the source and destination address, while the second type of algorithms adapts to the dynamic network conditions in order to avoid congestion. Using deterministic routing is an advantage when the network is not congested, providing low latency and simplicity of routers design. On the other hand, deterministic routers are going to have decreased throughput when the packet injection rate increases, because the mechanism cannot handle network congestion. Unlike deterministic routers, the adaptive ones are able to handle congestion and use alternative routing paths in order to avoid congested links. However, in order to decide a good routing path, extra logic is needed and so adaptive routing is likely to have higher latency when network congestion is not very high.

The packets that are conveyed through the network are either request/control messages or data messages and are partitioned into flits: the smallest independently-routed unit of traffic which is conveyed by one link in one cycle.

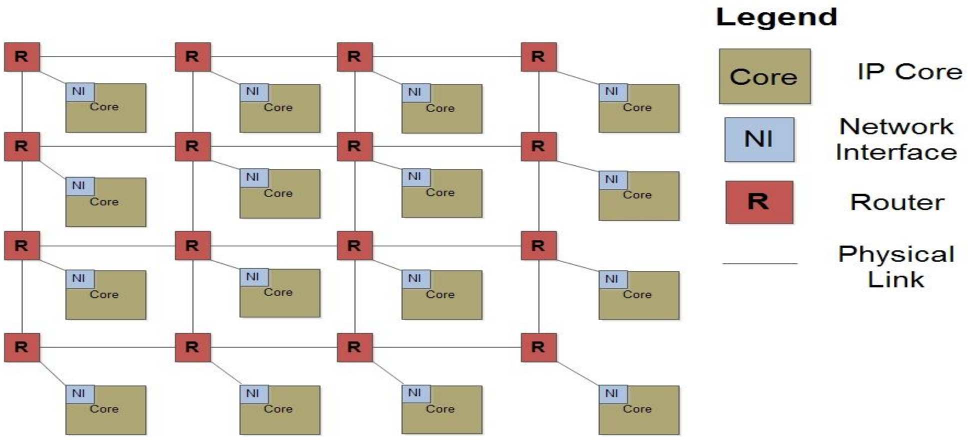


Fig. 1

2.2 NoC simulators

According to the designer's requirements, the NoC architecture should carry out a wide variety of parameters and exploring these items can be done only at a high-level [6] When designing a Multi-Processor System-on-Chip architecture which contains a heterogeneous set of processing nodes interconnected by a NoC there are problems which should be taken into consideration regarding hardware/software codesing. Before the manufacturing process, one must evaluate all kinds of trade-offs, e.g. operation bit-widths, power consumption, memory sizes but also performance variables like latency, throughput. In order to obtain a good product, the MPSoC architecture is simulated intensively before having it in silicon. Two types of simulations are used for such large designs: the first one uses high level modelling which is non-cycle accurate, and the other one requires long simulation times because it uses low level modelling and high cycle accuracy. For systems which consist of several hundreds of nodes, high level modelling is better because it leads to lower simulation times than the low level modelling. These tests are done by the designer and let's him now if the design is ready for manufacturing.

In the near future the routers with communication infrastructure will have increasing amount of services compared to traditional busses. The design must provide guarantees such as: guaranteed throughput, bounded latency, and low risk of congestion. These requirements are related to the request of real-time applications which number has increased significantly. In order to determine in which manner these requirements influence the NoC design we need to study precisely the behaviour of the application-generated traffic through the network. The designer cannot decide just analyzing a single router that the local optimal schedule will be enough for the whole network, because this choice may cause buffering problems in the neighbouring routers. We will analyze methods to study the design choices of the network by reviewing the latency and throughput behaviour of NoC having different parameters [7].

2.3 Congestion

According to data networking and queuing theory, network congestion occurs when a link or node is carrying so much data that its quality of service tails off [7]. The effects of congestion are: queuing delay, packet loss or the blocking of new connections. When the number of lost packets increases, the throughput decreases, but also the latency increases because of the delayed packet which don't arrive at the destination in the expected time. When congestion occurs in high levels it is very much possible that no longer will be possible to efficiently inject packets into the network, because the router won't have too often free slots. Such situation is also known as starvation [7].

2.3.1 Critical package injection rate

Packet injection also known as spoofing packets is a computer networking term that defines the process of interfering with an established network connection, by means of generating packets and inserting into the network as if they are part of normal communication stream. The packet injection process in NoC allows every node to generate a packet which usually can be described by the source which created the packet, the destination where the packet is expected to arrive, data to transfer but also to have a better flow [4] of the generation time frame should be described by the time when it has been created and why not how much time should it be valid. When the validation time expires, a packet should be dropped from the network because it is considered that didn't arrive at destination in the expected time.

Two flows in NoC will affect each other when the two routes belonging to these two different flows share common nodes, and if the flow cannot continue because the routing scheme doesn't allow it at that moment the congestion appears, and considering the fact that all the packets where generated using a package injection rate, we will take for granted that the critical value is the one that is was used at the moment when congestion appeared. Different values of critical injection rate can be chosen considering different values for congestion value.

3 Experimental support

This paper is a help support for the designer of NoC architecture by analyzing the results of simulation traffic through the network, in order to enclose as best as possible the network design performance. Quality-of-service (QoS) regards to the guarantees given by data transfer speed and fidelity for data transfers from source to destination. Worst case scenario should guarantee timing parameters like: minimum throughput, maximum latency, maximum latency jitter, but also packet delivery parameters like: in order, or out of order.

Best effort (BE) scheme is one that forwards packs as soon as they are generated by the source, but not having any guarantee regarding the latency or throughput. Prioritizing packets can offer relative guarantees, but none exactly and even if no performance results are sure, it is the most common approach nowadays.

Guaranteed throughput (GT) scheme extends from a priori knowledge about the traffic load conditions, and can offer a minimum level of successfully transferred packets through the network. Even it is said to have a clean guarantee this scheme has a complex traffic modelling and it is error-prone if simultaneously activated circuits aren't statically determined also.

3.1 Methodology

Supposing that the NoC design that it will be used for simulation will have a packet-switched router which implemented wormhole routing with virtual channel (VC) flow control we can easily notice the advantage of routing scheme used that the packet-size doesn't depend on buffer-size. The virtual channels are used to decrease the change of congestion and increase the GT traffic value. Each packet in the network consist of H header flits to setup a route, an arbitrary chosen number of data flits that contain the packet's information and tail flit which is supposed to free the router's resources for other packets. Let's suppose that one packet has been generated by S1 (source) and has to arrive to D1 (destination), but also another packet has been generated by S2 (source) and has to arrive to D2 (destination) as shown in Fig. 2. During the whole process we assume that flits from both packets have to go throw a common path which contains three routers. Well for having the throughput that we want, and not to encounter congestion we have chosen 2 VC for that common path, so that flits which need to reach D2 won't get through the same VC as the flits which need to reach D1.

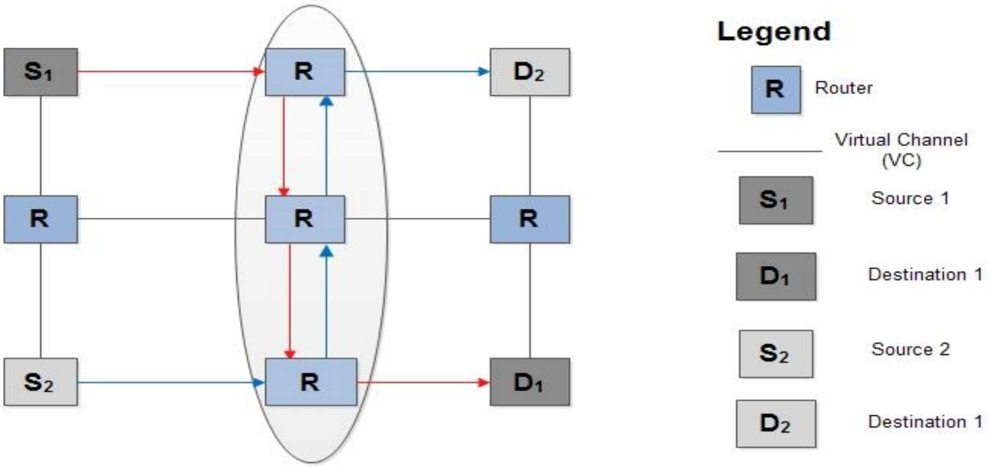


Fig. 2

We assumed that the chosen topology will have first of all every neighbour for each node from the network defined, the number of ports for each node, but also the number of virtual channels per each port or the router, or the designer can use the user interface to set different number of virtual channels. In other words the total number of flits per router equals ports * VC * flits, average being 5 * 3 * 6 flits, that means 90 flits per router.

3.2 Graphical interface

In order to have a quick view of the machine state that runs the simulation we implemented a module to show two gauges which show the CPU usage, but also the RAM memory usage in MB. The values are automatically refreshed by using timer which gets the actual state of the machine every 0.5 seconds. The user can set a threshold value which will act like a warning state, as it can be seen in Fig. 3 and a red led will blink because the value was set to 4500 MB.

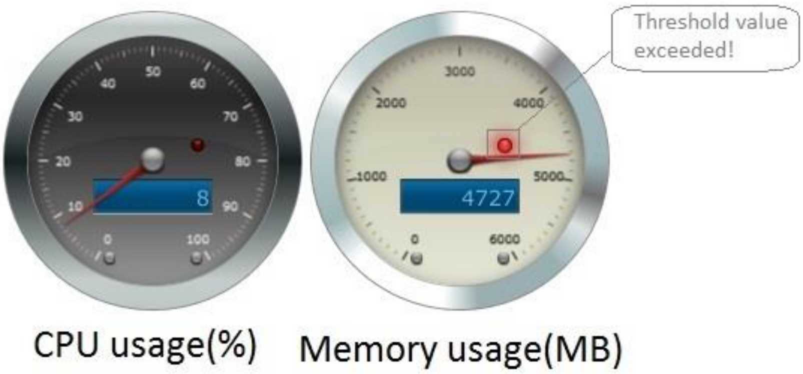


Fig. 3

In order to select the topology file we used a file browser which will let the user select and load the topology file but also the routing scheme if it is a deterministic one.

After the topology file is loaded a short summary is shown like: number of nodes, average degree, degree distribution, and average path length.

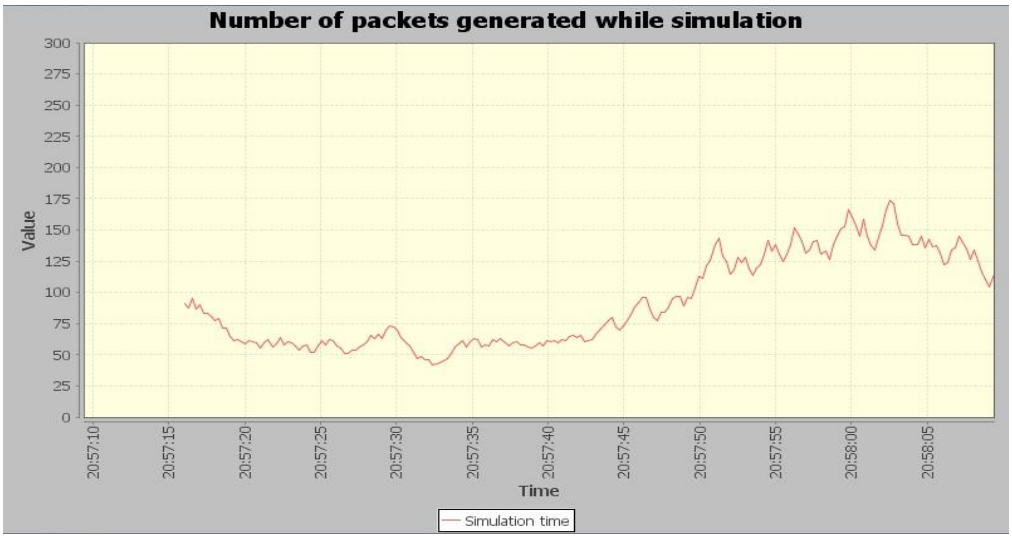


Fig. 4

Before starting the simulation the user can set up also simulation time (in clock cycles), so if the congestion will not overcome while simulation, at the end of the predisposed time a log file with the simulation result will be created. The results are interpreted using this log file, but also during the simulation live status of simulation is shown as it can be seen in Fig. 4. It can be shown the number of packets generated, the number of packets received, the number of packets dropped, the throughput but also the average delay. While simulation the value of packet injection rate can be changed that's why it's easier to determine the critical injection rate. The critical injection rate is playing a major role in causing congestion because as it can be seen in Fig. 5 the average delay is still growing while the injection rate remains the same. The good part of the user interface is that settings for a project can be saved as a project and next time just reload the project that has been saved. This way will be easier for the designer to compare results from different simulations. The log which results after the simulation can be saved as a .pdf format which will enhance reviewing it on different platforms.

4 Application architecture

Typical used design for NoC is mesh so we consider the architecture from Fig. 5 where every router has 7 virtual channels but also buffer for every virtual channel. One of the virtual channel connects the router to the core, while the others are unidirectional and connects two neighbors [4].

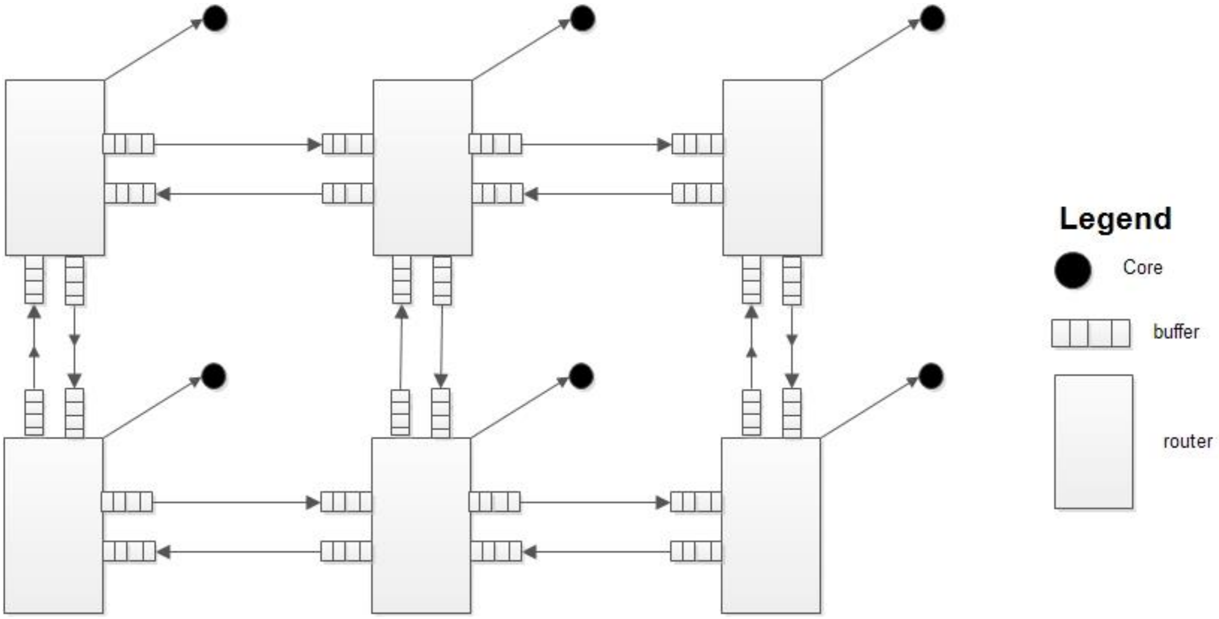


Fig. 5

Every core can generate packets and can consume packets. While the buffer from each virtual channel is there to postpone the packets that arrive to quick to destination and the destination is not available.

5 Conclusions

As a conclusion we would like to mention that NoC is a full diversity theme, that's why there are still things to discover and implement. We thought that having a nice user interface for a NoC simulator will help the designers two decide easier which design is better, what things can be improved, but also what are the constraints of the design.

As NoC arrived to cope with issues of on-chip networking we hope that new solutions will be found for today's problem, but also to get new ideas for tomorrow.

Being aware that this paper is not ended and can be improved, we hope that, in the near future, to implement a better and more useful tool which will make simulation of NoCs much easier. So our thoughts will go on.

Acknowledgement:
This article would not have been written without the support and help of my teacher coordinator: Mircea Vladutiu.

6 References

[1] L.-S. Peh and W. Dally. *Flit-reservation flow control. In High-Performance Computer Architecture. Proceedings. Sixth International on*, 2000

[2] J. Santos, Y. Turner, and G. Janakiraman. *End-to-end congestion control for infiniband*. In Twenty-Second Annual Joint Conference of the IEEE Computer and Communications., volume 2, 2003.

[3] R. Das et al. Aérgia. *Exploiting packet latency slack in on-chip networks*. International Symposium on Computer Architecture (ISCA), 2010.

[4] U. Y. Ogras and R. Marculescu. *Prediction-based flow control for network-on-chip traffic*. DAC-43, 2006.

[5] D. Abts, N. D. Enright Jerger, J. Kim, D. Gibson, and M. H. Lipasti. *Achieving Predictable Performance Through Better Memory Controller Placement in any-core CMPs.* SIGARCH Computer Architecture News, 37(3):451–461, 2009.

[6] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. *The M5 Simulator: Modeling Networked Systems.* In IEEE Micro, ages 52–60, July/August 2006.

[7] MADSEN, J., MAHADEVAN, S., VIRK, K., AND GONZALEZ, M. 2003. *Network-on-chip modeling for system-level multiprocessor simulation.* In Proceedings of the 24th IEEE International Real-Time Systems Symposium (RTSS). IEEE, 82–92.

[8] DALLY, W. J. 1992. *Virtual-channel flow control*. IEEE Trans. Parall. Distrib. Syst. 3, 2 (March) 194–205.

Dumitru VILCEANU
Politehnica University Of Timisoara
Computer Science Department
Vasile Parvan 3, Timisoara
ROMANIA
E-mail: doruvil@yahoo.com

LISTE DES PARTICIPANTS:

CRISTIAN-IONEL ALDULEA
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: alduleacristi@yahoo.com

ANA-MARIA BARTI
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50, Brasov, 500091
Romania
E-mail: barti.anamaria@ymail.com

ALEXANDRU PETRUT BECHERU
University of Craiova
Faculty of Exact Sciences
Department of Computer Science
Str. A.I.Cuza, Nr. 13, Craiova
ROMANIA
E-mail: alex.becheru@inf.ucv.ro

MICHELLE BLACKWOOD
The College Of Wooster
Department of Mathematics and Computer Science
1189 Beall Avenue
Wooster, OH, 44691
U.S.A.
E-mail: mblackwood14@wooster.edu

BOGDAN-ANDREI BLĂNIȚĂ
Transilvania University of Brasov
Departament of Mathematics and Informatics
Str. Iuliu Maniu Nr. 50, Brasov, 500091
ROMANIA
E-mail: bogdybba@gmail.com

ALEXANDRU CRISTIAN BOZDOG
Politehnica University of Timisoara
Department of Computer Science
Pta Victoriei No. 2, 300006 Timisoara
ROMANIA
E-mail: bozdogalex@gmail.com

GEORGI CHAVDAROV
University of Ruse
Department of Informatics and Information Technologies
8, Studentska str., Ruse 7017
BULGARIA
E-mail: powerbsd@abv.bg

OCTAVIAN GEORGE CIMPU

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov, 500091
ROMANIA
E-mail: octa_ge@yahoo.com

LAURA CODREANU

Politehnica University Of Timisoara
Computer Science Department
Vasile Parvan, 3, Timisoara
ROMANIA
E-mail: codreanulaura15@yahoo.com

ALEKSANDRA CUGALJ

University Megatrend of Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Beograd
SERBIA
E-mail: fuzefightforfreedom@yahoo.com

MYKHAILO DOROKHOV

Kharkiv National University of Economics
Department of Information Systems
9A, Prospect Lenina, Kharkiv
UKRAINE
E-mail: michaeldorokhov@gmail.com

IONUT DRAGOMIR

PG University of Ploiesti
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
ROMANIA
E-mail: dragomir.ionutz92@gmail.com

VASILE MARIAN FĂLĂMAŞ

“Lucian Blaga” University of Sibiu
Department of Mathematics and Computer Science
No. 5-7, Dr. Ion Ratiu St., 550012 Sibiu
ROMANIA
E-mail: falamas.vasile@yahoo.com

BOGDAN MIHAI FIERARU

Transilvania University of Brasov
Departament of Mathematics and Informatics
Str.Iuliu Maniu Nr. 50 Brasov , 500091
ROMANIA
E-mail: fierarubogdan2000@yahoo.com

HONCHAROVA MARIIA

H.S. Skovoroda Kharkiv National Pedagogical University
Department of Roman Languages
29, str. Artem, Kharkiv
UKRAINE
E-mail: morganmuse8@gmail.com

SIMEON ILIEV

University of Ruse “Angel Kinchev”
Informatics and information Technology
Ruse ,Str. Strudentska 8
BULGARIA
E-mail: simeon.iliev.bg@gmail.com

MARKO KAROVIC

University Megatrend of Belgrade
Faculty of Computer Science
Bulevar umetnosti 29, N. Beograd
SERBIA
E-mail: gamzi91@yahoo.com

PAUL MERLAS

Technical University of Cluj-Napoca
North University Center
Str.Victoriei, 76
ROMANIA
E-mail: crustamet@gmail.com

KALOYAN MIRONOV

Ruse University
Department of Informatics and Information Technologies
8 Studentska Str., Ruse
BULGARIA
E-mail: mail@kaloian.net

PETRE NISTOR

PG University of Ploiesti
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
ROMANIA
E-mail: nistor_petre@ymail.com

DRAGOȘ OBANCEA

Transilvania University of Braşov
Faculty of Mathematics and Computer Science
Computer Science specializaition
Street Iuliu Maniu, Nr. 50, Braşov
ROMÂNIA
E-mail: obancea_dragos@yahoo.com

ISTVAN BENIMAIN PALLO

„Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No. 7-9, Sibiu, 550012
ROMANIA
Email: eu.stefy@yahoo.com

ANDREEA-OANA PETAC

Ovidius University of Constanța
Department of Mathematics and Computer Science
124, Mamaia Bd., 900527, Constanta
ROMANIA
E-mail: andreea.petac@gmail.com

ANDREEA PETRE

PG University of Ploiesti
Faculty of Letters and Sciences
Computer Science Specialisation
Department of Information Technology,
Mathematics and Physics
Bd. Bucuresti, No. 39, Ploiesti 100680
ROMANIA
E-mail: andreea_petre92@yahoo.com

OCTAVIAN FLORIN PETRESCU

University of Pitesti
Department of Mathematics and Informatics
Str. Targul din Vale nr. 1, Pitesti, Arges
ROMANIA
E-mail: petrescuflrn@yahoo.com

ADRIAN POPA

„Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Rațiu Street No. 7-9, Sibiu, 550012
ROMANIA
E-mail: ad_robotics@yahoo.com

DANIEL GRIGORE ROSCA

"Petru Maior" University of Tg. Mures
Computer Engineering
Str. Nicolae Iorga, nr.1
ROMANIA
E-mail: rosca.daniel.grigore@stud.upm.ro

ALIN ROSU

"Petru Maior" University of Tg. Mures
Computer Engineering
Str. Nicolae Iorga, nr.1
ROMANIA
E-mail: rosu.alin@stud.upm.ro

DORIN ADRIAN RUSU

"Petru Maior" University of Tg. Mures
Computer Engineering
Str. Nicolae Iorga, nr.1
ROMANIA
E-mail: rusu.dorin.adrian@stud.upm.ro

DRAGOŞ SEBEŞTIN

University of Craiova
Department of Informatics
A.I. Cuza nr. 13, CRAIOVA
ROMANIA
E-mail: sebestin.dragos@gmail.com

MIHAI STANCU

„Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Raţiu Street No. 7-9, Sibiu, 550012
ROMANIA
E-mail: stancu_meehigh@yahoo.com

SIMONA – MIHAELA TECŞA

„Lucian Blaga" University of Sibiu
Computer Science
Dr. I. Raţiu Street No. 7-9, Sibiu, 550012
ROMANIA
E-mail: tecsa_simona@yahoo.com

ANA TUBA

University of Belgrade
Faculty of Mathematics
Studentski trg 16, Belgrade
SERBIA
E-mail: anatuba89@gmail.com

DUMITRU VILCEANU

Politehnica University Of Timisoara
Computer Science Department
Vasile Parvan 3, Timisoara
ROMANIA
E-mail: doruvil@yahoo.com

Sponsors (par ordre alphabétique):

Agence Universitaire de la Francophonie

Conseil du comté de Sibiu

EBS Romania – Expectations Beyond Software

iQuest Romania

PAN FOOD

ROPARDO

Top Tech