

Ergonomic Design of the Graphical User Interface of Integrated Software Systems for Business Management

Iliya Mitov, Anani Rizov, Krassimira Ivanova

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
{imitov, a.rizov, kivanova}@math.bas.bg

Metodi Vlachkov

Topic Service, PLC
Sofia, Bulgaria
m.vlachkov@prilsys.bg

Abstract

An overview of common used user interface practices applied in integrated software system for business management based on the current Bulgarian market is presented. An approach for ergonomic design of graphical user interface for such systems - FOI Smart Monitor, aimed to overcome the weak sides of observed practices, is proposed. The successful applications of this approach for building integrated software systems for business management in SMEs is mentioned.

Keywords: GUI; UXD, Integrated Software Systems for Business Management; FOI Smart Monitor; ArMSBuilder

JEL classifications: C88 (Other Computer Software), M15 (IT Managemet)

*Design is not just what it looks like and feels like.
Design is how it works.
Steve Jobs*

Introduction

It is difficult to determine the most important parameters for developing a worthy project for an information system. Along with expectations for speed, stability, a friendly interface and easy support, it is paramount for the system to account for other factors as well: responding to requirements for information content and for efficient management of processes and productivity. Without underestimating the role of other factors, this article explores existing Integrated Software Systems for Business Management (ISSBM) solely in terms of graphical user interfaces supported by the systems.

There are many studies devoted to formulating the rules for design and development of the User Interface, and in particular Graphical User Interface (GUI) (Markus, 1995) (Sollenberger, 2012) (Shishedjiev, 2015). They describe with utmost precision the rules that should be followed in GUI design:

- Emphasizing capabilities;
- Identifying priorities;
- Reaching consensus;
- Correct problems' solving;

- Helping orientation;
- Accelerating decision-making;
- Inspiring revolutionary innovation;

And the requirements that a good user interface has to meet (Martin, 2015):

- should be simple;
- should use unambiguous elements with immutable meaning in time;
- should indicate whether an object is accessible or not;
- should be subject and objected oriented;
- should have context-oriented help information;
- each form should preserve the general idea of the entire program (heritability of characteristics);
- evocative use of color and texture;
- use of typography for creating hierarchy and clarity;

These rules and principles are more or less wishful. Full compliance with them would produce the ideal GUI. Their implementation however depends on several external factors that determine the final look and feel of the user interface, namely:

- to what extent the technological capabilities of the software development environments and used programming interfaces allow compliance with of these rules;
- whether the software manufacturer has relevant knowledge and will to comply with these rules.

Review of screen forms of ISSBM available on the Bulgarian market

Because of the importance the GUI for the quality of software, and of ISSBM in particular, we conducted a special study of user interfaces of integrated software systems for business management found on the Bulgarian market (NIF-02-5:5, 2015). We have studied more than 60 systems including: payroll software, accounting software, ERP systems. The capabilities of 12 products have been thoroughly examined based on their demo versions. Information for the rest of the systems has been gathered from the websites of their manufacturers.

The following characteristics of the software from the studied class were addressed:

- arrangement of information on the screen;
- navigation during work /help (on/offline, context orientation, total volume);
- display / selection of features at work;
- completeness and effective functionality of the interface;
- functionality depending on different characteristics of terminals, with a view to maximizing available technological options;
- using colors (including for expression of semantics);
- default settings, with a recovery option after user intervention;
- methods used for storage and handling of information and types of databases;
- connectivity of fields and content structure of the information;

- labor protection of consumers with regard to the interface;
- data protection regarding the interface.

Based on a wealth of material from screen forms of different software (NIF-02-5:5, 2015) we show some weak features of the user interface design, which we will discuss below.

The overlapping screens effect

The graphical interface launched by Microsoft, is based on the form (window) as the basic unit for organizing and displaying information on the monitor screen. This approach has been adopted by almost all software manufacturers.

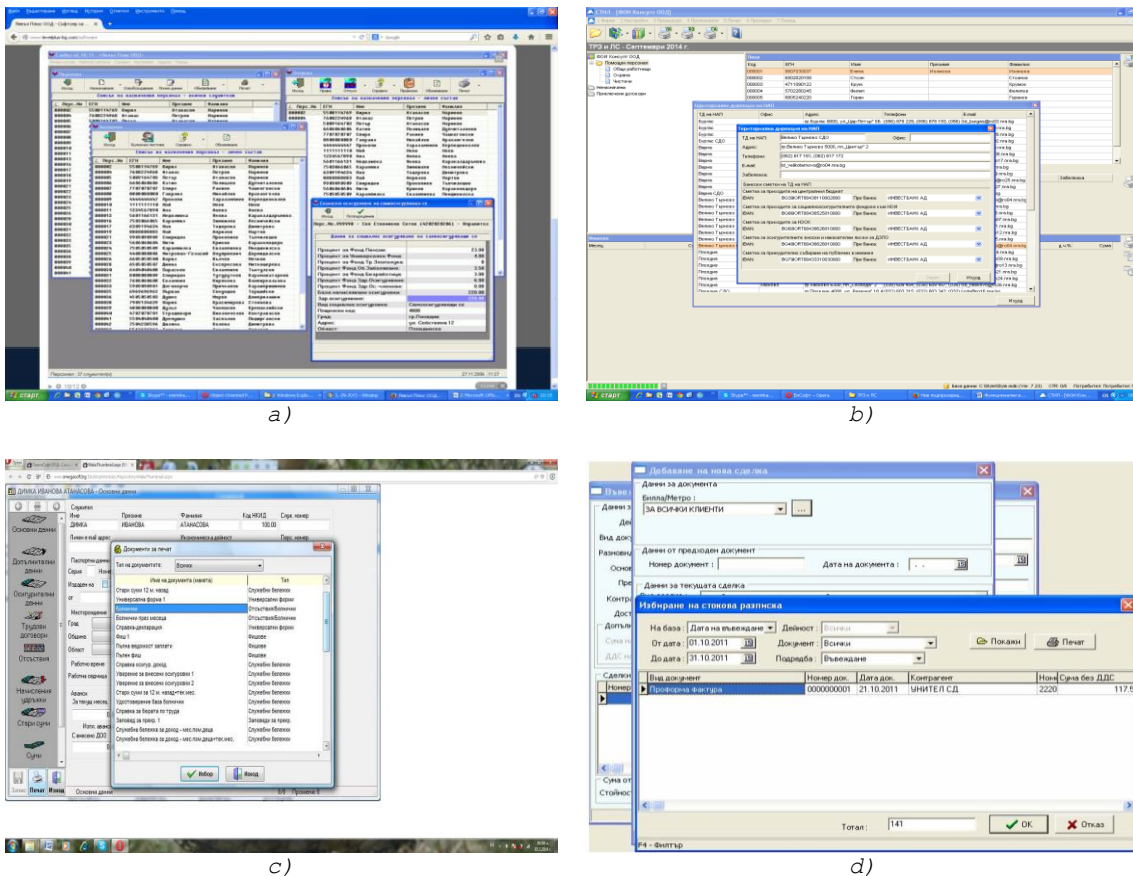


Figure 1: Screenshots of software screen forms displaying the "overlapping screens" effect

The display of an increasing number of screen forms during work is often deemed as highly efficient but we cannot agree that this is a good approach to develop GUI. Criticism against such user interface relates to the following points:

- the monitor screen that the user watches while working with the program displays lots of information from various screen forms (Fig. 1a) which does not assist current data entry. The user has simultaneous eye contact with scores of icons and functional buttons which are irrelevant for work with the active screen form (in some cases it is not immediately clear which form is actually active);

- displayed information is not of great use anyway, because it is more or less hidden by the next screen forms and it is not altogether clear what is actually seen on rear screen forms (Fig. 1b, 1c);
- why - given there is free space on the screen - information is not arranged in such a way so as to make possible to enter data with a single screen form on the entire screen (Fig. 1d), instead of using a great number of small forms?

A more detailed analysis suggests the serious impact the type of database used has on the simultaneous display of a large number of forms on the screen. Each group of relatively related data in relational data bases (RDB) is recorded as a row to a certain table. For work with data from each table the relational model of operation requires the creation of a separate form. When the performed information processes are more complex (described with several tables) we inevitably end up in the situation of the shown screenshots of screen forms overlapping each other. This is not a mandatory GUI layout in this type of software but it is widespread. Its common use is to a certain extent prompted by automated development environments which while assisting the process of code writing, result in such effects. This is not only a telltale example of how programming interface influences the type of GUI, but it also illustrates drawbacks in designing ISSBM, when using standard GUI for RDB.

Icons and/or menus from function buttons

One common solution of GUI in software products is making almost all functionalities of programs visible on screen forms. Such a mode of work violates a few basic rules in building GUI:

- the screen displays a great number of functional elements, a large part of which are irrelevant to main operations performed but they still attract user attention;
- there are many graphic buttons (icons) which often have unclear semantics as graphic symbols. Maybe after long enough work users finally learn the semantics hidden in the graphic buttons but this is hardly the best interface solution;
- current screen forms do not cover icons and buttons of rear forms and the user finds it hard to tell which menus are currently active;
- random positioning of functional elements, which leads to poor structuring of the action sequences.

Displaying information on the screen form

Computer technology is developing very rapidly. Contemporary monitors support a Full HD resolution as a standard and even Ultra HD. This circumstance is not fully regarded when developing GUI and results in:

- **Undervalued graphical capabilities of screens:** most of ISSBM currently in use were made more than ten years ago, when typically screen matrix was of 1280x1024 pixels. As time passed by computer technology developed but programs remained the same and if improved at all this does not seem to have included GUI. The example in Fig. 2a, in which less than 25% of the working area of the full screen is used, characterizes many software products (NIF-02-5:5, 2015). In practice it turns out that there is little benefit for the user working with current ISSBM for management of business operations from the improved quality of computer screens.

- **Overvalued ability for user reception of information from the screens:** resulting once again from the factor of computer technology development and unsatisfactory practical implementation, there is another poor-quality version of GUI (Fig. 2b). Screen forms display a large amount of information - in this case presented with many table rows or same type of text components. When this is coupled with a small font, operators are left with the option of minimizing the resolution of computer screens to be able to work at all. Our research suggests that there is no software product offering manual regulation of font size in screen form texts.
- **No visibility of the whole information:** we can define as historical legacy GUI solutions in which part of the information is not displayed on the screen. Usually this happens when using tables (Fig. 2b). When the columns exceed the screen width we cannot see the whole information, connected with chosen row. In some cases, besides being annoying and difficult to work with, such solutions create conditions for a lot of errors in data entry.

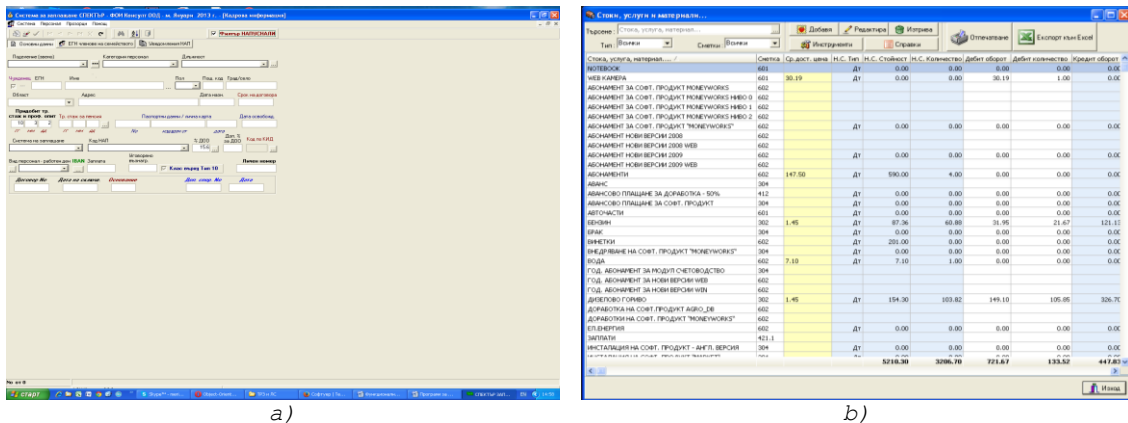


Figure 2: Presenting information on screen forms

Use of typography

Software products that we have surveyed do not typically use typographical elements in building the user interface (colors and fonts). The simplest example is the use of various colors while displaying information or emphasizing important elements of the form. Typically ISSBM are created in neutral colors, because with a richer color options the eyes get tired. We found just one software product in which the user is enabled to alter the colors of the background of forms and functional elements. There is a wealth of options in this regard - almost every element can be loaded with a different color. Such rich options may also generate problems because while users change colors randomly, some unsatisfactory combinations are likely to occur such as the one shown in Fig. 3. The software exerts no control in this regard.

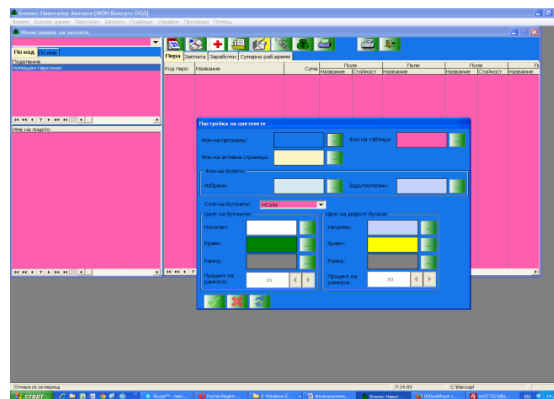


Figure 3: One example for unrestricted change of colors by the user

Models of information panels of screen forms

The analysis of information panels used in software suggests a relative consolidation with a view to the location and the type of information on working forms (Fig. 4). Usually the upper part of the form contains one or several panels arranged in part or across the length of the form in which various functional elements are positioned - buttons, drop-down menus, icons etc. The left side of the form is again reserved for command, functional elements. The most part of the central and right side of the screen form is used for data entry or for getting reference.

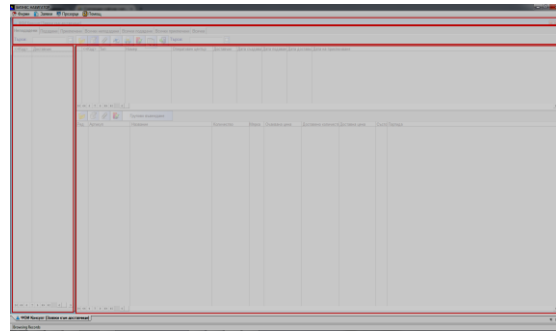


Figure 4: One model of information panels in software forms

Conclusions drawn from conducted research of GUI:

- Availability of a large number of software products (ISSBM) is mostly due to the fact that these products have serious issues and the look and feel of GUI is one of the problems;
- There are issues regarding the training of staff in informatics and computer technology. The observed non-compliance with basic rules in GUI development suggest that these rules are not known or are not implemented;
- The main problem in the development of ISSBM user interface refers to certain thinking patterns arising from the use of the relational model of data processing. This model does not reflect well the information patterns that man uses;
- In surveyed software the user interface is built without taking into account simple ergonomic rules. The software development environment and the type of the database have emerged as factors of primary importance;
- Surveyed ISSBM do not operate well on monitors in a Full/Ultra HD mode. We end up in a situation in which the better computer technology the user has, the less convenience he gets during work.

With this analysis we do not seek to deny the right of existence of information technology used for programming and of design styles implemented by different manufacturers. We share the opinion that each manufacturer is free to decide what his product will look like. We believe that the latest IT and hardware achievements should be reflected in the development of programming and user interfaces of software products for information services of business operations in enterprises.

Principles of Man-Machine Interaction Concerning User Interface and User Experience Design

Fast and accurate output of information is the main task of ISSBM. These systems are intended to serve the information processes in the enterprise, and whether data in them carries the accurate information for the user is of paramount importance. On the other hand, to be able to work well with programs the user should quickly and accurately take

in the information displayed on screen forms. In this regard a well built user interface is of great importance.

In principle, every object displayed on the monitor screen can be considered a datum. When these objects (data) are observed by someone, in some cases they acquire meaning and become information for the observer. It is assumed that the user working with ISSBM is in the know of the subject area and data visualized on the screen like digits, texts and graphs etc. have definite meaning for him. The meaning of data for the observer - what information they deliver to him, is determined by different processes: education, experience etc. We assume that the screen of the computer device visualizes information (because the user comprehends what he sees on the screen form) and it is in this aspect that we are going to use this term further on.

To achieve better performance of GUI we shall widen the area of interaction and shall discuss not only the look and feel of the user interface but will also account for the user experience design. The screens of computer devices represent tools for visualizing information and as such they have to comply with requirements put to all other means of visualizing information - print publications, works of art etc. Based on this we can claim that (NIF-02-5:3, 2015):

- The attributive characteristics of objects that can carry information, but are treated separately from the information, can be categorized by:
 - shape, size and deformity of the image;
 - effective resolution (linear, contrast and color)
- All attributes (objects with a role to carry information) of the interface consist of attributive elements. Attributes as information carriers can have three functions:
 - Target function - direct representation of models that are the target of the current interaction;
 - Management function - providing for management of the interaction, but do not representing target models;
 - Aesthetical (emotional) function - aiming to influence the operator's emotions.
- A given attributive object can be void of any information function, or have one, two or all three functions.

Apart from rules for information visualization, a good quality GUI should be based on correct models of automated activities, so that information communication could proceed properly. In this sense, in designing GUI we have to distinguish clearly and define the meaning of the notions (NIF-02-5:3, 2015):

- Prototype: the represented model, in the memory of the source;
- Representation: the totality of attributes and processes, in the interface, used to represent the original;
- Representer: a model of the original located in the memory of the recipient formed on the basis of conducted information communication.

On the basis of increasing requirements to the content and quality of GUI, and with a view to additional requirements put to the above discussed areas, we can formulate the general rules responsible for

the development of a good GUI. Apart from known rules for building GUI presented in the previous chapter, the development of a good user interface should account for the following aspects of displaying information on screen forms:

- Ergonomic requirements for displaying information in time:
 - displaying information needed at the moment;
 - displaying only the information needed at the moment;
 - no displayed objects which are not carriers of information;
- All attributes and their functions are recognizable by the recipient, without sensory and cognitive problems;
- All attributes and their functions are presented by the sender, without effector and cognitive problems;
- Basic parameters regarding the content of information communication at a certain point of time:
 - simultaneity of representation: how much of the prototype is represented simultaneously;
 - completeness of representation: how much of the prototype is represented to the recipient;
 - representation limits: how much of the prototype is allowed to be represented to the specific recipient;
 - veracity: to what extent the prototype and the representer are equivalent, a possibility to adapt the representation to various characteristics of interfaces and operators;
- Basic parameters regarding the speed of information communication, for a time interval:
 - time for training in representation/reception of information;
 - an option for self/guided training;
 - an option made available for help with quick reminder of forgotten knowledge;
 - an option to adapt the interface to various states and level of preparedness of operators;
 - an option for dynamic invocation (only if necessary) of the tools needed to navigate in the prototype, the model and the representation, as well as for search, analysis and editing of objects.

FSM mechanism

Long-standing experience in developing and implementing ISSBM has given us the chance to be together with users almost on a daily basis and to become well aware of the problems and various inconveniences caused by existing GUI. Aiming to improve work with GUI and to offer to users a more convenient reception of screen forms we have developed a special software mechanism in charge of automatically setting screen forms to make sure that they have maximum visibility on different monitors. We have called this mechanism **FOI Smart Monitor (FSM)**, a title we shall use from now on in the text.

With the development of technology, methods and forms for displaying images have developed as well. From the early, DOS visualization on a screen consisting of 80x25 characters, to present-day full-color visualizations. At present the options of visualization of images on the computer screen look almost infinite.

These ample technological opportunities and their fast implementation in practice set a few problems (tasks) to application software:

- One and the same operating system with one and the same application software are able to operate on computers that have screens (monitors) not only from different generations but also very different in terms of the number and characteristics of horizontal and vertical pixels. With this in place, how shall information - identical in volume and content, be visualized on peripherals with different hardware features?
- Even if we limit the case to only one monitor - it has the capability to work in different modes in terms of resolution, orientation, color characteristics, etc. One and the same image may look different depending on the current monitor mode.
- Last but not least - what will be the psychological reception of one and the same image depending on the quality and the capabilities of the screen on which it is reproduced? What are the volume and the visual perception ability of users depending on the features of output devices?

There are various techniques and capabilities set in software for solving this problem. There are mechanisms for scaling of visual components depending on technological characteristics and the actions of the user. Practically, applying these approaches, the visualization of some elements suffers from some side effects. For instance, image scaling includes rearrangement of command buttons, loss of information and the appearance of control rulers; the scaling affects only part of the working area of the window, not the entire window, causing imbalance and aggravating the perception of information and work with the program, etc.

The implementation of the FSM mechanism in developing GUI of integrated software systems allows avoiding some of these problems. A possible result of its implementation is a qualitatively new GUI:

- The images of screen forms of software products look the same way regardless of the characteristics or parameters of output devices on which they are run;
- The higher the quality of output hardware, the higher the quality of images of the user interface. It is possible to use the full potential of user hardware;
- Attaining good ergonomics of images in screen forms. By using the FSM mechanism software products are able to adapt to the specifics of the computer-human interface and the available hardware. Having in mind that past a certain age most ISSBM users experience eyesight problems, this solution is of great relevance in trying to cater for comfort in using computer technology.

The successful implementation of the FSM mechanism in the process of developing software requires compliance with certain rules in designing and building screen forms.

Organization of the information panels of ISSBM forms

The formal analysis suggests that there is relative uniformity between models. All of them feature an upper (in some cases more than one) command row with command buttons or information material. Usually, there is a panel with functional menus to the left. A substantial part

in the central and right section of the form is reserved to serve as the main working area.

Based on the practical implementation of ISSBM, the experience that we have and using as reference existing psychological research about the reception of screen information we have built a model for the arrangement of information and the functional panels of the screen forms of ISSBM (Fig. 5). The benefits from compliance with the rules of the model in building screen forms are the following:

- the process of designing forms is speeded up because the model of laying out information is clear and there is no need to invent it from scratch every time;
- function buttons are concentrated in suitable places and are more accessible in working with the mouse;
- the user gets accustomed to the pattern of command keys and when the form is replaced he does not waste time to look for them. Compliance with the information template leads to automation of the process of work and saves time during data input;
- by arranging the important information diagonally (upper left - lower right corner) we improve its reception by the user.

Of course, the strict adherence to the model should not be exaggerated. When necessary other options can be used of arranging information on screen forms.

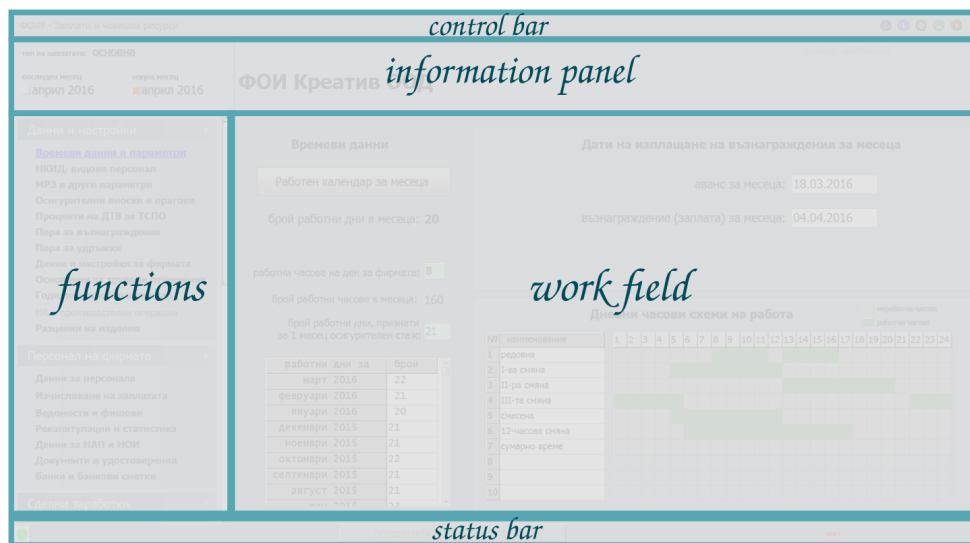


Figure 5: Model of organization of information panels in ISSBM screen forms

By using this model of organization of information and the functionalities in GUI we can:

- **Avoid the overlapping screens effect.** The working area of the form is large and if we use the full screen mode we can display a sufficient amount of information. (Fig. 6). The proposed model for building GUI can be implemented in practice with different program mechanisms. Simultaneous use of extra work forms in implementing the approach to ISSBM development that we describe is needed very seldom;

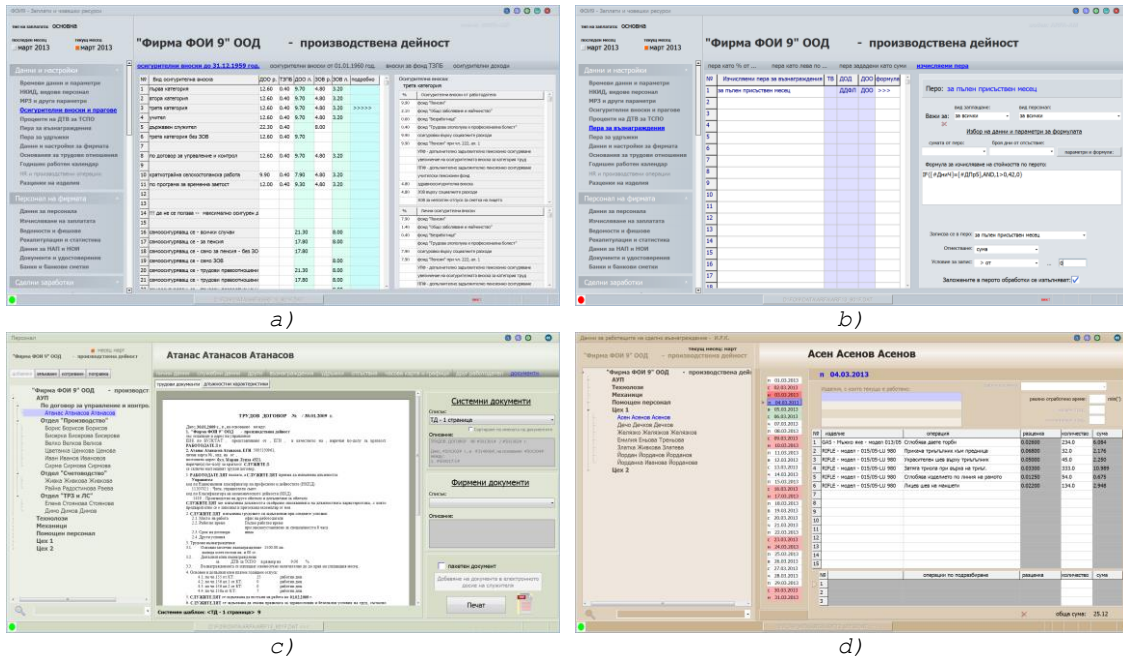


Figure 6: Implementation of GUI using the presented model of organization

- **Use effectively the entire working area of the screen.** With new, high resolution screens this can significantly increase the quality of GUI. In designing the user interface the priority should be a maximum use of the working screen area. This approach has many more benefits than the use of a small area several times. When using the FSM mechanism there is no risk of having the effects as in Fig. 2a;
- **Minimize use of icons.** The great diversity of functionalities accessible by using various graphic images (icons) does not contribute to the user's efficient work:
 - when they are bright the colors of icons affect attention and may lead to omitting important information that the user should enter or report as available;
 - often icons lack in clear semantics and their meaning has to be clarified with text information. This combination occupies space and reduces the actual working size of the basic information model of the form;
 - there is no adopted standard to determine the precise semantics of icons. Icons being graphic images are subject to copyright and cannot be used freely. For this reason different manufacturers use different icons to denote the same functionalities. This is a serious obstacle for users who use different programs in their work;
 - scaling is a problem with icons. When working with screen resolutions for which they are not intended, their use might lead to unexpectedly poor results in GUI. Managing buttons with text in this regard is much easier.

Sparing use of icons with clear semantics spells a good style of building GUI. Consumer attention can easily be drawn by a bright graphic panel displayed in the right moment of work. While in fact the user is likely to ignore a constantly appearing red spot of the form whose functionality should be performed later, during work with the program.

Mechanism for data protection from unintended modification

Users of ISSBM for management of business processes usually work surrounded by various aids: tools, documents, primary accounting documents, personnel data etc. While working with them, they may easily and unintentionally press a key on the keyboard and thus enter information that they not only do not want to enter but are not aware of having entered it. When there is not much data on the screen the user might notice the modification he has unintentionally carried out. When there is much data displayed on the screen (a complex multi-row table) the chance of not noticing the modification is great with all entailing problems in a situation like this. Such cases are common during work with Excel and with ERP systems where data is entered in large table structures.

The risk of such unintentional errors can be avoided by implementing the developed mechanism for protection of data. The user interface can be built in a way so that any time the user enters information he should explicitly declare that (in FSM - by pressing "Enter" or double clicking the mouse). From a formal point of view such a way of work might be seen as an action that burdens the user, but practical experience from implementing this mechanism suggests that it is accepted well and is very effective during work. The implementation of this new mechanism opens up various opportunities for aiding the work of the user and carrying out interim controlled and distributed data processing sessions. Such capabilities cannot be implemented with the old style of work with direct data entry.

This innovative mechanism for data protection from unintended modification is one of the key elements in the proposed new approach to developing ISSBM. With a good programming implementation it is set to deliver ample opportunity for improving work with data in GUI.

GUI: look and feel design

Building GUI does not boil down to simply arranging components on screen forms or at least this should not be the case. One of the reasons for not making good user interfaces is that they are developed by programmers, who are focused exclusively on how to create fields for data without giving a thought to how they should be arranged and what visual aesthetic characteristics these fields should have.

In building the software product (in graphical software environments) part of the job is done in a "look and feel" mode. This mode implements the functionalities of the software project by using visual components. How good the final result will be depends not only on the skills of programmers but also on the art director (look and feel designer) of the product (they should not necessarily be different persons).

The term "look and feel" of a product is commonly understood as artistic and aesthetic design of that product, but there is little awareness that in creating the look and feel of the product many other parameters and characteristics of psycho-physiological, physical, technological and other nature are taken into account.

To create a good design of a particular product, apart from artistic skills, the designer should also know in detail the production technology and the objectives set for the implementation of the

product. This is only possible if the designer is member of the staff of the company that manufactures the product. To corroborate this assumption, we can refer to a statement made by the chief designer of Google. He once said how by increasing the font size in the field he achieved an increase in search engine speed. Look and feel designers are no external staff of companies that can afford keeping such professionals on their payrolls. By the way, Google is the software with the simplest user interface which is not necessarily a drawback.

In practical implementations of software systems we have tried various artistic methods for the improvement of the GUI look and feel and have thus established a couple of rules that give great results and are simple to use.

Using gradient mechanisms for color design of screen forms

Color characteristics of the screen form emerge as a powerful factor to impact on the psychology of users. It is a challenge how to achieve evocative color design given the diversity of forms that can be used in ISSBM. Another challenge is what the limit is of ergonomic color diversity. Here we show the solution we use and which is incorporated in the proposed approach to building ISSBM. The solution refers to using gradients of a certain default color implemented to build the whole screen form and the visual components in it. In fact this is nothing new: artists have been using this technique and it is also welcome to software implementations. The results are quite satisfactory in terms of both artistic merit and ergonomics without any need of specialized and highly paid interference from a design specialist in developing and building GUI. The user can easily modify color design without causing an explosion of colors in the form (as in Fig. 3). Fig. 7 shows different screenshots of the same form on which the mechanism of the gradient fill has been applied to its visual components. It is clearly seen that almost every color is admissible for work.

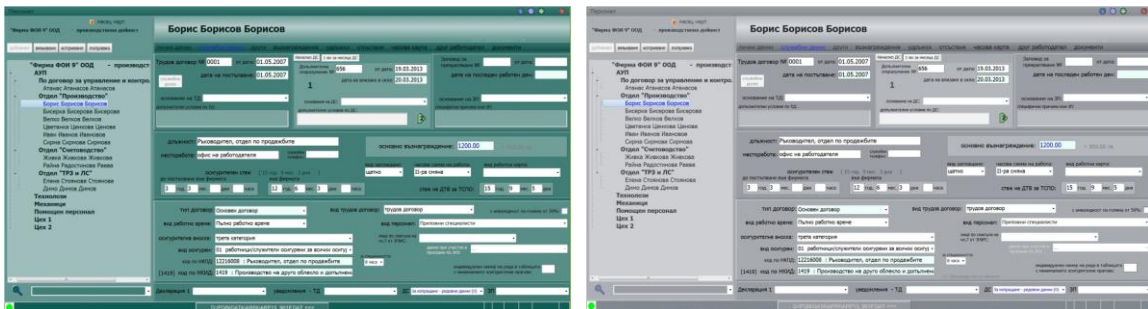


Figure 7: Gradient fills of different default colors applied to one and the same screen form

Using typography to display important information

Texts are among the basic carriers of information. Their information content could be reinforced by using a range of typography techniques – different fonts, size and color of characters, background of the text etc. Regardless of the color, important text information has been emphasized by using a different font, highlighting of the background of the text (technically this is easy to achieve by modifying the parameters of the default color by means of a coefficient) and special visual effects for emphasizing certain texts.

Conclusion

The proposed approach for building the user interface of ISSBM is a part of the complex concept aimed to assure all functionalities of software systems for business management intended for mass use.

The concept for developing complex software systems at the programming and technological levels is implemented in the software environment ArmSBuilder. Using ArmSBuilder enables small teams to develop complex integrated business applications very fast. Programs can run on all kinds of computer configurations, local and global networks. They are low-cost and do not depend on external support programs. These characteristics are of great importance for the implementation of this type of software products in Bulgarian SMEs.

Several ISSBM built with c ArmSBuilder (<http://www.foi9.eu/>) have been implemented and released for exploitation. The existence of these software products in practice corroborates the viability of the proposed innovative approach for developing ISSBM.

Acknowledgements: The research is financed by Bulgarian National Innovation Fund under the project 7IF-02-5/25.07.2014 "A New Approach to Developing Software Systems for Business Management and Control Intended for Mass Use: Concept, Methodology, Software Tools, Experiments and Analysis".

References

- Markus, A., 1995, "Principles of effective visual communication for graphical user interface design", in Human-computer interaction, Morgan Kaufmann Publ., 425-441.
- NIF-02-5:3, 2015, "Visual Characteristics and Possibilities of the Man-Machine Interface", Tech. Report No.3, Project 7IF-02-5/25.07.2014, 2015, 80 p. (in Bulgarian), <http://armsbuilder.foi9.eu/materials/NIF-Survey3.doc>
- NIF-02-5:5, 2015, "Review of the Industrial CRM and ERP Systems in Bulgaria", Tech. Report No.5, Project 7IF-02-5/25.07.2014, 2015, 140 p. (in Bulgarian), <http://armsbuilder.foi9.eu/materials/NIF-Survey5.doc>
- Shishedjiev, B., 2015 (accessed), "Design of User Interface", (in Bulgarian), ff.tu-sofia.bg/~bogi/StopFak/SoftEngr/e7ch16-User_interface.ppt
- Sollenberger, K., 2012, "10 User Interface Design Fundamentals", <http://blog.teamtreehouse.com/10-user-interface-design-fundamentals>
- Stoyanova, S, 2008, "Problems in the Usage of Business Information Systems", Economic Alternatives Journal, Issue 6/2008, 19-29 (in Bulgarian).
- Suzanne, M., 2015 (accessed), "Effective Visual Communication for Graphic User Interfaces", http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html