

Automatic evaluation of information dashboard usability

Jiří Hynek, Tomáš Hruška

Abstract—This paper combines generally known knowledge about information dashboards and tools used for data visualization with findings in the field of visual perception and user interface evaluation, and discusses the idea of a new approach of information dashboard evaluation which predicts a rate of information dashboard usability and comprehensibility. Described approach is based on decomposing the evaluation problem into several views in which different aspects of dashboard usability are analyzed. Emphasis is put on automation of this approach, which can save time and money, on the contrary to testing on real users.

Keywords—information dashboard, automatic evaluation, predictive evaluation, data visualization, visual perception, human computer interaction

I. Introduction

In the field of data visualization, it is often required to present a big amount of data. These data can be used for specifying particular tasks, goals or complex strategies. Therefore, it is important to show these data well-arranged and understandable.

One of the tools used for this purposes is *information dashboard* (simply *dashboard*). This utility is usually a common part of information systems. It helps to summarize data obtained by monitoring of activities which needs to be performed for achieving of particular goal or strategy (e.g. increasing of profit by 10%). The purpose of this tool is to provide a quick view of results or actual state of these activities.

If a dashboard is properly designed, it can help to understand a semantics of visualized values. Then, data can be easily transformed by a user to information and knowledge used for specifying tasks, which need to be done as a response to an actual state of system. On the contrary, improper design of a dashboard can lead to an incomprehension of visualized data and unsuitable tasks specifications.

As described in [6], dashboards are improperly used in a lot of information systems. Presented data are usually misleading and poorly arranged. Detecting such problems is often neglected due to undesirable expenses, lack of time or little awareness of negative impact on a dashboard usability caused by these problems. The tool which would automatically detect dashboard usability problems and inform designer about usability problems is therefore required.

II. Definitions

A. Dashboard

[6] defines a *dashboard* as:

a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.

As the definition says, the dashboard should not exceed boundaries of a single screen, therefore no scroll bars or menus for switching between multiple views should be contained in here. A selection of data which will be visualized by dashboard should be done in a connection with a purpose, for which the dashboard is required. An emphasis is given to the simplicity and graphical visualization of presented data because it can help to better understand meaning of these data.

[5] describes a dashboard as a presentation layer of a business intelligence architecture and distinguishes between three kinds of dashboard, depending on its purpose. These three kinds are strategic, analytical and operational dashboard. They differ from each other by a detail of view. Operational dashboards may show actual raw data, which are frequently refreshed, while analytical and strategic dashboards are rather used for presentation of summarized views or future predictions of some processes which are used for analytical or strategic purposes.

According to a type and semantics of visualized data, appropriate diagrams (graphical parts of dashboard) should be chosen. There exist various types of diagrams [8], but each of them is applicable only to a specific data domain (e.g. pie chart could be suitable for value comparisons, but not for monitoring of value changes in time).

No matter which diagrams are picked, they should be as simple as possible and they should show data accurately. For measuring of these factors, *data-ink ratio* and *lie factor* metrics, presented by [14], can be used:

$$\text{data-ink ratio} = \frac{\text{total ink used to print the graphic}}{\text{data-ink}} \quad (1)$$

$$\text{lie factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}} \quad (2)$$

B. Visual perception principles

Information presented by a correctly designed dashboard should be quickly and correctly perceived. To design such a dashboard, it is necessary to understand principles of visual perception, which relates with the cognitive psychology.

Vision is the dominant human sense. Eyes, as a receptors of vision, detect light and send it as electrical impulses via neurons to a brain, which processes this signal [7]. Brain plays a crucial role in making sense of what is really seen. It recognizes patterns in a processing image and tries to assign meaning to them. Only a fraction of what is seen and is also an object of a focus is registered by the brains attention [6]. This is caused by limited capacity of brains memory.

According to [9], we can distinguish between two types of memories: *short-term* and *long-term* memories.

- The short-term memory stores items of an actual focus of attention. It is volatile and it can store highly limited amount of items (3-5 items [9]).
- The long-term memory can store a higher amount of items which don't need to be focused. Manipulation with these items is however significantly slower.

Items stored in the short-term memory are processed attentively. [15] pointed out, that before that happens, human partially preprocesses image without his attention. It is done *pre-attentively* and very fast in the *iconic memory* [6], which is demonstrated in figure 1.

2460987245187249 2460987245187249
1895278629723678 1895278629723678
1672375323785137 1672375323785137
2509012112523089 2509012112523089

Figure 1. It is easier to count amount of digit 5 in the right image, because different intensities of color are distinguished pre-attentively. Digits in the left image needs to processed sequentially in the short-term memory.

Besides color, there are several another pre-attentive attributes, which are explained in [15] or [6]. These attributes relates with principles of *Gestalt psychology* (described in [7]), which explain rules of recognizing shapes in a picture (clustering visual items into a groups). It is important to take these pre-attentive attributes and principles into consideration because they can significantly decrease time of an image processing by a human. For instance, these principles can be used for choosing a suitable type of diagram (figure 2).

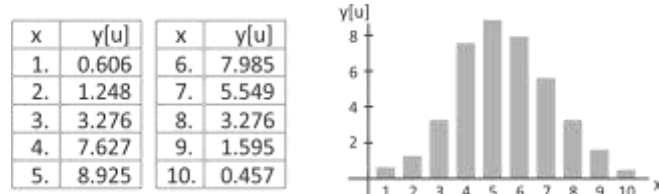


Figure 2. Realitions between values are more clear when they are presented graphically because more preattentive attributes are applied there.

Besides pre-attentive attributes, there are also factors which are not directly related with a data depiction but with an observer. These factors are: *experience of the observer*, *goals of the observer* and *current context* [9].

Recognition of the true meaning of a data depiction can take a long time and sometimes the result of this recognition can be distorted from reality (e.g. an optical illusion [2]). These side effects are undesirable in a data visualization.

THE CHT

Figure 3. Two same symbols are recognized differently as a letter H or a letter A according to current context. Someone who can't read english words probably doesn't distinguish these symbols. Source of image: [9].

C. Usability of user interface

Usability can be considered in different ways. [11] considers usability as a set of five attributes: *learnability*, *efficiency*, *memorability*, *errors* and *satisfaction*. It relates with a whole system, thus usability can be also considered as a kind of software quality (ISO 9126, ISO 25010:2011) [1].

There are several ways to evaluate usability of a user interface. [13] classifies evaluation methods into the four groups – *usability testing*, *field studies*, *predictive evaluation* and the methods of quick user feedback called as “*quick and dirty*” evaluations.

According to [3], evaluation methods differs in a rate of the three attributes – *generalizability* (an extensibility to other people or situations), *precision* and *realism*. For instance, field studies which observe a user in a natural environment can be considered as realistic but not precise (on the contrary of usability testing, which measures user performance of specific tasks). These attributes are influenced by *obtrusiveness* – factor which describes whether (or how much) evaluators affect what is being observed.

Data can be obtained from the evaluation process by several ways. Tested users can be observed without an intrusion (monitoring of their behavior), they can be asked for questions (e.g. dialogues, questionnaires) and they can be measured (e.g. Google Analytics). Their performance can also be predicted without their presence. Experts are however needed. Predictive evaluation requires to simulate a real user behavior, which is a complicated task. Cognitive psychology needs to be taken into a consideration.

A possible automation is advantage of predictive evaluation. It can save time and money. To automatically evaluate user interface, decision algorithms (sometimes called as *heuristics*) are needed to be implemented. These algorithms are based on an expert knowledge, according to which decisions are made. This knowledge can be obtained by an observing of users or from experts. Such base of knowledge can be found for instance in [10, 12].

[4] discusses the user evaluation framework which uses a base of knowledge consisted of set of *ergonomic guidelines*. These guidelines contains restrictions defined by expert, which specify allowed and disabled attributes of visual elements (*widgets*). The restrictions are automatically tested when a designer put or modify widget in a canvas of a user interface. The framework does not specializes on

particular kinds of user interfaces. It could be used as an extension of widget library.

D. Problem definition

Our problem is to decide whether some selected dashboard is usable. Existing solutions have been briefly discussed in the previous subsection. Their disadvantages are:

- either dependency on an opinion of end users,
- or little focus on specific design principles of information dashboards.

Our solution tries to solve the evaluation problem automatically without presence of final users and tries to define as precise as possible function evaluating usability of information dashboards. It is done at the expense of generalizability to other user interface types.

III. Our solution

The principle of our solution is based on decomposition the evaluation function into several sub-functions, whose goal is to analyze dashboards from different perspectives. These dashboard perspectives store different type of information about evaluated dashboard. Therefore, different types of dashboard analyses can be done.

Then, a goal of every dashboard analysis is to take values of particular dashboard attributes and test whether selected values fit particular constraints, which are related to a type of these values. Tested constraints represents dashboard usability requirements:

- requirements based on end users attributes,
- requirements based on principles of a dashboard design and a visual perception.

The first type of usability requirements are different for every dashboard, on the contrary to the second type.

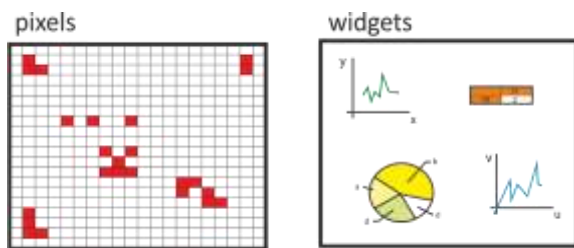


Figure 4. Dashboard can be considered either as matrix of pixels or as a set of arranged widgets presenting data.

Yet, our solution considers two type of dashboard perspectives (Fig. 4):

- a matrix of pixels,
- a set of graphical elements (*widgets*) containing presented data and attributes which determine appearance of these widgets.

These perspectives and possible evaluation approaches considering these perspectives are discussed in the next subsections. Specific evaluation constraints are not mentioned, because they will be the object of our future work.

A. Dashboard as a matrix of pixels

The first perspective represents dashboard as a 2-dimensional matrix of pixels, which contain color values. Size of matrix is defined by a tuple (w, h) , which indicates a matrix resolution. Possible color values are represented by a reduced set of basic colors (e.g. 8-bit color or set of basic named colors).

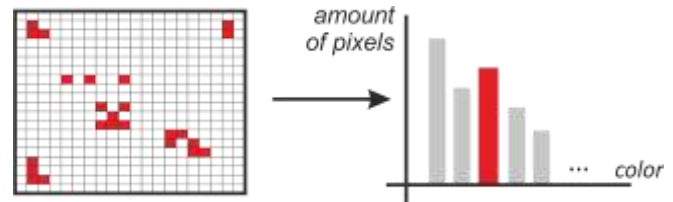


Figure 5. Matrix of pixels is translated to a color histogram for further analyses.

The evaluation is done by analyses of statistical values (e.g. color histograms). Especially, the following kinds of dashboard evaluation are suggested for future research:

- Analyzing which colors and how many of them are used in a dashboard.
- Making of color histograms and measuring frequency of color occurrence.
- Thresholding of image and measuring amount of black pixels and their distribution in the image.

By using this approach, it should be taken into consideration that some matrices can represent random sequence of pixels. That means, that even if results of analyses are satisfactory, the tested matrix of pixels needn't be a correct dashboard. Also, this approach doesn't reflect the principles of human perception (e.g. principles of Gestalt psychology). Therefore this dashboard perspective can be rather used for additional analyses.

B. Dashboard as a set of widgets

The second perspective considers dashboard as a *set of widgets* containing *attributes* and presenting *data*. The evaluation strategy of this approach is to analyze these three aspects separately in three steps. For this purpose, we defined a simple theory which defines three levels of a dashboard description – *model of dashboard*, *dashboard template*, and *realization of dashboard* – which corresponds with the three parts of the dashboard – widgets, attributes, and data.

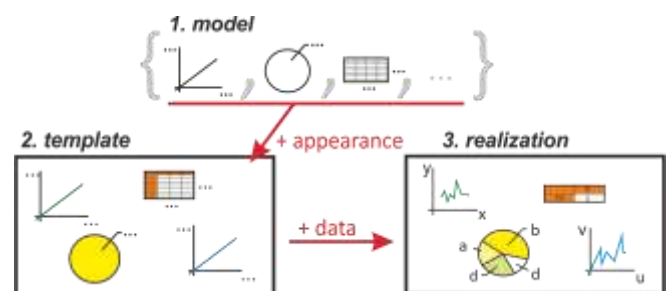


Figure 6. At every level, different aspects of dashboard can be analysed – widgets definitions at model level, appearance of widgets at template level, relation between widgets and data at realization level.

1) Model of dashboard

At the first level, we focus on a set of graphical elements from which a dashboard can be assembled. The set of these elements is called *model of dashboard*.

Definition 3.1: Model of dashboard $M = \{w\}$ is a set of graphical elements (widgets) w , which can be contained by a dashboard. Widget w is defined as a 4-tuple (S_w, B_w, A_w, X_w) , where:

- S_w : shape of widget w . *
- B_w : behavior of widget w . *
- $A_w = \{a_w\}$: finite set of attributes $a_w = (i_{aw}, V_{aw})$ of widget w , where:
 - i_{aw} is unique identifier of attribute a_w ,
 - V_{aw} is finite set of possible values of attribute a_w .
- $X_w = \{x_w\}$: finite set of variables $x_w = (i_{xw}, V_{xw})$, which store presented data by widget w , where:
 - i_{xw} is unique identifier of variable x_w ,
 - V_{xw} is set of possible values of variable x_w .

* For purposes of this paper, S_w and B_w is let undefined. B_w can be considered as an empty set of rules, because behavior of widgets is not considered in this stage of research and S_w can be considered as a list of geometrical shapes, whose quantity, size and appearance depends on real values of attributes and variables which will be specified by dashboard template and realization of dashboard.

Goal of this level is to analyze whether widgets contained in a dashboard model are understandable and suitable for dashboards (e.g. according to [6]: 3D charts are not suitable for data visualization).

2) Dashboard template

The second level is focused on values which are assigned to the attributes of dashboard widgets. For this purposes, *dashboard template* has been defined.

Definition 3.2: Dashboard template $D_M = (M, \{(w, A)\})$, where:

- M is model of dashboard.
- (w, A) represents one object of dashboard template which is specified by:
 - widget $w \in M$,
 - set of object characteristics $A = \{(i_{aw}, v)\}$, where:
 - i_{aw} identifier of widget attribute $a_w \in A_w$ (e. g. *background*),
 - $v \in V_{aw}$ is value of characteristic (e. g. *blue* or *0000FF*).

Dashboard template doesn't contain any informal data which are intended to be presented to an end user. It is used for analyses of dashboard appearance. For instance, following factors can be analyzed:

- number of used widgets and their kinds,
- positions of used widgets (dashboard layout),
- text attributes (size, alignment, font type),
- widget styles (size, used colors).

3) Realization of dashboard

If the dashboard template requirements are satisfied, it is not still guaranteed that dashboard based on this template will help to accomplish desired goals. Dashboard needs to correctly visualize semantics of data. For this purpose *realization of dashboard* has been defined.

Definition 3.3: Realization of dashboard $D_M^i = (M, \{(w, A, X)\})$, where:

- M, w, A has the same meaning as in the definition of dashboard template (definition 3.2).
- (w, A, X) represents one object of dashboard realization, where:
 - $X = \{(i_{xw}, v)\}$ is set of object valuations, where:
 - i_{xw} identifier of widget variable $x_w \in X_w$ (e. g. *x-axis*),
 - $v \in V_{xw}$ is value of variable (e. g. *[1, 2, 3, 4]*).

In this level, dashboard contains data. Therefore, the following new factors can be analyzed:

- relation between widgets and visualized data (suitability of widgets),
- amount of visualized data,
- data-ink ratio and lie factor,
- missing values (e.g. axis names).

In comparison with dashboard template factors, evaluation at this level is more complicated, because semantics of data need to be understood (e.g. analysis of widget suitability for selected kind of data). Users may also need to be familiarized with a context of visualized data.

C. Contributions

Described approach is not dependent on any particular user interface design technology (e.g. programming language, designer tools). Problem is solved theoretically in general way.

However, there is relation between the second dashboard perspective, where dashboard is considered as a set of widgets and tools used for designing of user interfaces. These tools works in a similar way. Person who designs or implements dashboard choose widgets (from some palette of graphical library), then assign attributes to these widgets (defines their appearance) and then connects these widgets with data (which can be represented by some database).

As shown in figure 7, this approach could be mapped to some real technology.

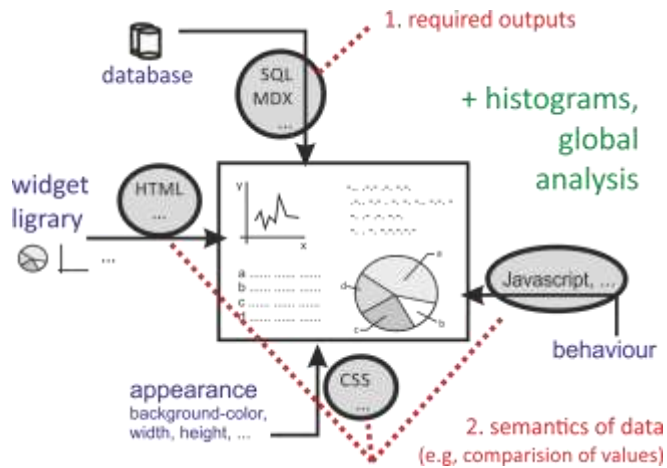


Figure 7. As an example, we can take a dashboard implemented as a single webpage, where structure of page is described via HTML using some widget library, appearance of page is defined by CSS, data are acquired from database using PHP with SQL queries and optional behavior is implemented via Javascript. Then evaluation could be done by analyses of these parts.

D. Limitations

Designer can choose ideal and highly understandable widgets, arrange them suitably in a dashboard canvas and bind them appropriately with data, whose semantics is presented correctly without distortion. Still, this dashboard can be unusable if the dashboard presents information which don't help reach goals, for which the dashboard has been designed. This problem refers to issues of conceptual design of whole system, in which the dashboard is contained. For now, our solution doesn't consider this problem.

IV. Future work

This paper discusses only basic philosophy of the new solution of information dashboard usability evaluation. Solution of the problem has been described by the approach, which considers dashboards in two perspectives, where different kinds of dashboard analyses are automatically performed.

Future work will be primary focused on specifying dashboard requirements based on principles of a dashboard design and visual perception. Then, these requirements will be used for setting of dashboard usability constraints, which will represent base of knowledge used for analyses of the two dashboard perspectives defined in this paper. Then, we will determine whether these analyses are able to be converted into formal algorithms which can be solved in acceptable time.

Acknowledgment

This research was supported by the BUT FIT grant FIT-S-14-2299, "Research and application of advanced methods in ICT".

References

- [1] Abran, A., 2010. *Software metrics and software metrology*. John Wiley & Sons, New Jersey, USA.
- [2] Bach, M., Poloschek C. M., 2006. Optical illusions. *Advances in Clinical Neuroscience and Rehabilitation* 6.2 (2006), pp. 20-21.
- [3] S. Carpendale, 2008. Evaluating information visualizations. In *Information Visualization, Lecture Notes in Computer Science*, Vol. 4950, pp. 19-45.
- [4] Charfi, S., et al., 2014. Widgets Dedicated to User Interface Evaluation. In *International Journal of Human-Computer Interaction*, Vol. 30, No. 5, pp. 408-421.
- [5] Eckerson, W. W., 2010. *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons, Hoboken, USA.
- [6] Few, S., 2006. *Information dashboard design*. O'Reilly Media, Sebastopol, USA.
- [7] Gibson, J. J., 1950. *The perception of the visual world*. The Riverside Press, Cambridge, USA.
- [8] Harris, R. L., 1999. *Information graphics: A comprehensive illustrated reference*. Oxford University Press, New York, USA.
- [9] Johnson, J., 2010. *Designing with the mind in mind: Simple guide to understanding user interface design rules*. Morgan Kaufmann Publishers, Burlington, USA.
- [10] Molich, R., Nielsen, J., 1990. Improving a human-computer dialogue. In *Communications of the ACM*, Vol. 33, No. 3, pp. 338-348.
- [11] Nielsen, J., 1993. *Usability engineering*. Academic Press, San Diego, USA.
- [12] Nielsen, J., 1994. Enhancing the explanatory power of usability heuristics. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Boston, USA, pp. 152-158.
- [13] Sharp, H. et al, 2002. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, New York, USA.
- [14] Tufte, E. R., 2007 *The visual display of quantitative information*. 2nd ed. Graphics Press, Cheshire, USA.
- [15] Ware, C. 2004. *Information visualization: perception for design*. Morgan Kaufmann Publishers, San Francisco, USA.

About Authors:



Jiří Hynek is a student of Ph.D. degree program at Brno University of Technology, Faculty of Information Technology. Subject of his Ph.D. thesis is *Models of Graphic Interfaces*.



Tomáš Hruška graduated at the Brno University of Technology, Czech Republic. Prof. Hruska received his CSc. (Ph.D.) in Computer Science and Engineering from the Brno University of Technology, Czech Republic. He worked in the field of Information Systems and Visualization.