2015 The 5th International Workshop on Computer Science and Engineering
doi: 10.18178/wcse.2015.04.032

Ontology-Based Approach for Engineering Web Interface Design Resolutions

Maxim Bakaev, Tatiana Avdeenko

Department of Economic Informatics, Novosibirsk State Technical University, Russia

Keywords: Kansei Engineering, Human-Computer Interaction, Knowledge Engineering, Recommender System.

Abstract. We propose to develop novel approaches in the field of computer-aided web interface design, which is based on Kansei ("emotional") Engineering that already has recognized results, e.g. in auto-manufacturing industry. The idea is to establish formal relations between key aspects of software requirements and design resolutions, based in particular on experimental research with large sample of existing web projects. As the first step, in the current paper we describe a frame-based ontology that can serve as a base for saving design resolutions, each of which is seen as a combination of interface element, its design characteristic, and its value. The sub-concepts of interface elements and design properties are extracted, in particular, from HTML and CSS specifications and are also presented in the paper. We plan to use the results of the study to enhance the capabilities of the previously developed web interface design support intelligent system. The ultimate goals are to reduce the chances of making erroneous decisions and detrimental trade-offs for designers, as well as to improve the interaction quality for modern web user interfaces.

Introduction

It was estimated that about 50% of all programming code produced when information systems and software applications are built is devoted to user interfaces, the corresponding percentage likely being even higher for web applications. However, until recently, this area was receiving comparably less attention than "true" software development, and engineering approaches to interaction design were relatively scarce. As the result, the interaction quality (often called usability) for software applications and websites remains far from perfect, with even simplest metric such as "user task completion success rate" being at about 80% only [1].

Probably the most straightforward approach to introduce formalism is prior specification of user interface or interaction model (abstract user interface) using a formal modelling language, which then would allow to automatically generate the interface code. However, in perfect accordance with the complex system's formal description elaboration principle, as the complexity of interaction increases, it turns out that the effort required for specification exceeds the one needed for producing an actual user interface. Thus, viable methods and tools in this domain are mostly aimed towards a narrowed range of user tasks or deal with special categories of users (see, e.g., [2] or [3]).

Also, considerable amount of related research is dedicated to measuring usability of interfaces and aspects of their subjective or emotional perception by users, with attempts to identify key factors influencing positive assessments. For example, the following factors for visual appearance of websites could be highlighted (see more detailed review in [4]), by categories:

- **Layout:** balance, homogeneity, equilibrium, symmetry; cohesion, proportion; simplicity, rhythm, economy; density; unity, regularity; sequence.
- **Color:** brightness of the dominant color, brightness of the secondary color, the number of colors, contrast between hues.
- **Graphics:** shares of screen occupied by images, texts, whitespace, etc.
- **Shapes:** regularity (how closely a figure resembles a regular geometric shape); roundness (round, straight or somehow mixed); shape edge (thickness).
- **Text:** amount of text on a page, font styles and sizes, the number of times text is re-aligned, the number of text colors, etc.

However, a major disadvantage is that the above factors are relatively high-level and do not form an integrated system with particular resolutions in web design – even design patterns or guidelines, let alone automatable actions. As the result, this methodology seems to be much more suited for analyzing and validating existing websites, when all kinds of metrics and factors can be calculated (see [5] as an example of large-scale research in this area), but poorly fit to support interface design process. The logical culmination here is almost purely statistical approach, such as in [6] or the infamous [7], which does have its place, but is inherently unable to explain **why** or **how** the particular resolutions at the design stage lead to resulting user experience with the web interface.

Thus, we believe there's a need for more formal engineering approach to describe and possibly recommend design trade-offs made during the course of creating interfaces for websites. In our paper we suggest to start from the infamous Kansei Engineering method, but without viewing it as purely "emotional" engineering. In the following chapters we describe theoretical basis behind the proposed approach, which also uses knowledge-engineering methods, and provide brief description of the actual frame-based ontology we created.

Methods

1.1 Kansei Engineering

The Kansei Engineering method that is used to transform desired customers' feelings and impressions into certain features of the product is widely known since the 1980s, and its application started in Japanese automotive industry [8]. It includes the following principal steps:

1. Creating the list of concepts describing the emotional sphere of potential customers or users of the product and choosing the scale to measure the intensiveness of these factors, e.g. from 1 to 5 or 1 to 7.
2. Developing the general set of a product's characteristics and design-related decisions that can be made regarding them. Usually this is a tree-like or network-like structure, where each possible design resolution is represented as a pair: **category** (e.g. color or size) and **value**.
3. Selecting existing products or their prototypes that will be assessed, and then running the experimental research – generally a survey, when subjects are asked to evaluate the products per emotional scales.
4. Using statistical methods to analyze the obtained data and establish the relations between the emotional scales and the product's characteristics.
5. Or, alternatively, if the relations are already known, steps 3-4 can be replaced by directly specifying the desired customer's Kansei (emotional feeling) and receiving as the outcome the corresponding list of product's characteristics and design resolutions.

Now, Kansei Engineering is mostly famous because it's **Kansei**, but we'd like to highlight that it's an **engineering** method, and the input doesn't have to be subjective impressions – many kinds of software requirements could be successfully used, if their relations with product's characteristics are known and formally described. As said in software engineering, there's an "explosion" of "derived requirements" when moving from requirements to design stages, and the number of these implicit design requirements is up to 50 times higher compared to the number of original requirements [9]. However, website design is no longer an entirely creative process, and there are millions of exiting websites that can be analyzed, so the degree of this industry maturity is gradually approaching the one of auto-manufacturing, with its decades of accumulating customer, usage, and technological data. Thus, we believe that these implicit requirements too could be handled via formalization and AI methods, and even partial success on this path, at least for relatively simple and standard websites, would offer considerable increase in designer's efficiency.

Then, before anything else, it's deemed necessary to specify the website elements and their plausible characteristics and to develop the structure to save design resolutions made regarding them. For that, we propose the application of ontology-based approach, which we describe in subsequent chapters.

1.2 The Ontology Model

In [10], we proposed the specification of ontology (O_F) that is based on Minsky's frame model and incorporates thesaurus T_H – a structured vocabulary of terms corresponding to the domain in one or several languages, with T_{DEF} being the subset of preferred (default) terms for a concept or relation:

$$O_F = <F_C; F_R; F_A; F_E; I_F> \quad (1)$$

where $F_C = <N_C; T_R; a_C; r_C>$ is a set of frames-concepts (correspond to Minsky's frames-prototypes). The frame name $N_C \in T_{DEF}$, i.e. it is the preferred term from thesaurus T_H, while the set T_R contains other terms, in different languages. There are also a set of frame-concept attributes $a_C \subseteq F_A$ and relations with other frames-concepts, $r_C \subseteq F_R$ (since a frame slot's value may be another frame);

$F_R = <N_R; R; a_R>$ – a set of frames-relations linking concepts: N_R is the frame name, R – a set of possible binary relations defined for concepts, $a_R \subseteq F_A$;

$F_A = <N_A; A; TDL>$ – a set of frames-attributes for concepts or relations: N_A is the frame name, A is a set of attributes for classes or relations, $TDL = T \cup D \cup L$, corresponds to a set of attributes types (T = {integer, string, class, instance,…}), data domains ($D = \{D_1, \ldots, D_n\}$) or constraints for the attributes values ($L = \{L_1, \ldots, L_m\}$);

F_E – a set of frames-instances created based on frames-concepts and representing state-dependent knowledge of the domain;

I_F – a set of logical rules establishing semantic correctness of the domain (thus roughly corresponding to axioms generally used in ontologies) or implementing additional logics of a knowledge-based intelligent system as production system.

It should be specially noted that T includes such types as *class* (allowed value is frame-class) and *instance* (allowed value is frame-instance). The latter are quite common in data modelling, but using frame-classes as slot values would allow an important novelty – describing the context of design resolutions with the ontology concepts, not just particular instances.

In the next chapter we present the corresponding part of the ontology which is the base for web design support intelligent system developed by us.

Results – the Web Design Support Ontology

The ontology that finally incorporated more than 150 frames-concepts and 300 frames-instances collected from various sources and reflecting domain knowledge was implemented in Protégé-Frames editor developed by a Stanford University team (http://protege.stanford.edu). Then the intelligent system was built with CLIPS (C Language Integrated Production System), which incorporated the ontology as part of the object-oriented (OO) model, and the prototype of the system was made available for online access at http://clips.vgroup.su.

Of particular interest for the topic of the current paper are the classes that support designer's decision-making in the process of creating a web interface, which is considered as the result of trade-offs that are aimed towards best satisfaction of specified requirements, given technological and other constraints [11]. From user's point of view, website is a set of *services* or chapters, i.e. logically connected web pages, while structurally it is a set of *web pages*. In turn, web pages consist of *blocks* (logically connected areas, such as e.g. authorization form), *elements* (logically diverse units, e.g. main text, header, etc.), and finally *nodes* (that implement ordered sequence of (X)HTML tags). The de-facto standard for describing visual appearance of web pages are CSS, so such classes as *CSS rule*, *CSS declaration* and *Style (CSS) property* were also implemented in the ontology. The resulting structure of classes (see Fig.1) can be used to logically represent web interface prototype and generate wireframe in real HTML and CSS code. The subclasses for classes *Interface element* (extracted in particular from HTML specification) and *Design property* (corresponding to the CSS specification) are shown on Fig. 2.

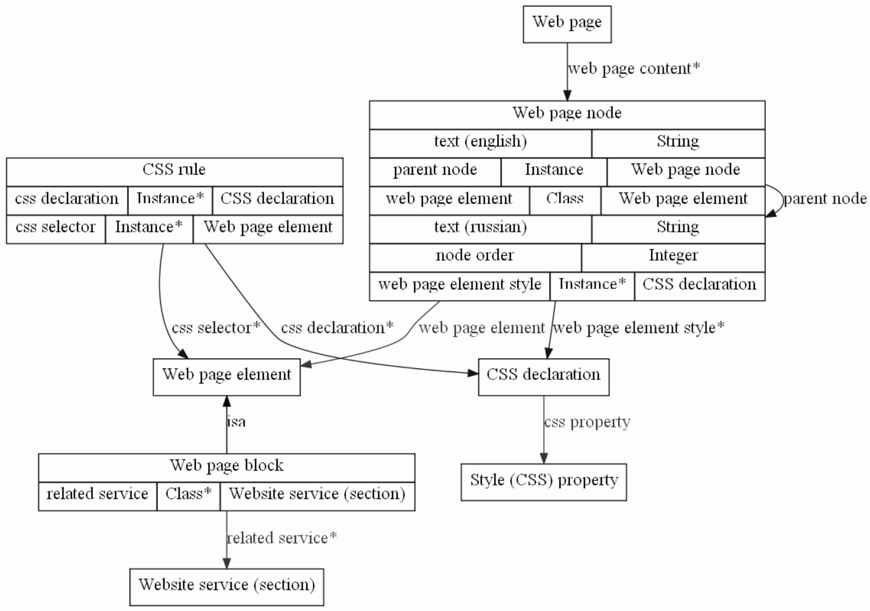

Fig. 1: The structure of classes to represent web interface.

Fig. 2: The structure of *Interface element* and *Design property* classes.

Now, in accordance with Kansei Engineering method, we can construct the *Design resolutions* class, which sets a concrete value via *CSS declaration* for a certain *Design property* of an *Interface element* or *Website element*. For the sake of requirements traceability, there's also relation to *Requirement* class of the ontology; and to allow justification of the resolution, there's relation to *Guideline* class. The resulting structure is presented on Fig. 3, and we also show the *Interface design* class there, which is regarded as a set of design resolutions.

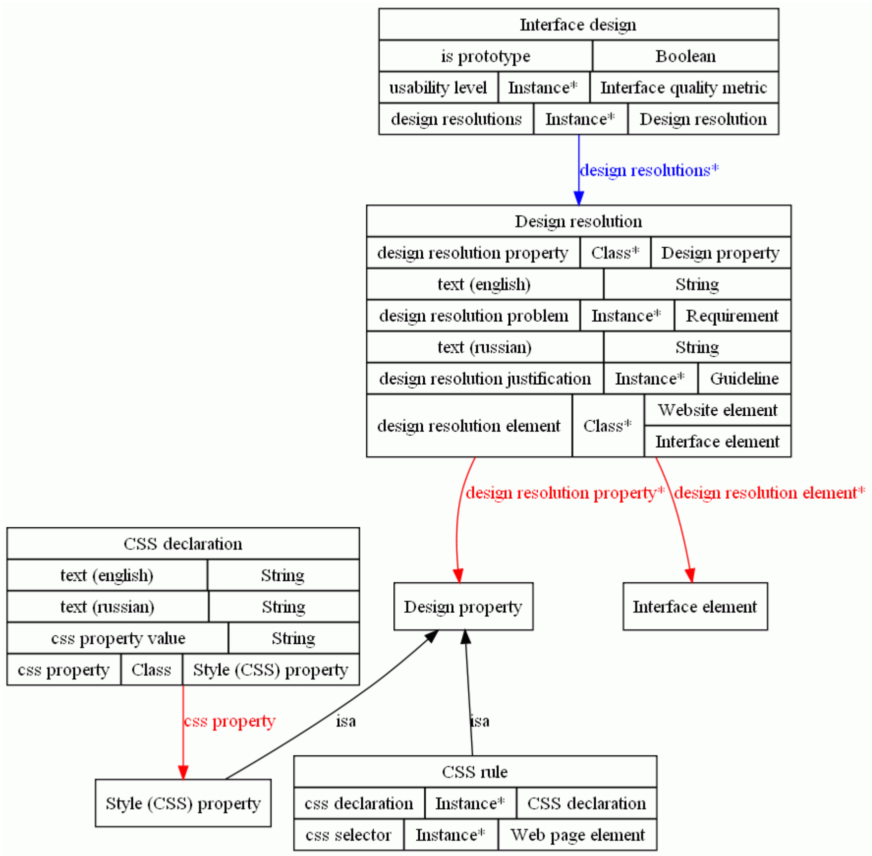

Fig. 3: *Design resolution* class representing Kansei Engineering method.

The usage of *Design resolution* can be illustrated with the following example from one of our projects, a website dedicated to computer courses for senior citizens.

- The value for the slot *design resolution problem* is an instance of *Requirement* class: "Target users for the website are seniors".
- The value for the slot *design resolution element* is ontology class *Main text*.
- Via the slot *design property*, the following values are related: *CSS selector*: "p"; *CSS property*: "font-size"; *CSS property value*: "12 pt".
- The value for the slot *design resolution justification* is an instance of *Guideline* class: "Minimum font size for senior users must be no less than 12 pt [Web Usability for Senior Citizens]".
- So, the value for the slot *text (english)* is "The font size for general texts on the website is 12 pt".

In a similar way the knowledge base of the intelligent system is able to store information on design resolutions for web design projects, so that after accumulating enough statistics the relations between software requirements or target emotional responses and design resolutions could be derived, as per Kansei Engineering method.

Conclusions

The current interaction design and, in particular, web design, lacks engineering approach and the quality of results in this field remains highly dependent on professionals' skills and experience. At the same time, the number of active websites on the Internet has far exceeded 150 million, and we believe the time is coming to introduce more formal methodology into the web design industry. For relatively simple websites we put forward an approach based on Kansei Engineering, which proved itself well in auto-manufacturing, since the 1980s.

As the first step, in the paper we propose the structure of frame-based ontology to incorporate *Design resolutions* as the combination of allowable web interface elements, their characteristics, and values (e.g. Navigation – Font-size – “0.8 em”). The resulting web interface is the set of such design resolutions, each of which may also have relation to initial project *Requirements* and justification in design *Gudelines*. We call the researchers and practicians to the discussion of the approach – while it is true that so far most attempts to formalize the transition from software requirements to design stage were fruitless, we believe as the web design industry matures, the degree of its indescribable creative component diminishes.

Further research prospects logically are in the study of requirements, to identify key aspects that most fully describe the interaction design context, so that they can be transformed into design resolutions for web interfaces. If formal relations can be established, the web interface design support intelligent system that we previously developed can be significantly enhanced, thus contributing to higher quality of interaction on the web in the current multitude of diverse websites.

Acknowledgements

The reported study was partially supported with RFBR, research project No. 15-37-21058 mol_a_ved.

References

[1] Jakob Nielsen's Alertbox, April 16, 2009: IA Task Failures Remain Costly, http://www.nngroup.com/articles/ia-task-failures-remain-costly/

[2] Gajos, K.Z., Weld, D.S. & Wobbrock, J.O., Automatically generating personalized user interfaces with SUPPLE. *Journal of Artificial Intelligence*, 174(12-13), pp. 910-950, 2010.

[3] Nichols, J. & Myers, B.A. *Automatically generating high-quality user interfaces for appliances. Doctoral dissertation*. Carnegie Mellon University, Pittsburgh, PA, 358 p., 2006.

[4] Bakaev, M., Lee, K.H. & Cheng, H.I., The aesthetic and emotional preferences of the elderly and the design factors for e-business web sites. *Proc. of the Eighth Pan-Pacific Conference on Occupational Ergonomics (PPCOE 2007)*, Bangkok, Thailand, 11 p., 2007.

[5] Nathan, R.J. & Yeow, P.H.P., Crucial web usability factors of 36 industries for students: a large-scale empirical study. *Electron Commer Res* 11, pp. 151–180, 2011.

[6] Hearst M.A. & Ivory M.Y. Statistical Profiles of Highly Rated Web Sites. *CHI 2002 Proceedings*, pp. 367–374, 2002.

[7] Nielsen, J. & Tahir, M., *Homepage Usability: 50 Websites Deconstructed*. New Riders Publishing, Indianapolis, 336 p., 2002.

[8] Nagamachi, M., *Kansei/Affective Engineering*. CRC Press, Technology & Engineering, 334 p., 2010.

[9] Glass, R.L. *Facts and Fallacies of Software Engineering*. Addison Wesley, 224 p., 2002.

[10] Bakaev, M. & Avdeenko, T., User Interface Design Guidelines Arrangement in a Recommender System with Frame Ontology. *Lecture Notes in Computer Science*, 7240, Springer, pp. 311-322, 2012.

[11] Howard, S. Trade-off decision making in user interface design. *Behavior and Information Technology*, 16 (2), pp. 98-109, 1997.