Evaluation of User-Subjective Web Interface Similarity with Kansei Engineering-Based ANN

Maxim Bakaev, Vladimir Khvorostov
Economic Informatics department
Novosibirsk State Technical University
Novosibirsk, Russia
{bakaev;xvorostov}@corp.nstu.ru

Sebastian Heil, Martin Gaedke
Distributed and Self-organizing Systems department
Technische Universität Chemnitz
Chemnitz, Germany
{sebastian.heil;martin.gaedke}@informatik.tu-chemnitz.de

***Abstract*— Ensuring similarity of user interfaces (UI) is often desirable, e.g. in software migration and redesign projects, to minimize experience disruption for regular users and increase subjective satisfaction with new versions. In our paper we explore applicability of artificial neural networks (ANNs) to support test-driven development by predicting similarity assessments without employing the actual users. Having reviewed requirements engineering (RE) standards and practices for HCI-related requirements, we identified two dimensions for similarity of web UIs: 1) objective, the data for which we collected with a dedicated web intelligence miner and 2) user-subjective, operationalized with the renowned Kansei Engineering method. Then we constructed the respective ANN models predicting perceived similarity between websites of a same domain and trained the models with the data we collected in experimental sessions with 209 participants of different nationalities and 21 operational university websites. The results of our pilot study suggest that subjective "emotional" factors are considerably more important in predicting similarity evaluations provided by users. Thus, employment of trained ANNs as test oracles may be feasible in automated measurement and control of UI similarity.**

***Index Terms*— re-use, software quality, non-functional requirements, usability evaluation, Kansei Engineering, Neural Networks**

I. Introduction

Nowadays, very few software products are written from scratch, and Software Engineering (SE) experts consistently name re-use among most important advances in terms of increasing programmers' productivity, although its applicability is domain-dependent [1]. Such widespread activities in SE as migration and re-design are effectively also re-using, where the legacy system acts as the single source, supplying pieces of code, architecture, design, etc. In most projects, however, there are multiple sources for re-use, and identification of the ones that are relevant and appropriate to the current requirements is a challenging problem. Still, programming code today is more and more seldom written from scratch, but rather composed from existing components, which indeed allow creating higher quality software with less effort. As both importance and extent of usability engineering activities continue growing, grasping the power of the "developer-oriented" (code and architecture) re-use that used to be in the focus of SE and then using it in the HCI domain appears quite attractive.

It is well-known that testing, debugging, and other activities associated with software quality assurance (QA) constitute the most time- and work-consuming part of the development cycle (see e.g. [1, Fact 31]), and their share in it generally increases together with the scale and complexity of the project. For a long time, academicians and practitioners from SE and related fields have been working on enhancing approaches and tools for more productive generation of programming code, but at some point it became clear that it's QA that holds much more reserves for improvement. We may even argue that limitations on today's software complexity and size (the order of 10^8 lines of code [2]) are due not to hardware productivity, nor even to economy considerations, but to the current advancement of analysis, testing and error removal technologies. So, the power of the "developer-oriented" re-use is not in re-use of code, but in re-use of good quality code, i.e. taking advantage of the time and effort once put into debugging this code. Thus, the "user-oriented" re-use must be capable of identifying verified, good quality cases of interactions that are relevant context-wise.

Re-use of design, which is considered to be even more promising than re-use of code [1] started to attract special interest in SE in the 1990s and at the time was popularly shaped as design guidelines or patterns. However, most design examples and patterns collections so far has been suffering from incompletion, contradictions and general poor organization and were not extensively used by actual developers. A more recent tool, Webzeitgeist design mining/search engine [3], is capable of auto-extracting designs from available web pages, but in terms of indexing and querying it focuses on rather technical, structural and stylistic features. Thus, problem- or user-oriented aspects, more prominent for design re-use sources identification, are not covered, and an enquirer can not specify what the sought designs must do, or for whom.

So, just like for code (and even more so), for designs we observe the challenge with finding relevant solutions (cases), particularly due to ambiguities in specification and verification of HCI-related requirements, of which we'll further speak in the Methods section. Reliance on existing interface solutions seems even more appealing then, since determination of the desired usability metrics through comparison with current software is natural (the next version is usually expected to be a reasonable improvement over the previous one) and can be

978-1-5386-3488-2/17 $31.00 © 2017 IEEE
DOI 10.1109/REW.2017.13

well-supported by user data. Also, it offers the potential to tackle the notorious problem of hidden requirements, as a similarity requirement may substitute a lot of guideline-based requirements and current systems' user studies. We can actually argue that the popularity and efficiency of the A/B Testing are stemming exactly from its "usability by analogy" capability.

Additionally, currently widespread migration and re-design projects demonstrate that users don't like their expectations and built-up experience being violated [4] and exhibit strong preference for familiar interface designs [5] (and for a conventional user, interface essentially equals the system). Thus, not just other things being equal, but even the new version having better objective interaction quality parameters – being more usable, faster, and more novel – it may still seem subjectively inferior to habitual users of the old one. The SUPPLE tool capable of auto-generating user interfaces employing model-driven approach even includes interface dissimilarity metric in the optimized goal function, so that the familiarity of new interfaces to users is enhanced [6]. The proposed metric was quite simple, based on linear combination of factors {0/1} reflecting whether or not the two considered interface widgets are similar, but even such consideration of the preceding version is rare in computer-aided re-design, although it's largely seen as desirable [5]. For example, in trendy adaptive systems the sought user-adaptation has to be balanced with the potential disruption of the accumulated user experience by the changes.

So, in our paper we explore the concept of website similarity as perceived by users with the goal of facilitating re-use of good web design cases. Most likely, this approach wouldn't help much in innovative projects, but for test-driven migrations or re-designs it would be quite practical to have the test oracle capable of telling if new design version passes the similarity threshold. In our work towards this automation, we identified "subjective" (user impressions-related) and "objective" (UI-intrinsic) dimensions of websites and compared their effects on similarity as perceived by users. In Section 2, we analyze the current standards and practices in HCI-related RE and outline the Kansei Engineering-based ANN apparatus for operationalizing the two dimensions. In Section 3, we describe the collection of the objective and subjective data, as well as construction, training and comparison of the ANN models. Finally, we make conclusions that the "emotional" factors had greater effect on user-assessed similarity of websites, mention limitations of our research and outline the prospects for further work.

II. Methods and Related Work

The similarity calculation apparatus had been well developed even before the WWW Era [7] – for example, within case-based reasoning approach, where case closely corresponds to design pattern, – but the main challenge is identifying the set of important features (factors), which is of course domain-specific. We are so far unaware of profound works that specify the features for web design, although web content similarity analysis is growing in popularity (see e.g. distance calculation algorithms in [8]). There is also a solid body of relevant research in recommender systems, where advanced similarity measurement methods rely not just on user preferences, but also on additional information (context) [9]. In any case, similarity assessment implies identifying the set of important features and the ways for obtaining their concrete values. Thus, let us examine how the HCI-related requirements can be specified and their compliance evaluated.

A. HCI-Related Requirements Specification and Testing

Although HCI-related aspects made their way into SE standards quite a long ago (commonly under the name of non-functional usability requirements), there's still little practical coherence on their elicitation and specification. Surely, today everyone emphasizes the importance for a software or a website to be usable, and many provide recommendations on arranging a proper usability engineering process exercising such respectable techniques as task analysis, early and frequent prototyping, usability testing, etc. Still, in some cases formal specification of such requirements, complete with the target usability metrics, is necessary – particularly, when some kind of rigorous testing for compliance is involved, such as in looking for software vendors, outsourcing the development, or performing AI-based usability evaluation. Let us consider what foundations could requirements engineers employ when they have to go deeper than the notorious "the system shall be easy to use".

A key standard for the HCI industry, ISO 9241-210:2010, *Human-centred design for interactive systems*, which replaced the outdated ISO 13407, rightfully claims the necessity to base design upon user-centered evaluations. However, it basically leaves out how exactly the effectiveness, efficiency and satisfaction should be measured or how to determine the appropriate levels for them in the requirements specification process. Another well-established standard in the field is ISO 14915-1:2002, *Software ergonomics for multimedia user interfaces*, the focus of which is multimedia presentation issues. It implies frequent user testing, but doesn't prescribe measurable factors for user interface quality requirements either.

A much more recent instance is the nascent standards of the ISO/IEC 250nn series, *System and Software Quality Requirements and Evaluation*. They are replacing the ISO/IEC 9126 and ISO/IEC 14598, which were rightfully criticized for ambiguity of characteristics and sub-characteristic, weaknesses in usability-related aspects of the quality model and imprecision of the metrics (see in [10]). The Quality-in-use in the ISO/IEC 25010:2011, *System and software quality models*, is supplemented with Freedom from Risk and Context Coverage, in addition to the three traditional usability factors of Effectiveness, Efficiency and Satisfaction. The actual metrics are now specified in the pioneering ISO/IEC 25022:2016, *Measurement of quality in use*, but certain ambiguity remains for some of them: e.g. "self-explanatory user interface" in the adjacent ISO/IEC 25023 sounds as vain as "easy to use" (and if one tries to measure the share of casual users who reads the manual before starting to use the interface, 0% is always a sure guess). The set of usability metrics is based on ISO 9241-110:2006 (Dialogue principles), but "in principle the evaluation of almost any usability guideline (of which there are hundreds in the literature) could be treated as a measure" [11]. This hints at severity of the completeness problem for usability requirements: even if 100 guideline-based metrics are included in the

requirements specification, another 1000s of quite reasonable guidelines are left out as hidden (latent) usability requirements.

As for HCI-related RE, the ISO/IEC 25030:2007, *Quality requirements*, although being the only ISO standard dedicated to specifying the system/software quality requirements, does not quite accomplish its main objective, as it doesn't offer a process effective for real projects [12]. The examples of quality-in-use requirements provided in the Table 1 of the ISO/IEC 25010 are non-illustrative, rather tautological and, according to the standard, are just given as starting points. In the dedicated standard on RE, ISO/IEC/IEEE 29148-2011, the sub-chapters on usability requirements (9.4.5 and 9.5.12) are probably the shortest of all, basically saying "thou shalt define usability requirements". Other HCI-related items there merely provide advices on considering user characteristics (9.5.5) or specifying user interfaces (9.5.3.2), such as making "a list of do's and don'ts on how the system will appear to the user".

Turning to practical experience in the industry, one can discover that the number of related works is quite small. A comprehensive and illustrative classification of usability requirements was proposed in [13] about 20 years ago, and despite that numerous new ISO/IEC standards have emerged since, supposedly evidencing progress in the field, we'd like to summarize the six styles in Table I. It should be noted though that the defect-based style and, to some extent, the subjective style are rather subsets of the performance-based one. The standards that we considered previously, by and large prescribe performance-based approach (e.g. see the Table 1 in the ISO/IEC 25010), but [13] recommends mixing the styles to improve traceability, verification and completeness of requirements.

B. Dimensions of Website Similarity

In our own feature engineering for web designs, we decided to start from the notorious model-based approach, within which the common models can be divided into: 1) requirements-related models: Tasks and Domain, 2) interface models: Abstract UI, Concrete UI, and Final UI, and 3) context of use models: User, Platform, and Environment. However, for the purposes of website similarity assessment it didn't appear to be fully suitable – for example, in software migration projects modifications would be done mostly in the Platform, but we'd be lacking the detail to judge how exactly user experience with the interface changes. The generated Final UI may feel pretty much the same, since interaction is performed through the same web browser and with the same HTML-defined elements, although the model may have changed considerably. On the contrary, in case of legacy desktop information systems migration, the changed model would lead to the set of completely different interface elements, while the processes and the forms remain pretty much the same. So, we propose to identify the following dimensions in UI similarity:

- Task – similarity in the user's workflows, i.e. how much different a user achieves the same task in different versions of a UI.
- Behavior – similarity in the way user interacts with the interface, i.e. the types of input, gestures, assisting functionality.
- Thesaurus – similarity in the building blocks of UIs, in particular in textual vocabulary (language) used in labels and descriptions, as well as in the visual vocabulary such as icons, images, visual representation of data and controls, etc.
- Layout – similarity in the order of elements, orientation or density [5].
- Material – similarity in the basic constituents of UIs, such as buttons, checkboxes, text fields, etc. (defined by the platform specification).

TABLE I. USABILITY REQUIREMENTS SPECIFICATION STYLES (ADAPTED FROM [13]).

Style	*Example*	*Pros*	*Cons*
Performance-based: user group + task + performance.	"Of customers without previous ATM experience, 90% must be able to withdraw a preset amount of cash within 4 minutes."	• Covers most objective usability factors well (for the selected tasks).	• Doesn't cover system-wide aspects and hidden requirements. • Need to collect lots of testing data.
Defect-based: limit on the number and severity of usability problems.	"For novice users performing the money withdrawal task: at most 0.2 failures per user."	• Provides insight to developers, points to concrete problems.	• The limits are hard to elicit. • Most usability factors are not covered.
Subjective: set of criteria for satisfaction with the system.	"80% of customers having tried the ATM at least once must find the system pleasant and helpful. 60% must recommend it to friends if asked."	• Close connection to marketing goals. • Covers user experience on the system-wide level.	• Needs appropriate questionnaire. • Correlation to the actual performance may be weak. • Hard to relate to concrete usability problems.
Process-based: specification of the design process, not the end results.	"During design, a sequence of 3 prototypes has to be made. Each prototype must be usability tested and the most important defects corrected."	• Easy to verify from knowing the development process. • Predictable time and work effort.	• Unpredictable results – critically depend on the development team skills. • Doesn't allow comparison between versions in re-design.
Design-based: specifies interface prototypes.	"The system shall use the screen pictures shown in App. xx."	• The requirements are easy to verify and trace.	• Mixes requirements (the what) with design (the how). • Final results unpredictable.
Guideline-based: compliance to certain guides and standards.	"All dialogue boxes must be non-modal so that users can look at other windows while responding to the dialogue box."	• Re-uses existing knowledge in the HCI field and experience from previous projects. • Easier to cover hidden/latent requirements.	• Appropriate set of guidelines for a specific domain and user group must be formed. • Verification isn't easy and involves interpretation.

In our current work we explored visual similarity of websites, which correspond mostly to the Thesaurus and Layout. The actual interaction with the websites was eliminated, so that the Task and Behavior dimensions were not affected; while the Material remained fixed as defined by the HTML specification. The subjective impressions of the users were operationalized with the use of Kansei Engineering method.

C. Kansei Engineering and ANNs

Kansei Engineering (KE) is a set of methods and techniques relating customers' feelings and impressions with existing or prospective products or their certain features. In case of already existing products, its "analytical" method includes the following principal steps:

1. Creating the list of concepts describing the relevant emotional dimensions for customers using the product – Kansei words, usually a dozen or two.
2. Developing the design space, i.e. a set of attributes and design resolutions related to the product – usually, a tree-like or network-like structure.
3. Selecting products or their prototypes for assessment, then running the experimental research – generally a survey, when customer representatives evaluate the artifacts per the Kansei words (scales), e.g. from 1 to 5, or from -3 to +3.
4. Using formal methods to analyze the obtained data and model the relations be-tween the Kansei dimensions and the products' attributes.

The synthetic method of KE is obtaining the list of the prospective product's attributes and design resolutions from the target impressions in customers – the de-facto emotional requirements specification of the desired product. A recent and quite robust review of KE applications in website construction can be found in [14]; however the authors conclude that they are still relatively scarce. Proposals to build KE knowledge bases in web design domain have been made repeatedly (e.g. [15]), but seemingly none of them were widely used by web engineers in practice so far.

Although KE is often called "emotional" engineering, in fact it can also incorporate physical and cognitive dimensions of customers' interaction with a product, e.g. by employing Kansei words like "slow" or "complicated". However, KE employs the subjective evaluation method to measure customers' emotions not due to its accuracy or robustness, but because it currently remains the most practical approach to detect complex feelings in humans. Still, there's no reason why KE-based models can't be supplemented with ratio-scale objective factors, if they can presumably contribute to better specification of the desired product in KE's synthetic method.

In building such an extended model, ANNs would appear a robust and natural apparatus, as this has long been a popular method in KE (e.g. see [14], [15]). Neural networks also have long history in automated verification of requirements and quality assessment: predicting defects [16] or the resulting quality plus costs [17], even explicitly acting as test oracle [18]. Thus, KE-based ANN seems capable of becoming the mixed or the "7-th" style in non-functional requirements specification for websites, suitable for our similarity-detection and reuse-support purposes. ANNs are first trained (in this, diversity of input data is essential [18]) and then tested on real data, attempting to generalize the obtained knowledge in classification, prediction, decision-making, etc. The available dataset is generally partitioned into training, testing, and holdout samples, where the latter is used to assess the constructed network – estimate the predictive ability of the model. The network performance is estimated via percentage of incorrect predictions (for categorical outputs) or relative error that is calculated as sum-of-squares relative to the mean model (the "null" hypothesis). So, to explore the advantages of utilizing objective or subjective factors, or the two groups in combination, we need to pick real data and properly design the structure for each of the respective ANN models.

III. Data Collection and the ANN models

The research material was university websites, selected by hand with the requirements that: 1) the website has an English version that is not radically different from the native language version; 2) the website has information about a Master program in Computer Science; and 3) the university is not too well-known, so that its reputation doesn't bias the subjective impressions. In total there were 11 websites of German universities and 10 of Russian ones, so that their designs (in terms of layout, colors, images, etc.) were sufficiently diverse in each group. Note that in our current work we are going to equate "subjective" to "emotional" and relate factors of website responsiveness, complexity, etc., to objective factors.

A. Mining Objective Website Metrics

To wholly extract website-intrinsic objective factors, we would have to perform metric-based design evaluation, which generally involves quite advanced processing and structuring of the collected data – website code and probably even content. Instead, for our pilot study we decided to identify only some potentially representative objective factors related to user experience with websites and their quality-in-use. Further, we decided to collect the respective data not from the actual websites, but to exploit the capabilities of global web metric services, relying on their sophisticated algorithms and long history of the web monitoring.

Thus, we developed the "Web Intelligence" data miner capable of collecting data about a specified website, structuring, and keeping them in the database. The structure of the miner corresponds to the model-view-controller pattern; the diagram of the classes, together with some other supplementary material, is available at http://webmining.khvorostov.ru/docs.zip. Particularly, using the miner we were able to extract the following data for the 21 involved university websites:

- Flesch-Kincaid Grade Level – complexity metric, collected from a respective service, https://readability-score.com (range 2.1~11.9, mean=8.84, SD=1.83);
- Number of website sections at top level – complexity metric, collected via processing the actual website, but needs human verification (range 4~13, mean=6.5, SD=1.95);

- Number of errors and number of warnings – design-intrinsic metrics, collected from the W3C code validator, https://validator.w3.org (errors: range 2~56, mean=18.5, SD=12.5; warnings: range 0~28, mean=5.10, SD=5.01);
- Page load time – physical and quality-in-use metric, collected from Alexa.com global service (range 0.41~2.58 s, mean=1.17 s, SD=0.52 s);
- Popularity rank – quality-in-use metric, collected as Alexa.com's Global Rank (range 22865~175059);
- Bounce rate – quality-in-use metric, also collected from Alexa.com (range 32.1%~53.1%, mean=39.7%, SD=4.71%).

For more detailed description of the miner and the validation of the collected data accuracy, see our previous work [19].

B. Subjective Impressions and Similarity

To obtain the subjective impressions and perceived similarity evaluations for training the ANNs, we ran survey with human evaluators that consisted of two parts: 1) individual websites' Kansei assessment and 2) pair-wise websites similarity assessment, the interval between them being several months. The participants in the two sessions were different, so that we could minimize the effect of individual subject preferences and test if subjective factors' (Kansei) effects on similarity could be generalized. The subjects were students (mostly of Computer Science) or staff members of two technical universities: from Russia (Novosibirsk State Technical University) and Germany (TU Chemnitz). The details of the two subject groups are presented in Table II.

TABLE II. THE SUBJECTS IN THE TWO EXPERIMENTAL SESSIONS

		Session 1 (Kansei)	*Session 2 (Similarity)*
Total number of subjects:		**82**	**127**
Gender	Male	80.5%	59.1%
	Female	19.5%	40.9%
Affiliation	Russia	51.2%	78.8%
	Germany	48.8%	21.3%
Age	Range	19~33	17~31
	Mean (SD)	23.1 (2.39)	20.9 (2.45)

To access the websites, the participants used diverse equipment and software: desktops with varying screen resolutions, mobile devices, web browsers, etc., to better represent the real context of use. Before the sessions, informed consent was obtained from each participant, and afterwards they could submit comments to their evaluations. To conduct the sessions, we developed the dedicated survey software, currently available at http://ks.khvorostov.ru. More details about the experimental procedure and the employed websites can be found in our relevant technical report [20].

1) Kansei Evaluation (first session)

In the first session, each participant was asked to evaluate 10 websites, randomly selected from the 21 and presented in random order. According to the scenario given to the participants, their friend was considering enrolling for a Master in Computer Science program in one of the universities, being yet not concerned with the program's content or educational fee. The subjects were asked to browse each website for a few minutes, find the information about the Master program, and evaluate their impressions of the website.

Our Kansei words were defined based on several related research works, including [14] and especially [21], in which the authors applied the KE method to university websites, and the five possible evaluations for each scale ranged from -2 to +2. The total number of Kansei evaluations recorded in the survey software's database from the 82 participants was 13991, and the aggregated data are presented in Table III. Significant (α=.07) differences between subjective evaluations for participants of the two different affiliations are marked in bold and the respective p-values are shown.

TABLE III. MEAN (SD) KANSEI EVALUATIONS FOR SUBJECTS OF DIFFERENT AFFILIATIONS

ID	*Kansei scale*	*German subjects*	*Russian subjects*	*Difference*
K1MF	masculine – feminine	-0.15 (0.37)	-0.36 (0.38)	
K2CC	conventional – creative	-0.03 (0.68)	-0.16 (0.70)	
K3HG	homely – global	0.25 (0.30)	0.40 (0.39)	
K4RP	reasonable – premium	0.02 (0.42)	-0.18 (0.71)	
K5AP	academic – practical	**-0.12** (0.35)	**0.10** (0.30)	p=.069
K6HP	handcrafted – professional	0.13 (0.47)	0.43 (0.66)	
K7NT	natural – technical	**0.19** (0.27)	**0.64** (0.42)	p=.001
K8SD	stable – dynamic	-0.03 (0.61)	-0.08 (0.73)	
K9EA	exclusive – attainable	**0.30** (0.21)	**0.53** (0.34)	p=.039
K10BT	bright – temperate	-0.21 (0.28)	0.04 (0.71)	

2) Similarity Evaluation (second session)

In the second session, 45 distinct pairs composed from 10 randomly selected websites were assessed by each subject in respect to perceived similarity. The participants were assigned no concrete tasks – they were presented the pair of screenshots linked to the actual websites and asked to open and browse the two homepages for a few seconds. The five possible similarity evaluations ranged from 0 (very dissimilar) to 4 (very similar).

In total, the 127 users (see Table I) submitted 5715 similarity evaluations, so for each of 210 possible website pairs the average number of evaluations was 27.2. The resulting subjective similarity values averaged per website pair ranged from 0.296 to 2.909, mean=1.524, SD=0.448. The distance measure for each input factor (F_i) was calculated as square of the difference in its values for the two websites in the pair:

$$Dist(F_i) = [F_i(website_j) - F_i(website_k)]^2, i = \overline{1,17}; j = \overline{1,21}; k = \overline{1,21}$$

C. The Neural Network Models

While the KE method was used for "website identification", the ANNs allowed us to compare the effect of the two groups of factors on subjective website similarity. The ANN apparatus is reasonably well developed, and the performance of a model

is commonly determined from the relative error (RErr) of the holdout sub-dataset, that is calculated as sum-of-squares relative to the mean model (the "null" hypothesis). In each model, the single output neuron was the similarity evaluation, and the input neurons varied according to the nature of the model:

1) **Objective model:** the 7 input neurons were the distances for the 7 objective factors collected by the web mining script;

2) **Subjective model:** the 10 input neurons were the distances for the 10 Kansei scales;

3) **Joint model:** the 17 input neurons were the objective and the subjective impression factors (F_i) put together.

To construct the ANN models, we used Multilayer Perceptron method with Gradient descent optimization algorithm in SPSS statistical software. The partitions of the datasets (210 pair-wise similarity values) in each of the three models were specified as 65% (training) – 20% (testing) – 15% (holdout). The number of neurons in the single hidden layer was set to be selected automatically, and amounted to 4 neurons in the objective, 6 in the subjective, and 5 in the joint model. Table IV presents the details of the resulting ANN models, and in Fig. 1 we show the predicted-by-observed charts.

TABLE IV. THE RELATIVE ERROR (RERR) AND OTHER DETAILS FOR THE THREE NN MODELS

	Objective model		*Subjective model*		*Joint model*	
	No. of cases	*RErr*	*No. of cases*	*RErr*	*No. of cases*	*RErr*
Training	144 (68.6%)	0.903	138 (65.7%)	0.554	130 (61.9%)	0.198
Testing	36 (17.1%)	0.793	48 (22.9%)	0.572	46 (21.9%)	0.581
Holdout	30 (14.3%)	**0.844**	24 (11.4%)	**0.559**	34 (16.2%)	**0.588**

It should be noted that the joint model, compared to the subjective one, showed very good RErr for the training dataset, but poor RErr for training and holdout dataset, from which fact we should conclude that the joint model suffers from overtraining. To assess whether the objective factors contributed to the joint model performance, we carried out the factor importance analysis, whose results are presented in Table V.

IV. CONCLUSIONS

To avoid losing existing customers due to problems related to changed user interfaces and interaction, particularly when migrating or re-designing software, it is important to measure and control the similarity of user interfaces. Measuring similarity of user interfaces is not trivial and deserves close examination, but this issue has been somehow neglected, especially for websites. In our research work we constructed and trained ANN models with the input neurons based on: 1) "objective" factors, collected for the considered websites using dedicated website data collection script, and 2) on "emotional" scales composed per Kansei Engineering method. The RErr for the second model (0.559) turned out to be considerably better than RErr for the first one (0.844), which suggests that subjective factors were better suited to predict website similarity as perceived by users. This conclusion is supplemented by analysis of the factors importance in the joint model (incorporating both groups of factors), where 9 out of top 10 most important factors turned out to be subjective – the Kansei scales (see Table V).

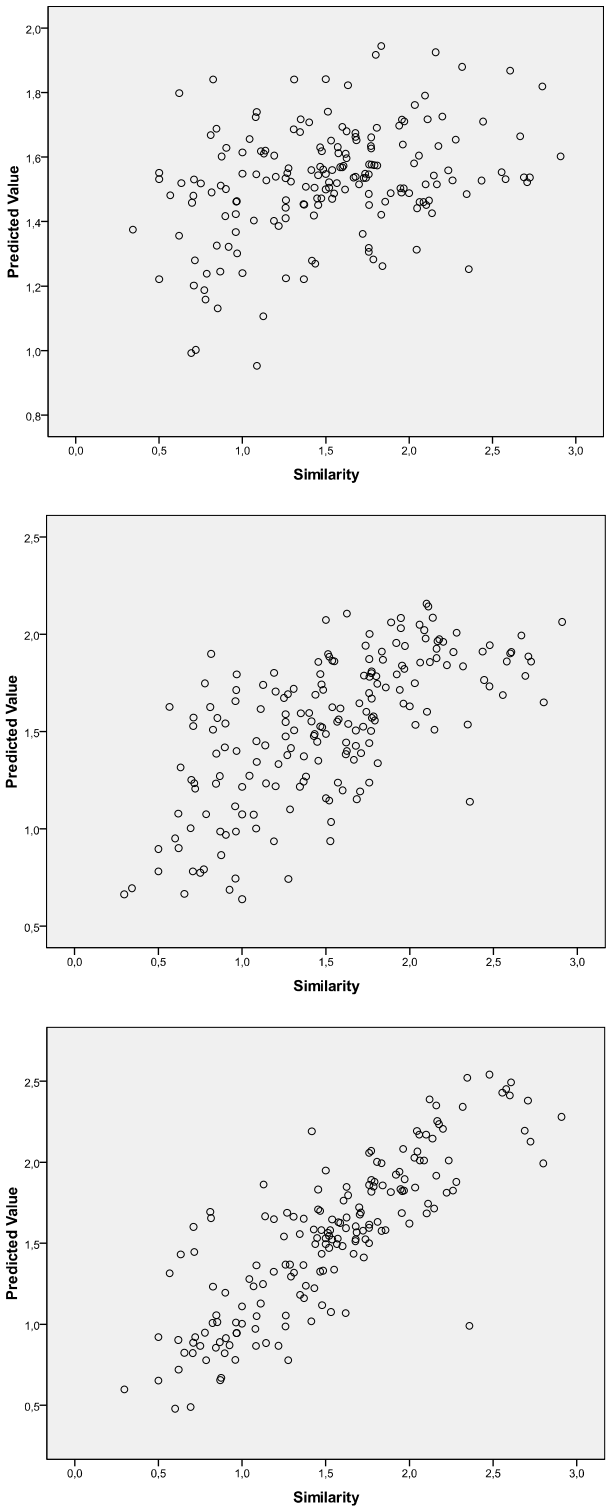

Fig. 1. The predicted-by-observed charts for the objective (top), subjective (middle), and joint (bottom) ANN models

TABLE V. IMPORTANCE OF OBJECTIVE AND SUBJECTIVE FACTORS IN THE JOINT ANN MODEL

Factor		*Importance*	*Normalized importance*
ID	*Type*		
K3HG	Subjective	0.125	100.00%
K6HP	Subjective	0.123	97.80%
K4RP	Subjective	0.117	93.30%
K8SD	Subjective	0.070	56.10%
K5AP	Subjective	0.059	46.90%
number_of_warnings	Objective	0.052	41.90%
K10BT	Subjective	0.048	38.60%
K1MF	Subjective	0.047	37.30%
K7NT	Subjective	0.046	36.40%
K9EA	Subjective	0.042	33.10%
Alexa_global_rank	Objective	0.041	32.60%
number_of_errors	Objective	0.039	31.10%
FK_grade_level	Objective	0.039	31.10%
page_load_time	Objective	0.037	29.50%
Alexa_bounce_rate	Objective	0.037	29.20%
K2CC	Subjective	0.035	28.00%
site_sections	Objective	0.023	18.20%

The RErr for the joint model (0.588), representing the objectively-extended Kansei Engineering approach, was not an improvement, which we partially attribute to the model overtraining due to few values in the dataset (ANNs are known to need a lot of data for proper training). Also, due to pilot nature of our study, we did not perform proper engineering of meaningful "objective" website metrics, but rather used the ones that were easy to collect. Still, the conceptual validity of our approach of collecting website-related data from global web services seems justified, as the use of Linked Open Data is increasingly used to acquire contextual information [9].

Our further research prospects include deeper analysis of website-intrinsic metrics and collecting more robust set of evaluations. In addition to the already employed machine learning approaches, statistical techniques such as principal component analysis and factor analysis to identify predictors of similarity could be used.

ACKNOWLEDGMENT

The reported study was funded by RFBR according to the research project No. 16-37-60060 mol_a_dk.

REFERENCES

[1] Glass, R.L., Facts and fallacies of software engineering. Addison-Wesley Professional, 2002.

[2] Millions Lines of Code - Information is Beautiful. Accessed 07 Apr 2017 at http://www.informationisbeautiful.net/visualizations/million-lines-of-code/.

[3] Kumar R. et al., "Webzeitgeist: design mining the web," SIGCHI Conf. on Human Factors in Comp. Systems, 2013, pp. 3083–3092.

[4] Nielsen, J., Fresh vs. Familiar: "How Aggressively to Redesign," NNGroup, 2009.

[5] Heil, S., Bakaev, M., Gaedke, M., "Measuring and Ensuring Similarity of User Interfaces: The Impact of Web Layout," In: Web Information Systems Engineering – WISE 2016. Lecture Notes in Computer Science, vol. 10041. Springer, 2016, pp. 252-260.

[6] Gajos, K., Wu, A., and Weld, D.S.: "Cross-device consistency in automatically generated user interfaces," In: 2nd Workshop on Multi-User and Ubiquitous UIs, 2005, pp. 7–8.

[7] Liao, T.W., Zhang, Z., and Mount, C.R., "Similarity measures for retrieval in case-based reasoning systems," Applied Artificial Intelligence, vol. 12 (4), 1998, pp. 267–288.

[8] van Rooij, G. et al., "A Data Type-Driven Property Alignment Framework for Product Duplicate Detection on the Web," In: Web Information Systems Engineering – WISE 2016. Lecture Notes in Computer Science, vol. 10041, 2016, pp. 380–395.

[9] Allahyari M. and Kochut K., "Semantic Context-Aware Recommendation via Topic Models Leveraging Linked Open Data," In: Web Information Systems Engineering – WISE 2016. Lecture Notes in Computer Science, vol. 10041. Springer, 2016, pp. 263–277.

[10] Al-Qutaish, R.E., "An investigation of the weaknesses of the ISO 9126 international standard," In Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on, vol. 1, pp. 275–279. IEEE, 2009.

[11] Bevan, N., Carter, J., Earthy, J., Geis, T., and Harker, S., "New ISO Standards for Usability, Usability Reports and Usability Measures," In International Conference on Human-Computer Interaction. Springer International Publishing, 2016, pp. 268–278.

[12] Kui, K.M., Ali, K.B., and Suryn, W., "The analysis and proposed modifications to ISO/IEC 25030—software engineering—software quality requirements and evaluation—quality requirements," Journal of Software Engineering and Applications, 9(04), 2016, p.112.

[13] Lauesen, S. and Younessi, H., "Six Styles for Usability Requirements," In REFSQ, vol. 98, 1998, pp. 155–166.

[14] Guo, F. et al., "Optimization design of a webpage based on Kansei Engineering," Human Factors and Ergonomics in Manufacturing & Service Industries 26 (1), 2016, 110–126.

[15] Lin, Y.C., Yeh, C.H., and Wei, C.C., "How will the use of graphics affect visual aesthetics? A user-centered approach for web page design," Int. J. of Human-Computer Studies 71 (3), 2013, pp. 217–227.

[16] Park, B.J., Oh, S.K., and Pedrycz, W., "The design of polynomial function-based neural network predictors for detection of software defects," Information Sciences 229, 2014, pp. 40–57.

[17] Dash, Y. and Dubey, S.K., "Quality prediction in object oriented system by using ANN: a brief survey," Int. J. of Adv. Research in Comp. Science and Software Engineering, 2(2), 2012.

[18] Vanmali, M., Last, M., and Kandel, A., "Using a neural network in the software testing process," International Journal of Intelligent Systems, 17(1), 2002, pp. 45–62.

[19] Bakaev, M., Khvorostov, V., Heil, S. and Gaedke, M., 2017, June. "Web Intelligence Linked Open Data for Website Design Reuse," In International Conference on Web Engineering. Lecture Notes in Computer Science. Springer, 2017, pp. 370–377.

[20] Bakaev, M., Gaedke, M., and Heil, S., "Kansei Engineering Experimental Research with University Websites," Technical Report, TU Chemnitz, CSR-16-01, 2016.

[21] Song, Z., Howard, T.J., Achiche, S., and Özkil A.G., "Kansei Engineering and Web Site Design," In ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2012, pp. 591–601.