Assessing Similarity for Case-Based Web User Interface Design

Maxim Bakaev(✉)

Novosibirsk State Technical University, Novosibirsk, Russia
maxis81@gmail.com

Abstract. It has been said "all design is redesign", and it is particularly true for websites, whose number in the today's online environment has reached 1 billion. In our paper, we justify case-based approach (CBR) to designing web user interfaces (WUIs) and outline some currently unsolved problems with its application. In this research work, we focus on definition and measurement of similarity, which is essential for all the stages of the CBR process: Retrieve, Reuse, Revise, and Retain. We specify the structure of a case in the web design domain (corresponding to a web project) and outline the ways to measure similarity based on the feature values. Further, we construct artificial neural network model to predict target users' subjective similarity assessments of websites that relies on website metrics collected by our dedicated "human-computer vision" software. To train the model, we also ran experimental survey with 127 participants evaluating 21 university websites. The analysis of the factors' importance suggests that frequency-based entropy measure and the proposed index of difficulty for visual perception affected subjective similarity the most. We believe the described approach can facilitate design reuse on the web, contributing to efficient development of more usable websites crucial for the e-society advancement.

Keywords: Web engineering · User interfaces · Software reuse
Computer-aided design

1 Introduction

Given the significant amount of financial and human resources spent on creating new and re-designing existing websites, one should wonder if these expenses are entirely justified and valuable for the society. The number of existing websites that are operational and accessible on the World Wide Web is currently estimated as 100–250 millions. Reuse of such an extensive collection of solutions available to all should play more important role in today's web engineering for the needs of e-society. Nowadays, conventional websites are rarely created from scratch, as web design (front-end) and web development frameworks partially automate the process. The web frameworks provide libraries of pre-made functionality and user interface (UI) elements, but they are generally detached from the multitude of websites already existing on the Web.

Computer-aided design systems in architecture, mechanical engineering, etc. involve testing of existing solutions and evaluation of their performance in fulfilling the

D. A. Alexandrov et al. (Eds.): DTGS 2018, CCIS 858, pp. 353–365, 2018.
https://doi.org/10.1007/978-3-030-02843-5_28

requirements. But despite the emergence and development of web analytics and web design mining tools (such as [1]), there are currently no repositories of web design examples that would both allow finding existing solutions relevant to a new project's requirements and appraising their quality based on accumulated use statistics. As the result, neither a web designer choosing an appropriate web UI element in *Bootstrap* framework, nor a prospective business website owner browsing through an endless collection of pre-made web design templates [2], has any estimation of the solution's success chance with target users. Naturally, the existing website holders do mind sharing their use statistics to aid their prospective competitors succeed, while designs can be copyright-protected. But another impediment is that we currently don't have an integrated engineering approach or technical means to reuse solutions in the web design domain. For that end, we consider employing case-based reasoning (CBR), a reasonably mature AI method that has a record of fruitful practical use in various fields.

Case-based reasoning is arguably the AI method that best reflects the work of human memory and the role of experience. It continues to draw increased interest, particularly on account of the current rapid development of e-society and e-economy, with their Big Data and Knowledge Engineering technologies. CBR implies the following stages, classically identified as Retrieve, Reuse, Revise, and Retain [3]:

- describing a new problem and finding similar problems with known solutions in a CB;
- adapting the solutions of the retrieved problems to the current problem, in an assumption that similar problems have similar solutions;
- evaluating the new solution and possibly repeating the previous stages;
- storing the results as a new *case*.

So, each case consists of a *problem* and a *solution*, plus the latter can be supplemented with the description of its effect on the real world, i.e. the solution *quality*.

Overall, it's recognized that "…design task is especially appropriate for applying, integrating, exploring and pushing the boundaries of CBR" [4], but workability of the method depends of the design field's particularities. The attempts to use CBR in design were already prominent in the early 1990s [5], with AskJef (1992) seemingly being the first notable intelligent system in this regard, whose scalability however now seems doubtful, since it lacked a reliable knowledge-engineering foundation. Nowadays, with regard to web design, CBR appears to be better established in software engineering [6] and web services composition [7], compared to the user interaction aspect. A notable example of CBR application for web interaction personalization is [8], but the software operates on an already existing website and seems to be feasible mostly for projects in which repetitive visits of the same user are entailed.

The generally recognized advantages of CBR include its applicability to complex domains with unknown knowledge model that is not required for the method to work, ability to learn from both success and failure (since both can accumulate in the knowledge base), reliance on well-established database technologies, etc. However, the method depends heavily upon well-structured and extensive collection of cases, while adaptation of the end result from several relevant solutions can be problematic, as knowledge model have not been identified. In particular it means that a significantly large number of cases have to be collected before the method can start yielding any practically feasible results, and that feature engineering – that is, constructing the set of measurable properties to describe problems in CBR – is of crucial importance for the overall success. Web design appears suitable for the CBR approach since:

1. There are potentially a huge number of cases, given the 100–250 millions of websites currently openly available on the Internet. Today's web mining systems are capable of scraping their code, styling and content with reasonable efficiency.
2. The "lazy generalization" strategy of CBR is advantageous, since knowledge in web design is largely represented as qualitative principles and guidelines, while formal knowledge models or rules are relatively scarcely used.
3. The solutions can be promptly applied in the real world, their quality is not critical and they can be revised easily. That is, we consider rather conventional e-business or e-government websites, not e.g. a web-based interface of a nuclear plant management system.

At the same time, potential difficulties with CBR approach application in the web design domain include:

1. **Retrieve and Retain:** there's no yet an agreed structure of web design features that significantly influence the solutions usability, attractiveness for users, etc. Additionally, the case needs to accommodate quality attributes for several solutions, as different versions of websites can operate in different times, but basically solve the same problem (goal). The latest version is not necessarily the best solution – everyone probably has encountered a new design that is worse than an old one.
2. **Reuse and Revise:** to the extent of our knowledge, there are no established approaches for generating new web designs from several relevant solutions in the course of their adaptation to the initial problem. Actually, direct modification of existing solutions is very much restricted in the web design domain, and rather roundabout approaches have to be employed, where newly composed solutions are iteratively adjusted to match the retained ones (see in our other work [9]).
3. **Similarity measurements:** this missing link is actually required by both of the above items. The CBR algorithm for web design needs calculating similarity (a) between problems, to retain relevant cases, and (b) between solutions, to compose new solutions that are similar to the exemplar retained ones.

So, our current work is dedicated to the similarity measurements for the purposes of the CBR approach to web user interface design, which promises significant boost in conventional websites engineering. This paper is built upon several our previously published works (we provide references where appropriate) and integrates them into the unified case- and component-based approach to web design. Although the paper has a single author, "we" pronoun will be used throughout the text, to recognize the previous work of the collaborators. In Sect. 2 we consider the case structure in the web design domain and outline the ways to measure similarity based on the feature values. Further, we propose the metric-based technique and the software tool (the visual analyzer that we developed) to predict target users' subjective similarity assessments of websites using artificial neural network (ANN) model. In Sect. 3, we describe the experimental survey session where we collected the training data, and the construction and training of the actual ANN user behavior model. In Conclusions, we summarize our findings and outline prospects for further work.

2 Case-Based Approach in Web Design and Similarity Measures

2.1 Problem Features and Similarity

Different disciplines place distinct emphasis on the CBR-related activities of case storage, case indexing, case retrieval, and case adaptation (the Retrieve stage remains arguably the most popular). Still, there is a general consensus among researchers and practitioners about the crucial importance of devising an accurate form of problem description for the success of machine learning and automated reasoning in AI: e.g. "… a critical pain point in building (trained data) systems is feature engineering" [10]. Meanwhile, feature engineering appears to often remain a creative task performed "manually" by knowledge engineers, though the major stages of the conventional process can be identified as: forming the excessive list of potential features (e.g. through brainstorming session), implementing all or some of them in a prototype, and selecting relevant features by optimizing the considered subset.

First, it should be noted that there's a fair amount of research works that deal with feature selection for web pages, particularly for automated classification purposes [11]. Indeed a web page is a technically opportune object for analysis, as it is represented in easily processable code (HTML, CSS, etc.), but it's not self-contained, either content-wise or in terms of design resolutions, and can hardly be appropriate as a solution in a case. Clearly, such design- and goal-wise complete entity web project should correspond to case in CBR, while website is a solution, of which the case may have several. There are respective approaches aimed on selecting features for software or web projects, though focused rather on knowledge organization [6] or web service composition [7]. As we mentioned before, there seems to be no agreed structure of features in the web design domain, so we performed informal feature engineering for reuse and outlined their use in the cases' similarity calculation.

We founded ourselves upon the model-based approach to web UI development, which generally identifies three groups of models: (1) per se interface models – Abstract UI, Concrete UI, and Final UI, (2) functionality-oriented models – Tasks and Domain, and (3) context of use models – User, Platform, and Environment. Of these, we consider the Domain, Tasks, and User of higher relevance to web design reuse, while Platform and Environment models rather relate to website's back-office. Also, not all existing website designs are equally good (in contrast to e.g. re-usable programming code), so quality aspects must be reflected in the feature set.

Domain-Related Features. Reuse of design is considered domain-specific [12], and indeed a website from the same domain has a pretty much better chance to aid in solving the problem in CBR. Although the domain theoretically can be inferred from website content, this is complex and computationally expensive, so we propose using available website classifications by major web catalogues. For example, DMOZ claims to contain more than 1 million hierarchically-organized categories, while the number of included websites is about 4 million, which implies highly detailed classification. The domain similarity then can be defined as the minimal number of steps to get from one category item to another via hierarchical relations, divided by the "depth" of the item, to reduce potential bias for less specifically classified websites.

Task-Related Features. Although user activities on the web may be quite diverse, conventional websites within the same domain have fairly predictable functionality. For the purposes of CBR, there seems to be little need to employ full-scale task modeling notations, such as W3C's CTT or HAMSTERS [13], especially since by themselves they do not offer an established approach for evaluating similarity between two models. We believe that particulars of a reusable website's functionality can be adequately represented with the domain features plus the structured inventory of website chapters that reflect the tasks reasonably well. Then, the currently well-developed semantic distance methods (see e.g. [14]) can be used to retrieve the cases with similar problem specifications. A potential caveat here, notorious for folksonomies in general, is inventiveness (synonyms) or carelessness (typos) of some website owners – so, first the chapter labels would have to be verified against a domain-specific controlled vocabulary.

User-Related Features. User stereotype modeling in web interaction design employs a set of reasonably well established features to distinguish a group of users: age, gender, experience, education and income levels, etc. The corresponding personas or user profiles (usually no more than 3 different ones) are created by marketing specialists or interaction designers and are an important project artifact [8]. Evaluation of similarity between users is quite well supported by knowledge engineering methods and is routinely performed in recommender systems, search engines [15], social networks [16], etc. Thus, the real challenge is obtaining concrete values for the relevant features in the user model for someone else's website.

Quality-Related Features. The quality-related features won't be used for calculating similarity, but among the several potentially relevant cases (or among solutions in the same case – website versions) we would generally prefer solutions that have better quality. Website quality is a collection of attributes, some of them can belong to very different categories, e.g. Usability and Reliability, and their relative importance may vary depending of the project goals and context [12]. Correspondingly, today's techniques for assessing the quality attributes are very diverse: auto-validation of code, content or accessibility; load and stress tests; checklist of design guidelines, user testing, subjective impression surveys, etc. Thus, the set of quality-related features must remain customizable and open to be "fed" by diverse methods and tools – actually, the more quality attributes can be maintained, the better.

2.2 The Web Designs Similarity

After the cases are retrieved from the case base based on a certain similarity measure, the "classical" CBR prescribe adapting their solutions to the new problem. However, in web design domain this process (basically, the Reuse and Revise stages) can't be performed directly, as the solutions' back-office and server-side code is generally not available, while the designs are copyright-protected. The workaround (as we proposed in [9]) is to consider them as the reference solutions, generate new solutions from software and UI components, and iteratively make the new solutions similar to the reference ones. The problem, however, is that interactive evolutionary computation that involves human experts or even users to assess the similarity would make the adaptation process prohibitively slow. In resolving this, we propose relying on trained human behavior models – i.e. using pre-supplied human assessments to make predictions on the new solutions' similarity to the reference ones.

Classically, behavior models in human-computer interaction have interface design's characteristics and the context of use (primarily, users' characteristics) as the inputs, and they output an objective value relating to end users, preferably a design objective (usability, aesthetics, etc. [17]. In our study, we will fix the user characteristics by employing a relatively narrow target user group to provide the similarity assessments for a fixed web projects Domain. In representing website designs, we will rely on metric-based approach, i.e. describe the solutions with a set of auto-extracted feature values responsible for subjective website similarity perception in the target users.

There is plenty of existing research works studying the effect of website metrics on the way users perceive them and on the overall web projects' success (one of the founding examples is [18]). Particularly, both user's cognitive load and subjective perceptions are known to be greatly influenced by perceived visual complexity [19], which in turn depends of the number of objects in the image, their diversity, the regularity of their spatial allocation [20], etc.

In our study we employ the dedicated software tool that relies on computer vision techniques to extract the web interface metrics – the "visual analyzer", which we developed within the previously proposed "human-computer vision" approach. The visual analyzer takes visual representation (screenshot) of a web interface and outputs its semantic-spatial representation in machine-readable JSON format (see [21] for more detailed description of the analyzer's architecture, the involved computer vision and machine learning libraries, etc.). Based on the semantic-spatial representation, the analyzer is capable of calculating the following metrics relevant for the purposes of our current research:

1. The number of all identified elements in the analyzed webpage (UI elements, images, texts, etc.): *N*;
2. The number of different elements types: *T*;
3. Compression rate (as representation of spatial regularity), calculated as the area of the webpage (in pixels) divided by the file size (in bytes) of the image compressed using the JPEG-100 algorithm: *C*;
4. "Index of difficulty" for visual perception (see in [20]): *IDM*, calculated as:

$$IDM = \frac{N \log_2 T}{C} \tag{1}$$

5. Relative shares of the areas in the UI covered by the different types of UI elements:
6. Textual content, i.e. area under all elements recognized as textline: *Text*;
7. Whitespace, i.e. area without any recognized elements: *White*;

8. In addition to the metrics output by the analyzer, we also employed the standard Matlab's entropy(I) function (returns a scalar value E reflecting the entropy of grayscale image I) to measure frequency-based entropy of the website screenshot: *Entropy*.

The above metrics will act as the basic factors (F_i) for the ANN model we construct in the next chapter, in order to predict target users' similarity assessments of website designs. ANNs are gaining increased popularity recently, as they have very reasonable computational effectiveness compared to other AI or statistical methods and they don't require explicit knowledge of the model structure. The disadvantage is that they require a lot of diverse data for learning, and the results are hard to interpret in a conceptually meaningful way. ANNs are first trained 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 (the model quality) 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).

3 The Similarity Assessment

To obtain the subjective similarity evaluations for the ANN training, we ran experimental survey sessions with human evaluators. In the current research work, the input neurons are strictly the metrics that can be evaluated automatically for a webpage, without any subjective assessments. In one of our previous studies of subjective similarity, however, we relied on human evaluations for the "emotional" dimensions of websites, collected per the specially developed Kansei Engineering scales, to predict similarity of websites [22]. That ANN model had relative error of 0.559, which will act as the baseline for our current study, where the number of required evaluations is dramatically lower.

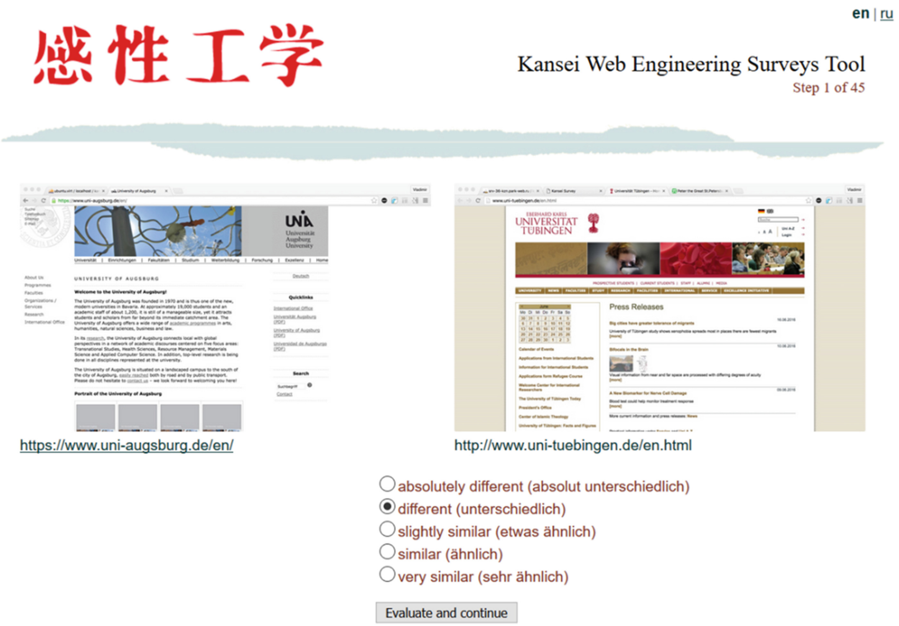
3.1 The Experimental Design

The research material was university websites (*Career and Education* domain in DMOZ), 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. Correspondingly, the total number of distinct website pairs for the similarity assessments was $C_{21}^{2} = 210$.

The assessments were collected from 127 participants (75 male, 52 female), aged 17–31 (mean = 20.9, SD = 2.45), who represented the target users. The subjects were university students (mostly majoring in Computer Science) or staff members: 100 from

Russia (Novosibirsk State Technical University) and 27 from Germany (Chemnitz Technical University). The participants used diverse equipment and environment: 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 subject, and afterwards they could submit comments to their evaluations.

The participants used our specially developed survey software (currently available at http://ks.wuikb.tech/phase2.php). Each subject was asked to assess subjective similarity for 45 distinct website pairs composed from 10 randomly selected websites (see in Fig. 1). 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).


en | ru
感性工学
Kansei Web Engineering Surveys Tool
Step 1 of 45
https://www.uni-augsburg.de/en/
http://www.uni-tuebingen.de/en.html
Evaluate and continue

○ absolutely different (absolut unterschiedlich)
● different (unterschiedlich)
○ slightly similar (etwas ähnlich)
○ similar (ähnlich)
○ very similar (sehr ähnlich)

Fig. 1. The survey software screen with similarity assessment for two websites

3.2 Descriptive Statistics

In total, the 127 subjects provided 5715 similarity assessments, so for each of the 210 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 similarity is in ordinal scale, so the values are given just for reference).

Further, we applied our visual analyzer to obtain the metrics for the experimental websites. Website #14 was excluded from the analysis due to technical difficulties with the screenshot (so, 90.5% of averaged similarity assessments were valid). The values for the 7 metrics extracted by the analyzer are presented in Table 1.

Table 1. The metrics for the website provided by the visual analyzer

Website ID	N	T	C	Text,%	White, %	Entropy	IDM
1	32	3	1.366	8.49	88.55	4.437	37.1
2	41	6	1.886	3.09	91.84	3.518	56.2
3	53	5	3.013	1.16	96.28	2.984	40.8
4	80	5	2.540	3.62	96.25	4.478	73.1
5	120	5	3.184	1.81	92.92	3.497	87.5
6	43	3	2.146	3.82	92.14	4.108	31.8
7	19	6	1.917	9.62	87.14	3.060	25.6
8	44	5	2.623	7.63	89.22	2.044	39.0
9	49	5	1.815	3.02	93.41	6.589	62.7
10	68	7	2.604	5.74	90.83	2.998	73.3
11	64	7	2.331	1.68	91.44	3.836	77.1
12	157	6	1.566	2.27	95.49	3.635	259.2
13	54	6	2.246	0.98	91.36	5.568	62.1
15	70	7	1.622	0.57	94.36	5.846	121.2
16	92	6	2.577	6.16	90.01	2.374	92.3
17	118	7	1.662	10.32	82.36	3.494	199.4
18	131	8	2.816	2.01	92.69	4.533	139.6
19	48	5	1.190	0.63	97.74	4.794	93.6
20	29	3	2.136	22.68	77.30	4.742	21.5
21	57	7	2.665	1.38	92.77	4.062	60.1
Mean (SD)	68.45 (28.89)	5.60 (1.14)	2.20 (0.46)	4.83 (3.68)	91.20 (3.30)	4.03 (0.89)	82.66 (41.42)

The distance measure between a pair of websites per each of the measured dimensions was introduced the ratio between the largest and the smallest value for the two websites (so 1 means no difference, larger values indicate greater difference):

$$Diff(F_i) = \frac{Max\{F_i(website_j), F_i(website_k)\}}{Min\{F_i(website_j), F_i(website_k)\}} i = \overline{1,7}; j = \overline{1,21}; k = \overline{1,21} \qquad (2)$$

Please note that the distance measure could be set this way since all the metrics were in rational scale, unlike in our previous work [22], where the human assessments of the factors' values were ordinal. The Shapiro-Wilk tests suggested that for all seven *Diff(F_i)* factors the normality hypotheses had to be rejected ($p < 0.001$).

The analysis of correlations (non-parametrical Kendall's tau-b for ordinal scales) for the Similarity assessments found significant negative correlations with distances

Diff(Entropy) ($\tau = -0.146$, $p = 0.003$), *Diff(IDM)* ($\tau = -0.100$, $p = 0.04$), *Diff(Text)* ($\tau = -0.147$, $p = 0.003$), and *Diff(White)* ($\tau = -0.180$, $p < 0.001$).

3.3 The ANN Model for Assessing Web Designs Similarity

In the ANN model, the single output neuron was Similarity, averaged for each websites pair (website$_j$, website$_k$) per all the participants who assessed it, whereas the input neurons were the seven *Diff(Fi)* covariates for the websites. We employed Multilayer Perceptron method with *Scaled conjugate gradient* optimization algorithm in SPSS statistical software, hidden layer activation function was *Hyperbolic tangent*, output layer activation function was *Identity*. The partitions of the datasets (210 pair-wise similarity values) in each of the three models were specified as 70% (training) – 20% (testing) – 10% (holdout). The number of neurons in the single hidden layer was set to be selected automatically, and amounted to 4 neurons in the resulting model. The relative error in the best model was 0.597 for the holdout set. We also performed the factors importance analysis, whose results are presented in Table 2.

Table 2. The factors importance analysis

Factor	Importance	Normalized importance
Diff (Entropy)	0.283	100.0%
Diff(IDM)	0.223	78.8%
Diff(White)	0.155	54.6%
Diff(Text)	0.118	41.8%
Diff(T)	0.115	40.5%
Diff(C)	0.067	23.6%
Diff(N)	0.039	13.8%

Alternative ANN models that for the input neurons employed the factors values for the two websites separately, i.e. $F_i(website_j)$ and $F_i(website_k)$ instead of the differences, had notably lower predictive quality. The best model, with all the 14 F_i plus the categorical values of the website country (Russian or German), had relative error of 0.737 for the holdout set. The model seemingly suffered from overtraining, which may imply more training data would be required.

We also attempted ordinal regression to test whether the assessed similarity could be predicted by the seven $Diff(F_i)$ factors. The resulting model was highly significant ($\chi^2(7) = 43.86$, $p < 0.001$), but had rather low Nagelkerke pseudo $R^2 = 0.206$. Moreover, the proportional odds assumption had to be rejected ($\chi^2(1155) = 1881$, $p < 0.001$), which suggests that the effects of the explanatory variables in the ordinal regression model are inconsistent.

4 Conclusions

The general idea of case-based design reuse has been around for quite a while, but its potential in web engineering is particularly appealing. In today's e-society, an archetypal web design company employs no more than 10 people, has no market share to speak of, and mostly works on fairly typical projects. Greater reuse of websites and automated composition of new solutions could significantly increase their efficiency, allowing to focus on e-marketing, content creation, usability refinement, etc.

In the current paper we focus on assessment of similarity, which is crucial within the CBR approach to WUI design, since retrieval of relevant cases and solutions is by and large based on similarity measure. We carried out informal feature engineering for web projects, inspired by the popular model-based approach to web interface development – thus the Domain, Task, and User dimensions – and outlined how similarity measures could be calculated for each of them. We also argue that CBR application in web design domain also requires measuring similarity between the new solution and the retrieved solutions, since direct adaptation of the latter is restricted by technical and legal considerations.

To predict similarity of web designs without actual users (as relying on human experts or users to assess all the similarities would make the adaptation process prohibitively slow), we proposed the approach based on auto-extracted website metrics. These values were extracted by our dedicated software, the web analyzer, and used as the basic factors in the predictive ANN model illustrating feasibility of the approach. Its relative error of 0.597 is rather appropriate compared to relative error of 0.559 in the baseline model relying on user assessments of emotional dimensions [22], while the other considered models showed lower performance. The analysis of the factors' importance suggests that frequency-based entropy measure was the most important for subjective similarity, in contrast to compression measure introduced in the analyzer to reflect spatial orderliness in WUI, which had considerably lower importance. The index of difficulty for visual perception that we previously devised [20] and that is based on the analyzer's measurements also had high importance, which implies significant effect of visual complexity on subjective similarity in websites. The aerial measures of shares under text and whitespace had moderate importance, while the number of elements in web interface was the least important factor – somehow unexpectedly, as our previous research suggests that the analyzer is rather accurate in this regard [21].

Our further research will be aimed on studying the dimensions of similarity and improving the model, particularly through: (a) getting more similarity-related website metrics to be assessed by the analyzer; (b) obtaining and utilizing more training data, as the extended ANN model suffered from their shortage.

Acknowledgement. The reported study was funded by RFBR according to the research project No. 16-37-60060 mol_a_dk. We also thank those who contributed to developing the visual analyzer software and collecting the human assessments: Sebastian Heil, Markus Keller, and Vladimir Khvorostov.

References

1. Kumar, R., et al.: Webzeitgeist: design mining the web. In: SIGCHI Conference on Human Factors in Computer Systems, pp. 3083–3092 (2013). https://doi.org/10.1145/2470654.2466420
2. Norrie, M.C., Nebeling, M., Di Geronimo, L., Murolo, A.: X-Themes: Supporting Design-by-Example. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) ICWE 2014. LNCS, vol. 8541, pp. 480–489. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08245-5_33
3. De Mantaras, R.L., et al.: Retrieval, reuse, revision and retention in case-based reasoning. Knowl. Eng. Rev. **20**(3), 215–240 (2005). https://doi.org/10.1017/S0269888906000646
4. Goel, A.K., Craw, S.: Design, innovation and case-based reasoning. Knowl. Eng. Rev. **20** (3), 271–276 (2005). https://doi.org/10.1017/S0269888906000609
5. Schmitt, G.: Case-based design and creativity. Autom. Constr. **2**(1), 11–19 (1993)
6. Rocha, R.G., et al.: A case-based reasoning system to support the global software development. Procedia Comput. Sci. **35**, 194–202 (2014). https://doi.org/10.1016/j.procs.2014.08.099
7. De Renzis, A., et al.: Case-based reasoning for web service discovery and selection. Electron. Notes Theor. Comput. Sci. **321**, 89–112 (2016). https://doi.org/10.1016/j.entcs.2016.02.006
8. Marir, F.: Case-based reasoning for an adaptive web user interface. In: The International Conference on Computing, Networking and Digital Technologies (ICCNDT2012), pp. 306–315 (2012)
9. Bakaev, M., Khvorostov, V.: Component-based engineering of web user interface designs for evolutionary optimization. In: 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2018), pp. 335–340
10. Anderson, M.R., et al.: Brainwash: A Data System for Feature Engineering. In: CIDR (2013)
11. Mangai, J.A., Kumar, V.S., Balamurugan, S.A.: A novel feature selection framework for automatic web page classification. Int. J. Autom. Comput. **9**(4), 442–448 (2012). https://doi.org/10.1007/s11633-012-0665-x
12. Glass, R.L.: Facts and Fallacies of Software Engineering. Addison-Wesley Professional, Boston (2002)
13. Martinie, C., et al.: A generic tool-supported framework for coupling task models and interactive applications. In: Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 244–253 (2015). https://doi.org/10.1145/2774225.2774845
14. Park, J., Choi, B.C., Kim, K.: A vector space approach to tag cloud similarity ranking. Inf. Process. Lett. **110**(12–13), 489–496 (2010). https://doi.org/10.1016/j.ipl.2010.03.014
15. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: Proceedings of the 16 ACM Conference on information and knowledge management, pp. 525–534 (2007). https://doi.org/10.1145/1321440.1321515
16. Kosinski, M., et al.: Manifestations of user personality in website choice and behaviour on online social networks. Mach. Learn. **95**(3), 357–380 (2014). https://doi.org/10.1007/s10994-013-5415-y
17. Oulasvirta, A.: User interface design with combinatorial optimization. Computer **50**(1), 40–47 (2017). https://doi.org/10.1109/MC.2017.6
18. Ivory, M.Y., Hearst, M.A.: Statistical profiles of highly-rated web sites. In: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, pp. 367–374 (2002). https://doi.org/10.1145/503376.503442

19. Reinecke, K., et al.: Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 2049–2058 (2013). https://doi.org/10.1145/2470654.2481281
20. Bakaev, M., Razumnikova, O.: Opredeleine slozhnosti zadach dlya zritelno-prostranstvennoi pamyati i propustkoi spospobnosti cheloveka-operatora. Upravlenie bol'shimi sistemami=Large-Scale Systems Control **70**, 25–57 (2017). (In Russian)
21. Bakaev, M., Heil, S., Khvorostov, V., Gaedke, M.: HCI Vision for Automated Analysis and Mining of Web User Interfaces. In: Mikkonen, T., Klamma, R., Hernández, J. (eds.) ICWE 2018. LNCS, vol. 10845, pp. 136–144. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91662-0_10
22. Bakaev, M., et al.: Evaluation of user-subjective web interface similarity with Kansei engineering-based ANN. In: IEEE 25th International Requirements Engineering Conference, pp. 125–131 (2017). https://doi.org/10.1109/rew.2017.13