Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100310

Intelligence System for Supporting Human-Computer Interaction Engineering Processes

*M. N. Sheriyev**, *L. B. Atymtayeva, I. K. Beissembetov and B. K. Kenzhaliyev*

Kazakh British Technical University, Almaty, Kazakhstan

Received: 10 Oct. 2015, Revised: 10 Jan. 2016, Accepted: 11 Jan. 2016
Published online: 1 May 2016

Abstract: The aim of the paper is to study the development of an intelligent system to support Human-Computer Interaction (HCI) engineering process of web applications. The reported review starts from improved system architecture and its main modules. We assume that such intelligent systems are able to assess the level of usability of web applications, improve it by representing a list of recommendations and extend the knowledge related to HCI engineering. To all other the proposed system may store results of each project, allowing to evaluate the level of improvement of web usability.

Keywords: human computer interaction, engineering knowledge, base knowledge, frame-based model, production model, intelligent system.

1 Introduction

Human-computer interaction (HCI) is relatively young and broad research related section in Information Technology (IT) area, and currently has no finally settled definition. It includes the study of design, evaluation and implementation of interactive computer systems intended for human use, as well as related aspects. The Engineering processes in HCI is accompanied with the efforts to improve the quality of interaction with a computer system, and first of all, improve the quality parameters of the user interface.

According to the recommendations of experts at least 10% of total budget of software development project must be allocated to development of the usable interfaces with enough level of usability. The average improvement of key business indicators belongs to the market of web applications which occupies about 83% of the software development market [1]. This fact gives us evidence of significant economic efficiency of interactive software systems based on web-applications development. However, not all projects related to software development are implemented by using the practical and effective interaction design methods. As a result, the value of the benchmark as the percentage of successful completed tasks for web applications is no more than 81% [1] and for certain categories of users it is 1.5-2 times lower.

We can identify the problem of poor process of using the practical HCI knowledge which accompanies by significant time costs in searching, interpreting and applying the relevant recommendations or ready design patterns (which are the standard solutions used in the design interfaces) for developers [1]. To solve this problem we can propose the development of the different intelligent systems (IS) in order to support the design of interfaces by dividing into elements and organization of recommendations [1] [2]. These types of systems mostly give the opportunity to automate code generation and the process of the interface validation. The effective combination of these proposed approaches would reduce the time required for searching of appropriate recommendations in design, in finding the usability problems, and in suggesting the associated with them solutions for improvement of the user interfaces.

There are a huge number of recommendations regarding the web usability which come from different sources. But the finding of the necessary recommendations takes a long time for web application engineer. Of course, it is possible to use well-known web services such as:

1. Google PageSpeed Insights - service on recommendations of accelerating page loading. (guideline *Slow pages lose users* [3]).

* Corresponding author e-mail: madi.sheri@gmail.com

2.Screenfly - represent website template screens in various devices.
3.Cybernetic analytic system - assessment of the quality of the site.
4.MetroWEB - tool which helps us to manage usability knowledge and uses this knowledge during evaluation process
5.Markup Validation Service - popular service to validate HTML structure.

Unfortunately, services listed above are not able to solve a number of problems related to web usability. For example, these services can't identify the appearance of the home page link, availability of search navigation or presence of heading and body elements. To fill these gaps our goal becomes the development of an intelligent system for supporting HCI engineering processes and solving the issues in web usability area.

Use of such type of intelligent system may reduce the requirements for qualification of engineers, for time and budget spending, decrease number of errors and improve the quality of interaction for all users. In order to meet all proposed requirements the developing Intelligent system (IS) should contain the recommendations with detailed description and examples coming from the different sources and guidelines.

In one of our previous paper [1], we constructed basic architecture of intelligent system. The main purpose of our first version of intelligent system was assessment of applicability of knowledge engineering methods, which are able to solve real problems in HCI subject area. Knowledge base of the first version of IS [1] system was relatively small but enough for providing the testing of the proposed system and to detect possible disadvantages and errors in the projects. This paper describes the improved architecture and ontology model of IS, the ways of its application, and focuses on the results of implementation of the system. Finally, it demonstrates the analysis of usability problem detection and quality assessment of the testing projects.

2 Core of IS: hybrid model

The main advantage of proposed Intelligent System (IS) is constructing of the hybrid model [4]. This hybrid model consists from two types of models including the frame-based model with object-oriented approach and production model with opportunity of knowledge representation in the form of implications (if-else statements). This section describes how to integrate these two models to implement the functionality of IS.

2.1 Improvement of IS Architecture

Firstly we can stop at the constructing of IS Architecture. Improved model of IS Architecture is represented on the Fig. 1. Here we have added some elements such as evaluation of usability level of the projects, the analysis of improved projects, and extension of Solver element by adding the Project list component. These corrections allowed us to improve the representation of output information and to provide the validation and knowledge acquisition in the stage of Solver functioning.

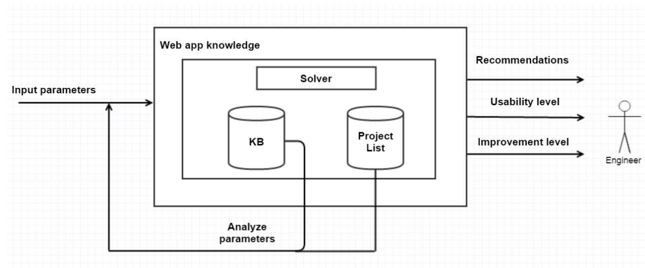

Fig. 1: Improved IS architecture

As it was specified in our paper [1] the solution of the problem for web applications requirement analysis contains the processing of the text information. Therefore Knowledge Base (KB) of IS includes vocabulary terms of system subject area used for text analysis, indexing recommendations and describing context design [5].

The existence of KB in intelligence system with controlled vocabulary can describe complex objects of subject area by comparing its list with basic terms. The context of project can be described in similar way but the set of terms formed by an intelligent system automatically directs to the target user whereas the requirements are defined as input information. By comparing these two sets the intelligent system can determine the relevance of each recommendation which is evaluated as a potential component of ”solving the problem” or output recommendation list. Validation mechanism and learning for intelligent system is special web portal of knowledge which proposes to the users opportunity to assess the applicability of the knowledge extracting from IS. Recommendations from HCI area which originally presented in natural language, can be transformed into knowledge formalized by the rules of inference production model.

One of the feature of IS is that after testing web application, it saves all results related to usability testing in project list. Project list contains two main information: list of guidelines and usability level. Usability level (UL) is the average percentage of quality of web application, which can be calculated by IS using OWL file.

To improve the UL, HCI engineer is able to improve web application according to guidelines presented by IS. After reconstruction, HCI engineer may pass web application in IS again to compare new and previous results. IS analyze new data, try to find tested web application from project list and remove guidelines which passed the validation. This way, we increase the usability level. So, if we subtract the value old UL from new UL, we get improvement level of tested web application. Engineer may iterate testing process many times till do not get desirable results of Usability level.

2.2 Ontology in IS Knowledge Engineering

There are various knowledge representation models such as formal-logical, production, framing, semantic networks, conceptual maps, and etc. One of the often used knowledge representation model is ontology. According to the definition given by T. Gruber [6] [7] for artificial intelligence applications - "ontology is a formal explicit description of the terms in subject area and relations between them". The main advantage of using ontology model as a tool of knowledge reflection and organization is a systematic approach in modeling of subject area which allow to achieve a holistic view of subject area, its uniformity and connectivity [7].

The most common form of ontology is its graphical representation [7]. Usually it is a fuzzy difference between Ontology and Knowledge Base definitions. Therefore by using N. Guarino theory [8] we can suppose that ontology contains state-independent information meanwhile Knowledge Base (KB) works with state-dependent content.

For constructing such types of the Knowledge Engineering Systems as proposed Intelligent System the most effective way is to create the functional ontology [4] [7], and the modern tools like Protege frame-based ontology system [9] may allow to develop and integrate any new ontologies automatically.

Basing on IS architecture (fig.1) we can create the main functional ontology of IS system (Fig.2).

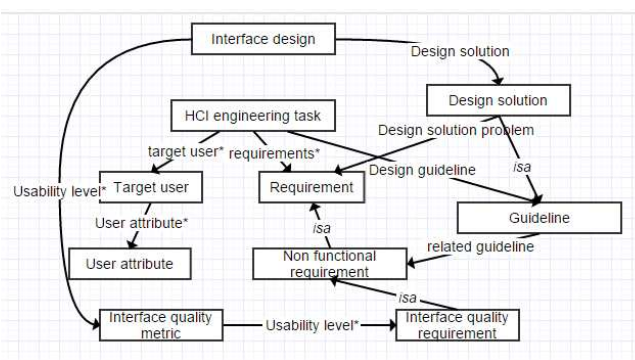

Fig. 2: Fragment of IS ontology

First class of ontology is HCI engineering task, which has relations with the classes HCI knowledge, guidelines and rules. In addition, the value of project context slot can be set, which includes any class of ontology (a child of the meta-class THING). The overall structure of the class "HCI engineering task" is shown in Fig. 3 [1].

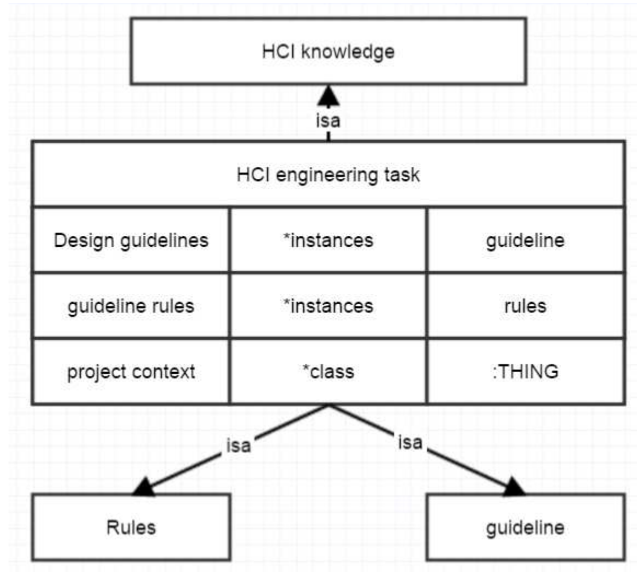

Fig. 3: HCI engineering task class structure

Law, Principle and Guideline are ontology classes, united in abstract meta class called HCI knowledge. As it is known the success of HCI engineering depends on the deep understanding of interaction context and human behavior patterns, which gives engineer the opportunity to make informed choices of the practical recommendations [1]. This shows the particular importance in organization the knowledge as a set of recommendations issued by IS, which corresponds to subsystem of explanations and makes it appropriate to provide the following:

1. To establish the links between various levels of HCI knowledge and knowledge of one level;
2. To create a link to frame instance;
3. To establish a link between an instance and context;
4. To be able to specify the "efficiency" or practical significance for recommendations.

Based on points 1 and 2 above, we added Finding, Source and Reference classes and also corresponding relationships with all classes in ontology. All relationships between classes can be combined into abstract class called HCI knowledge representation class. (Fig. 4).

Class *"Finding"* reflects some fact or empirical evidence obtained in research or practice in HCI area, class *"Source"* usually corresponds to a scientific publication, i.e. paper or book. Based on the points 3 and 4 above we can consider slots *"Tag"* and *"Efficiency"*. If

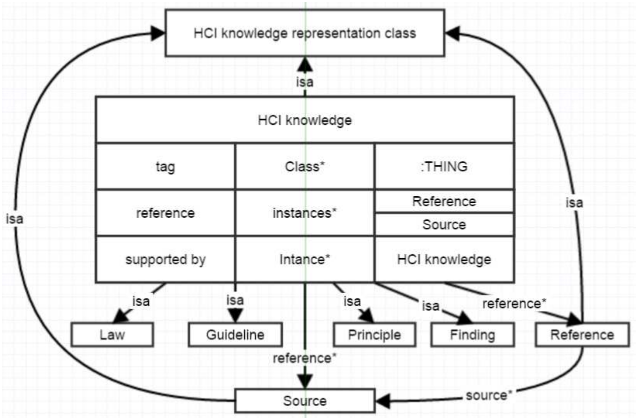

Fig. 4: HCI engineering representation class structure

the first one is the attribute of knowledge at all levels (i.e. class *"HCI knowledge"*), the second refers only to the *"Guideline"* class. Slot *"Tag"* is used to establish relation between an instance of knowledge and context of projected interaction. The value of this slot can be any class of ontology. Slot *"Efficiency"* reflects the evaluation of efficiency of recommendations in HCI engineering.

The end result of activity on designing interaction is interface which created as a result of trade-off solutions which meet the requirements and consider technological and other limitations [1].

Engineer can make a decision based on practical recommendations or higher level of knowledge existing in HCI. But some of rules in such decisions are not formalized. Today web applications can be divided into two components: content and design, which is closely connected each other. High-quality and user-friendly presentation of content and its adaptation for *"scanning"* style of perception is an important factor to improve the usability of web applications [1]

2.3 Frame-based model representation

As it was first introduced by Marvin Minsky [9] the term *"frame"* means the structure of knowledge which simulates human thinking and abstract image of an object, phenomenon, events, situations, and processes.

The main feature of frame model is that the original slots can be filled with *"lack of absence"* - ready-made values. Consequently, frame structure contains: name of the frame, pointers inheritance, slots, data type, and slots values (which can be omitted). The value of slot may be related to other frames, thereby realizing various kinds of relationships between concepts and procedural attachments, which implement procedural component in knowledge representation.

Thus, among the advantages of frame models is:

1. flexibility of the model, taking into account the possibility of all sorts of situations;
2. good way in organization of human memory and compatibility with textual knowledge extraction techniques;
3. the ability to represent knowledge in object-oriented approach;
4. the possibility of a combination of declarative and procedure components of knowledge representation.

Features and Benefits of frame model make it potentially applicable in our development of IS to support HCI engineering processes. Definitely we can use a frame script to represent the objectives of user interaction with web application. Thus, by using IS we are able to refine the relevant attributes of the frame: requirements for a web application, characteristics of the target user and set of recommendations. Fig. 5 represents a snippet how we integrate frame-based model by creating knowledge base with the list of recommendations.

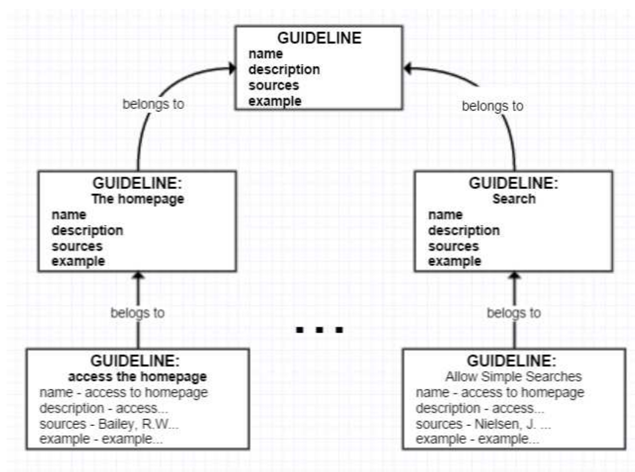

Fig. 5: Frame-based model of IS

One of the disadvantage of frame model is that there is no output mechanism, i.e. we can't represent and transfer data to HCI engineer. That's why we have to use the production model (output mechanism) to solve this issue.

2.4 Production model representation

Production model [4] can be considered as the most widespread model of knowledge representation. Production model is a model based on rules allowing to present knowledge in the form of sentences like:

$$IF < condition > THEN < action > \tag{1}$$

In general, the production model can be represented as following:

$$N = < A, U, C, I, R > \qquad (2)$$

where N - name of production, A - application of production, U - application condition of production, C - core of production, I - postcondition of production, R - comment, informal description of production.

As we described above intelligent system includes knowledge base, working memory and the interpreter of the rules (solver) which implements a specific inference engine. Any production rules contained in knowledge base consist of "antecedent" and "consequent" parts, where "antecedent" is the premise of rule (conditional part) which consists of elementary propositions connected by logical connectives "and", "or", and "consequent" (conclusion) part includes some sentences that express the fact or reference to specific actions which should be implemented. Production rules are usually written in the form of the antecedent-consequent [8]. To clear understand how production model works in our case let's consider the following example:

According to web usability guideline "Enable Access to the Homepage" [3], many users should return to the homepage for beginning a new task or starting a task over again. To meet the guideline requirements it should be existed an easy and obvious way for users to quick return to the homepage of the Web site from any point in the site. To cover such kind of guideline, web application should have container with class named *"logo"* and ¡a¿ tag with homepage link. Final code snippet of web application can be the following:

```
<div class="logo">
    <a href="website_url">
        <img src="/img/logo.png">
    </a>
</div>
```

We may suppose that our system got keywords (input information):

```
class="logo" & href="website_url"
```

There are two ways for output data:

First variant:

Step 1: Use only R1, don't work - need more data.

Step 2: Use R1 and R2, work - return guideline "Enable Access to the Homepage" from knowledge base.

If we write it in antecedent-consequent form it can be represented as follows:

$$\begin{aligned} &IF\ class = "logo"\ AND\ href = "websiteurl" \\ &THEN\ output\ guideline\ "The\ Home\ Page" \\ &ELSE\ need\ more\ data \end{aligned}$$

Second variant:

Step 3: use R1, return principle "class "logo" exist"

3 Intelligent system implementation tools

To implement our IS, we reviewed existing developed IS. After consideration of popular tools for creating ontologies: Protege, created at Stanford University, HOZO , developed by a group of organizations in Japan, OntoStudio, owned by the German company named ontoprise GmbH, our final choice was frame-based ontology editor Protege-Frames for the following reasons:

1. Full conformity with selected models of knowledge representation: support ontology-based frames and production component by integrating with CLIPS language.
2. Protege-Frames linking functionality with developed methodology of ontology [10].
3. Modular extensible architecture editor and ability import and export of data (XML, HTML, RDF, OWL).
4. Protege supports the concept of teamwork with several users (collaborative Protege) [10], which would be useful in our web application.
5. The editor is open source and free product. It is in active development and has extensive documentation.

According to Google Schoolar statistics [11], there was 2500 citations to Protege in 2012.

Protege-Frames editor uses MF ontology language [12] (Knowledge Interchange Format). As the basic relationships between classes Protege uses hierarchical R_{ISA} with inheritance. Other relationships (R_{ASS}) are implemented through the slots, which can be D_{class} or $D_{instance}$. The Protege has an option to edit metaontology [13] that allows to change meta-frames such as *"class"*, *"slot"*, *"attitude"*, etc.

Protege has a number of modules, which implement ontology representation through graphical visualization: Ontoviz, Jambalaya, TGVizTab and others. For example, Ontoviz module is able to represent not only Classification, Composition and Relation Schematics from IDEF5 notation [14], but also slots and their implications for frame classes and instances.

With the help of CLIPSTab special module, ontology has been transformed into an object-oriented data structure. CLIPS environment allows the implementation of intelligent systems based on production modeOne of the component which help us to build output mechanism is Semantic Web Rule Language (SWRL) [12]. It allows users to write rules that can be expressed in terms of OWL concepts to provide deductive reasoning capabilities than OWL alone. To represent the syntax of SWRL, consider the following example:ms of OWL concepts to provide deductive reasoning capabilities than OWL alone. The using of the syntax of SWRL we can demonstrate on the following example:

$$rec.hasClass(\$input, logo) \wedge rec.hasLink(\$input, \$url) \rightarrow rec.show(TheHomePage)$$

After running *Jess* [12] (SWRL rules interpreter), system try to find *logo* class and *url* from crawled HTML data ($input varible). As the result of interpretation "The Home Page" recommendation will be displayed.

Unfortunately, the possibility of Protege and CLIPS in organization of user interface is very limited (only test mode), but there is a possibility of launching output mechanism through web server running in Unix. This allows sharing with modern languages to create dynamic web pages (for example PHP) and the organization IS web interface. That's why to represent GUI for HCI engineer we used PHP Yii framework - one of the widespread framework based on MVC. Moreover Yii framework has special extension which give us an opportunity to communicate with knowledge base created in Protege. Graphical relationships between Protege (Knowledge base) and Yii framework (User interface) is represented in Fig. 6.

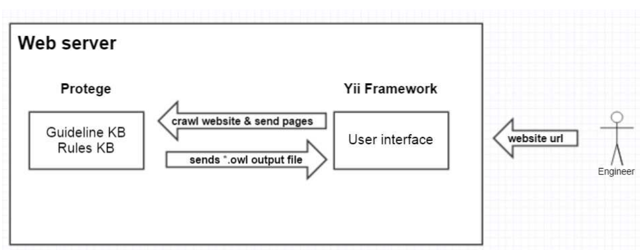

Fig. 6: Protege and Yii framework relation

4 Intelligent system functionality

During the development of IS ontology and Knowledge Base we created more than 300 recommendations with detailed description and examples coming from different sources and this list can be extended. Most of guidelines created by famous experts in web usability such as J. Nielsen, D. Norman, R. Bailey, R. Levine, T. Tullis and other.

Main classes involved in the system and their relationships are included to ontology of IS (see Fig. 2).

Total workflow of engineer with intelligent system divided by several steps:

Step 1: Analysis of software applications:

1. HCI engineer define url which should be tested for usability.
2. UI crawl all public pages of web application and send it to intelligence system as HTML files.
3. System creates an instance of new project (class *HCI engineering task*), where input information filled with slot values.

Step 2: Interface engineering:

1. System analyze the text of requirements by extracting terms presented in the dictionary. The result of analysis is a set of basic terms (B), which describe requirements of software product.
2. System send basic terms to knowledge base of rules (production model). Basic terms pass through all rules and if they satisfy one or more rules contained in KB, system create *Guideline* instance.
3. System generates OWL file which contains a set of recommendations (output information), ordered by relative importance, strength of evidence and comparative effectiveness (determined on the basis of the recommendations impact). After definite weight of generated guidelines, system create an instance of *Interface quality metric* to represent usability level. Usability level can be calculated by following equation:

$$UL = \frac{\sum_i^N s_i}{\sum_i^N s_i + \sum_i^N n_i} \quad (3)$$

where:
s_i is RI of each success recommendation
n_i is RI of each needed recommendation
4. Finally, system sends generated OWL file to UI

Step 3: Output implementation:

1. Engineer get a list recommendations (with detailed description, sources, examples etc.) and usability level of web application
2. If the number of guidelines or usability level value do not satisfy engineer's expectations, he is able to pass questionnaire to get more recommendations, which IS unable to determine (for example, the color of background image and color of text).
3. After passing questionnaire, UI sends output information intelligence system as XML files
4. system repeats steps I and II.
5. Engineer is able to learn all recommendation contained in knowledge base to improve web application manually.

Step 4: Interface evaluation:

1. Levels of usability (*usability level* slot of *Interface design* class) is stored in IS as the values of the quality of interface (*Interface quality metric* instances). This way IS allows to track usability changes of web

application, depending on embodied design decisions, i.e. HCI is able to test its web application again improve the usability level.
2. Based on the quality indicators of each interfaces, stored in IS can be automatically calculated the comparative effectiveness of recommendations.

5 IS for HCI engineering practical examples

As one of the practical tasks to assess the applicability of IS, we selected five web application of different companies and organizations:

1. Kazakh British Technical University
2. Sayer LLP
3. ALSI LLP
4. Ships2day LLP
5. Kivvi web app.

Evaluation of websites usability we divided into two steps:

Step I: Automatic analysis of the structure of web application.

Step II: Manual analysis by questionnaire.

We may use web application url as input information. Using url IS runs spider, implemented in php to extract public pages of website. Then IS generates a list of recommendations: recommendation chapter, detailed description, its sources, graphical examples and current usability level of web application.

To improve the usability level and number of recommendations, we are able to pass the questionnaire. At this stage, the input information for solver will be answer of questionnaire. Such kind of solutions needs to generate recommendations which solver cannot solve. For example, solver is not able to analyze the color of main text and background image behind them.

It should be noted that the questionnaire do not guarantee that the increasing of usability level. It can be if our answers will be negative related to website usability. According to equation 3, each negative answer from questionnaire increase the value of ni. Consequently, the value of usability level will decrease. One of the convenient feature of IS is that it saves results of the project, so if we try to test web application after refactoring, we are able to assess the improvement level before and after reconstruction. Intelligent system is available on hci-test.info and results shown in table 1.

Table 1: websites usability level

website url	UL on step I	UL on step II
kbtu.kz	59%	67%
sayer.kz	64%	74%
alsi.kz	47%	64%
ships2day.kz	30%	55%
kivvi.kz	56%	71%

The average success rate assignments obtained in experiment is equal to 66.2% after applying questionnaire. So, Jakob Nielsen noted the value of this index at the level of 52.9% [15], but since then it would be naturally to expect its increase due to general increase in level of usability of web applications. Tasks related to data entry caused greater complexity than search for information (success rate 59.4% and 67.2% respectively), which also confirms general validity of results.

Based on results of usability testing (analysis of user interaction with web applications) our IS proposed following main changes to web interface projects:

1. Increase number of text explanation and simplifying them. For example, web forums should be described at basic level and add clarification to each sections in site map.
2. Add changes to register form: simplify control of automatic registration, creation of the page with confirmation of successful registration.
3. Further development of site navigation: removal of some items in sidebar (to improve readability), add titles and descriptions of sections and pages.
4. Modification of some styles on the site for standardization. For example, the color of *"breadcrumbs"* should match to color of the remaining links.
5. To facilitate learning, use images rather than text whenever possible. The superiority of pictures over text in a learning situation appears to be strong. For example, pictures of common objects are recognized and recalled better than their textual names.

We made appropriate improvements to three web applications and our final results of usability level is shown in table 2.

Table 2: websites usability level after improvements

website url	UL on step I	UL on step II
kbtu.kz	59%	67%
sayer.kz	64%	83%
alsi.kz	47%	76%
ships2day.kz	30%	81%
kivvi.kz	56%	71%

6 Conclusion

Using such kind of intelligent systems facilitate engineer to solve problems related to web usability. Intelligent system require only website url which need to assess and improve. Another advantages of IS is knowledge base which would be useful for HCI engineer with detailed information about recommendations. KB contains about 300 recommendations with description, sources and examples and this list is extending. The more recommendations, the higher usability level we may reach corresponding to web applications. In previous paper [1], we defined three expert knowledge in HCI: laws, principles and recommendations. Future development will cover laws and principles, which haven't implemented yet. There is an opportunity to integrate Fitts and Hicks laws, which consider physical and cognitive human aspects.

Acknowledgement

This work was supported by the Ministry of Education and Science of the Republic of Kazakhstan for the project "Development and implementation of national software for solving technological problems in Oil and Gas industry in Kazakhstan" in a section "Development of Expert Systems". The authors are grateful for Kazakh British Technical University for the use of their laboratory facilities and library resources.

References

[1] Sheriyev M. N., Atymtayeva L.B.Automation of HCI Engineering processes: System Architecture and Knowledge Representation, Advanced Engineering Technology and Application. Vol. 4, No. 2 (May 2015), PP:41-46

[2] Mirseidova Sh., Atymtayeva L. Definition of Software Metrics for Software Project Development by Using Fuzzy Sets and Logic Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems, 20-24 November 2012, Kobe, Japan, pp. 272-276

[3] O. Leavitt, B. Shneiderman, Research-Based Web Design & Usability Guidelines, Washington, DC, 2014.

[4] Bakaev M., Avdeenko T. Ontology to Support Web Design Activities in ECommerce Software Development Process. In Proc. of IASTED ACIT'2010, Novosibirsk, Russia. ACTA Press, 2010. - P. 241-248.

[5] Atymtayeva L., Kozhakhmet K., Bortsova G. Building a Knowledge Base for Expert System in Information Security. Proceedings of the 14th International Symposium on Advanced Intelligent Systems (ISIS2013), 13-16 November 2013, Daejeon, Korea, Springer Journal "Advances in Intelligent Systems and Computing", Volume 270 "Soft Computing in Artificial Intelligence ", pp. 57-77

[6] Gruber, T.R. A translation approach to portable ontologies. Knowledge Acquisition, 5 (2), 1993. pp. 199-220.

[7] Y. Zhou, T. H. D. Dao, J. C. Thill, E. Delmelle, Enhanced 3D visualization techniques in support of indoor location planning, Elsevier, 50, 15-29 (2015)

[8] Guarino, N. Formal Ontology in Information Systems. In Proc. of FOIS'98, Trento, Italy. Amsterdam, IOS Press, 1998. - P. 3-15.

[9] M. Minsky, A framework for representing knowledge In The Psychology of Computer Vision, McGraw-Hill: P. Winston, 1975. pp. 211-277.

[10] Tudorache T., Noy, N.F., Musen, M.A. Supporting collaborative ontology development in Protege. In Proc. 7th Intl. Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, 2008. pp. 17-32.

[11] Xu R., Dai X., Yang, F. Lin, P. Research on the Construction Method of Emergency Plan Ontology Based-on OWL. In Proc. of the 2009 International Symposium on Web Information Systems and Applications (WISA A Z09) Nanchang, China, 2009 P. 19-23.

[12] Natalya F. Noy and Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology. In Proc. Semantic Web Working Symposium, 2001.

[13] Gavrilov T.A., Khoroshevsky V.F.: Knowledge base of intelligent systems. St. Peterburg, 2000. pp. 384.

[14] Benjamin, P. et al. Idef5 method report. Technical Report F33615-C-90- 0012, Knowledge Based Systems Inc, Texas, 1994. pp 187.

[15] Nielsen, J. Usability Engineering. Morgan Kaufmann, San Francisco, USA, 1994. - 362 pp.

Madi Sheriyev received the master degree in Information Systems at Kazakh British Technical University in Almaty, Kazakhstan. The Author has been pursuing his research work under the guidance of Lyazzat Atymtayeva, Doctor of Science degree in Mechanics, Mathematics and Computer Science. His research Interest includes Human Computer Interaction, Artificial Intelligence, Semantic Web and functional programming.

Lyazzat Atymtayeva received the PhD and Doctor of Science degree in Mechanics, Mathematics and Computer Science at al-Farabi Kazakh National University, Kazakhstan. Her research interests are in the areas of mechanics, applied mathematics and computer science including the numerical and rigorous mathematical methods and models for mechanical engineering and computer science, intelligent and expert systems in Information Security, Project Management and HCI. She has published research papers in reputed international journals of mathematical and computer sciences. She is reviewer and editor of international journals in mathematics and information sciences.

Iskander Beissembetov received the PhD and Doctor of Science degree in Economics at Moscow State University of Economics, Statistics and Informatics, Russia. He published about 30 research papers related to Petroleum Engineering, Management, Mathematics and Cybernetics. From 2004 Mr. Beissembetov is Rector at Kazakh British Technical University.

Bagdaulet Kenzhaliyev received the PhD and Doctor of Science degree in Physics and Mathematics at S. Kirov Kazakh State University, Kazakhstan. His research interests in coordinating efforts to attract investment for the development of the research base of the university, organizes the work on the development of fundamental and applied projects and programs, implementation of innovative development to implementation in production, which are of great interest in the near and far abroad. Now Mr Kenzhaliyev is Vice-Rector for Innovation at Kazakh British Technical University.