ONTOLOGY TO SUPPORT WEB DESIGN ACTIVITIES IN E-COMMERCE SOFTWARE DEVELOPMENT PROCESS

Maxim Bakaev *, Tatiana Avdeenko *
* Novosibirsk State Technical University
Pr. K. Marksa, 20, Novosibirsk, 630092
Russian Federation
maxis81@gmail.com; tavdeenko@mail.ru

ABSTRACT
The paper suggests an application of knowledge engineering methods to support web design-related decisions and enhance usability for e-commerce software products. A prototype of a frame-based ontology was designed with Protégé editor to integrate somehow scattered and discrepant knowledge in the fields of HCI, usability engineering, web design, etc. Knowledge discovery and information extraction are demonstrated for a sample target group of elder users, resulting in formal rules and text-based guidelines. The developed ontology could be used in an expert system or for training to improve e-commerce software quality. Considerations for online publishing of the ontology for further knowledge discovery, validation and formalization are provided.

KEY WORDS
Software, web design, artificial intelligence, usability

1. Introduction

E-commerce currently has various definitions, but in a narrowed, practical sense it is often understood as e-trade, that is, activities related to marketing or selling goods and services online. The share of online commerce in overall retail trade is growing worldwide, and though in Russian Federation in 2008 it amounted to relatively small value of 1%, the net volume has been increasing rapidly, with annual average growth of 30%, and came up to about $4.2 billion in the same year of 2008 [1].
Usability and user interaction quality are recognized as major factors for e-commerce software to succeed [2]. However, web usability still remains an obstacle for e-commerce projects' success, although some improvement is noted lately. So, even such simple metric as "success rate" for e-commerce websites in the USA was about 56% in 2001 [2, p. 5], already 66% in 2004 and 81% in 2009 [3]. However, for user groups with special needs the web interaction quality remains even lower: e.g. a research undertaken in 2002 reported that success rate for elder users with websites was almost 1.5 times lower than for younger control group [4].

1.1 E-Commerce Software Development Process

To explore the possible reasons for the current problematic situation with web usability in e-commerce, we started with analyzing the respective software development process. In the creation of e-commerce websites, the same principal stages may be identified as in the "classic" waterfall or iterative processes of software development (see e.g. [5]):

- Requirements analysis
- Architecture and design
- Implementation (coding)
- Testing and debugging
- Deployment and maintenance

Although some e-commerce software may be complex and unique, smaller corporate websites or e-shops tend to have a lot of common features and often use typical solutions. Below we present a brief review of the stages of software development process, considering distinctive properties of e-commerce websites creation and paying special attention to applicable human-computer interaction (HCI) and usability engineering methods.

1.1.1 Requirements analysis

The essence of this stage of the development process is gathering, analysis and specification of requirements for the product from all the involved stakeholders, including, of course, the users. The result of this phase is usually a set of documents that, as fully as possible, describe the product's behavior to meet the goals and constraints of the project.
To find out the needs of the future users of the software, their personal and professional traits as well as the context of use, such methods as user observation, surveys and interviews, focus groups, etc. are commonly used. However, in e-commerce software projects, future visitors and customers are generally not directly accessible to the development team, which makes using the above methods problematic or impossible. A possible solution is utilization of the so-called personas, fictitious characters embodying typical users' characteristics [6], the creation of which is yet another task during the requirements analysis stage. Another well-known and powerful method is

employment of use cases – specifications of actions that user can make with the system.
There are certain software tools that can support development activities of the requirements analysis phase: for instance, *IBM Rational RequisitePro* or *IBM Rational Requirements Composer*. Also, *MS Visio* or *IBM Rational Modeler* can be used to create specifications in UML, e.g., for use cases. However, we are not aware of tools taking into account characteristics and needs of a software product's target users to specify non-functional, usability requirements and at the same time providing pre-built use cases depending on the project business goals.

1.1.2 Architecture and design

During architecture and design phase, developers, based on requirements specified at the previous stage, define the product's internal properties and further work out the details of the external ones. The software system's architecture, its components' structure and user interfaces are designed. Accordingly, for e-commerce websites the following must be specified:

1. The website content – the list of sections and services, their substance and how they interact with each other.
2. The website design – in a narrowed sense, the structure and visual appearance of website user interface, which is a collection of web pages.
3. Implementation tools – web server, database, programming language or development environment, existing modules or libraries, etc.
4. Finally, refined plan of the project – contains the schedule of the tasks for the project's next stages and a list of team members who are going to execute the tasks.

Understandably, the main focus of HCI on this stage is on the design of structure and visual appearance of web pages. Among the applicable methods, the following two groups may be highlighted: 1) utilization of existing guidelines; and 2) prototyping. The former implies that developer, when designing interfaces or web pages, is grounded on models, recommendations or existing solutions. However, given the diversity of users and business goals of e-commerce projects, the search for guidelines relevant to the current development context may turn out to be quite time-consuming, and their correct understanding and application may require highly professional design and HCI engineering skills. The methods of the second group, ones that relate to prototyping, imply that a simplified interface is created and tested with future users. In practice, design prototypes are created on paper, or as a static image, with a graphic editor, such as *Adobe Photoshop*, or as an interactive application, e.g. with Flash technology.
It should be noted that the above editors, as well as existing dedicated web design tools, such as *Adobe Dreamweaver* or *MS Expression Web*, only supply developer with technical means to implement his or her interface design decisions. However, neither of them supports the core, HCI side of the interface design process by providing context-dependent guidelines relevant to target users' characteristics.

1.1.3 Implementation (coding)

At this stage, the programming code of the product is created, with implementation tools selected at the previous phase of the development process. In particular, code for user interface is produced, and it was estimated that on average it amounts to 50% of all code written for software applications [7]. Despite that, coding activities generally lie outside of HCI focus, because if interfaces or web pages were designed at the previous stage in sufficient detail, their implementation in code is a reasonably straightforward task.

1.1.4 Testing and debugging

At this stage, errors are discovered in the product during testing process, their severity is estimated, and, if this is considered worthwhile, they are removed from the software, which is said to be debugged. Generally, error is a product's behavior that doesn't correspond to the requirements or, as specifying completely exhaustive list of requirements is impossible, to certain accepted quality standards for software.
Testing stage doesn't necessarily come strictly after coding, and the use of HCI and usability engineering methods, such as heuristic evaluation, usability testing, etc., early in the development process is widely advocated [8]. It should be noted that the efficiency of automated testing tools for evaluating user interfaces is understandably lower than for general code assessment. Thus, thorough testing of web pages designed in an e-commerce software development project is possible only with participation of experts and users.
That said, testing of certain HCI-related aspects still can utilize automation tools. For instance, as e-commerce website is, by definition, a multi-user application, special attention must be paid to its reaction time (i.e. web page loading time), which should equal to about 1 second, while barely acceptable limit is considered to be 10 seconds [9]. In a process of load testing, aimed on finding out website's response times per number of simultaneous visitors, various tools emulating the load can be used, such as *jmeter*, *http_load* or *apache benchmark*.

1.1.5 Deployment and maintenance

Deployment is measures aimed on letting user start working with a product, while maintenance relates to activities concerned with software enhancement, its adaptation to new environment and errors correction. Unlike the previous stages, this phase generally has no planned fixed duration and may last for as long as users continue utilizing the product. It is estimated that about 60% of the entire software development cost falls on this stage, and most of the activity is concerned not with error fixing, as it is sometimes mistakenly believed, but with product's improvement and further development [5].

After an e-commerce website is opened for visitors, developers finally obtain firm access to real users and potentially unlimited capabilities to analyze their behavior, to adjust understanding of their needs and to evaluate the successfulness of current design, so that directions for website enhancement can be identified. It is possible to fully or partially automate some of the methods researching users' interaction with web interfaces: by analyzing visitors' paths on the website, performed tasks, emerging errors, etc. (for more complete review, see e.g. [10]). This is realized in numerous existing tools for automated usability and accessibility evaluation (see [11]), a certain drawback of which, however, are remaining difficulties in taking into account the characteristics of various target user groups.

1.2 Ensuring Usability in E-Commerce Web Design

It is already an accepted fact that usability and web design quality are major factors in success of e-commerce products [2]. At the same time, it is widely recognized that in software development the most costly mistakes are the ones made on the initial stages of the project. There are estimations that if an error was committed on the architecture and design phase, the cost of fixing it during testing would increase 15 times, and after release – up to 100 times [12].

However, the above review of the e-commerce website development process suggests that, in terms of HCI and usability engineering, the fewest support by automation tools for software developers is provided during requirements analysis and design stages. Existing development tools do offer capabilities to build interaction models (e.g. with UML) or interface prototypes, but they neither provide business goal- or target user-dependent guidelines, nor are able to evaluate the result of the design.

Such support is deemed especially necessary for e-commerce website building, as projects in this field have relatively lower budget and the development team may lack interface and web designers as well as usability engineers with sufficient professional skills. Actually, even highly skilled design specialists may have to spend significantly more time and effort on their tasks if target users come from a group with special needs, such as elder or disabled people. The reason for that is the ever-persistent designer-user gap, which gets wider as the difference between the designer's and user's characteristics, goals and use context amplifies, leading to increased necessity for systematic integration of HCI and usability engineering into software development process [13]. In reality, however, the gap was rather noted (see [14]) between common recognition of the importance of usability (92% of companies surveyed) and actual application of its engineering methods (67-72%).

In this paper we propose a knowledge engineering approach to create a tool providing support for developers in their web design-related activities in e-commerce software development projects, with special focus on HCI and usability engineering. There is no doubt that such system must extend, but not replace recognized and effective interaction engineering methods, such as heuristic evaluation or usability testing. Nevertheless, it could be able to alleviate requirements towards developers' professional skills, decrease work effort required for design as well as the probability of emergent errors.

2. Ontology to Support Web Design Activities in E-Commerce Software Development

2.1 Method

Currently, there is a significant body of knowledge accumulated – guidelines, models, recommendations – regarding software interface design and web design in particular. For instance, an impressive work on the matter was published by the U.S. Department of Health and Human Services [15], while Nielsen Norman Group contributed a thorough report related to e-commerce domain [2].

Unfortunately, the knowledge built up in the field of design and usability engineering does not seem to be organized and structured well enough. For example, the above mentioned impressive collection of guidelines [15] in terms of knowledge representation only provides a tool sorting the records by title, importance, etc. (http://www.usability.gov/pdfs/chap.html). The relatively poor knowledge organization in the area may be one of the factors hindering proper and inclusive use of guidelines by practitioners, leading to decreased usability of interfaces designed.

At the same time, in the field of artificial intelligence there are technologies already developed for knowledge representation and its use in various domains. Ontologies and expert systems had been created both on a global scale (e.g. CYC at http://www.cyc.com) and for such diverse areas as biology and medicine, linguistics and education, business and management, and even e-business modeling [16]. So, we presume that knowledge engineering approaches can be used in creating a knowledge-based system (KBS) supporting interface and web design by incorporating even plentiful and often weakly formalized guidelines and providing them on a use context basis.

2.2 Ontologies in knowledge engineering

The so-called knowledge-based systems are generally built so that declarative knowledge, i.e. factual information that is more or less static, is separated from procedural component that is responsible for reasoning and inferring. There exist several models for knowledge representation: production model, formal logic-based, frame-based, semantic network, etc. One of the up-to-date means in the field are ontologies – formal representations of a set of concepts within a domain and the relationships between those concepts [17]. Several knowledge representation models fit the definition of ontology: frames, semantic networks, concept maps and so on. Even a subset of predicate logic syntax can be used to describe ontology.

A possible formal definition of ontology is:

$$O = \{C, R, A\},$$

where O is ontology, C is a set of concepts (classes) of a domain, R – relationships between the concepts, while A is a set of axioms, i.e. assertions and rules, that describe the laws and principles that govern the concepts existence in the domain. Ontologies can be categorized: 1) by relationships between concepts: taxonomy, partonomy, genealogy, etc., 2) by goals or domain: education, management, marketing, e-business, knowledge management, etc., 3) by owner, 4) by ontology language, and so on. Graphic representation is a popular way to display ontology, and there is a common consent to place more general concepts, such as categories or classes, above, while more concrete ones, such as sub-categories, objects and instances, are positioned below.

Ontology that contains instances of classes becomes knowledge base (KB) for KBS, however it's actually challenging to define where ontology ends and knowledge base begins. Knowledge bases that store knowledge in logically consistent and computer-readable form and may have automated reasoning applied to them are called machine-readable; however there are also human-readable knowledge bases, whose purpose is allowing people retrieve and use stored knowledge, which doesn't have to be fully formalized. Populating KB with knowledge may also occur in two manners: 1) a human knowledge engineer extracts knowledge from a domain expert and then formalizes it, e.g. as an ontology axiom, or 2) an automatic knowledge retrieval process is applied to relevant sources, with subsequent information extraction (IE), which also results in machine-readable ontology rules. Up-to-date approaches to IE utilize natural language processing techniques, such as the so-called text mining (see [18] for example).

2.2.1 Frame-based ontologies

To represent knowledge in HCI and web design, we consider it feasible to use ontology that has taxonomy-like relationships between classes and includes capabilities of frame model as described by M. Minsky [19]. Frame, as any other ontology, is represented as a network consisting of nodes and links between them, and "higher levels" of the frame are formed by general concepts (classes). "Lower levels" of the frame contain the so-called terminals or slots that will need to be filled with actual data or examples. A particular feature of the frame model is that initially the slots may be either not filled or filled with default assignments, not necessarily proper for a given context. An advantage of the frame model is that as the initial model adapts to circumstances, the default assignments are gradually replaced with data that corresponds the most to the current situation.

This procedure can be carried out at the design stage of an e-commerce software development project, in accordance with specified business goals and target users' characteristics. As for the future use of a KBS built on the ontology, it would be feasible to represent it as a scenario. Scenario or script, according to [20], is a structured representation describing stereotypical order of events in a specific context. As the scenario executes, website target users, purpose, content, visual appearance are clarified in more and more detail, and in the result, design guidelines or design prototype are produced by the system, considering the ascertained context of the project. Creating scenarios based on frame model [19], allows ensuring flexibility necessary to take proper account of various, unforeseen situations.

The construction of functional ontology for a complex field is generally a labor-intensive and iterative process, possibly involving several domain experts, therefore in the current paper we propose the design for an ontology prototype and outline its utilization to support web design activities.

2.3 The ontology prototype design

2.3.1 The scope of the ontology

As mentioned before, we develop the ontology to be used in knowledge-based systems supporting web design for e-commerce via providing context-dependent rules and guidelines. As such, the ontology may contain knowledge related to HCI, interface design, graphic design, usability engineering, etc. Practical knowledge related to these fields appears to mostly exist in a form of guidelines or recommendations, at least some of which don't have strictly objective and quantitative nature or may even contradict each other. Consequently, it is deemed feasible to settle that the results our ontology provides to users regarding web design are not fully formalized and do require users to make ultimate decisions. Thus, some of the domain knowledge will be transformed into ontology rules, while another part should result in a set of possibly informal guidelines that are left to be interpreted by human web designer who uses the system. Later, however, as the system becomes operable and starts to receive data and feedback from users, information extraction methods may be used to convert the knowledge of the second group into ontology axioms and rules.

The analysis of e-commerce software development domain makes us believe that there are two most important factors influencing web design decisions: business goals of the project and characteristics of target users. Proceeding from them, it should be possible to infer results divided in the two groups: 1) formalized knowledge embodied in web pages template(s) and 2) non-formal knowledge, as relevant design guidelines. Then, the ontology could be used as outlined on Figure 1.

2.3.2 Ontology editor

Currently, there are probably dozens of ontology editors available, among most notable of them being Protégé by Stanford University (http://protege.stanford.edu/), HOZO that is jointly developed by several organizations in Japan (http://www.hozo.jp/), OntoStudio by Ontoprise GmbH (http://www.ontoprise.de) and many others. A review, though somehow dated, can be found in [21], while

ontology editors' features are additionally explored in [22]. To create ontology for our project, we elected Protégé because it: 1) supports frame-based ontologies (in Protégé-Frames) and provides high level of abstraction, 2) has rich integration and import/export capabilities, 3) is free and open-source, 4) is in active development, has detailed documentation and reasonably easy interface. Moreover, via CLIPSTab plug-in (see [23]) Protégé can be integrated with CLIPS rules and inference engine.

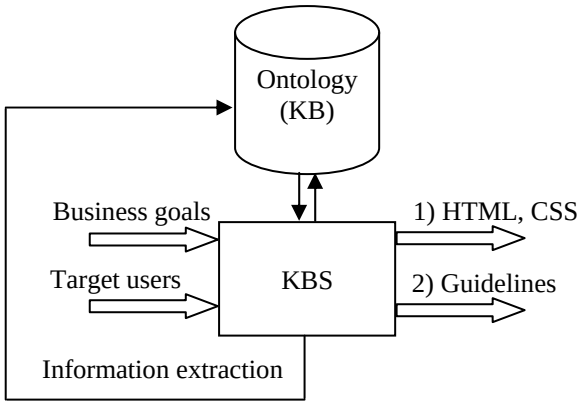

Figure 1. The use of the ontology in a knowledge-based system

2.3.3 The structure of the ontology

Initial brainstorming sessions with experts produced a list of 109 concepts related to the domain. Then, based on this list, the top-down design process was carried out for the frame-based structure of the ontology that in the end incorporated 54 classes and 97 slots (not including extra 15 system classes and 34 system slots). The ontology classes' hierarchy is shown on Figure 2.

Thorough literature review was undertaken to define slots for class *Target user*, and two groups of factors were identified: 1) personal user characteristics, such as age, gender, experience, education level, nationality, and 2) use context issues, such as browser type, screen resolution, installed software and plug-ins – these were grouped as "software environment" factor, internet access type, etc. Figure 3 shows template slots for class *Target user*, though later more properties may be introduced.

For the *Guideline* class, it was necessary to allow organization of guidelines so that they could be presented to potential ontology user on a context basis. For that, we decided to introduce tag-based classification, where a *tag* could be any existing class in the ontology. At the same time, *related guideline* slot would allow linking interconnected guidelines. Also, as the reasoning process in KBS would need to produce only guidelines relevant to a framework of a specific software project, the class *Guideline* would need to have *applicability* slot indicating instances that are affected by the guideline. Finally, the *source* slot would point to an origin of guideline – HCI model, law, experiment or heuristics. The slots for *Guideline* class are presented on Figure 4.

In the next chapter, we would like to show how the ontology could be populated with instances, rules and guidelines to be used in practice by a web designer via a KBS. As example target users, elder people were elected, as a user group that exhibits significant growth but so far remains considerably discriminated in terms of usability, accessibility and, consequently, overall web and IT usage.

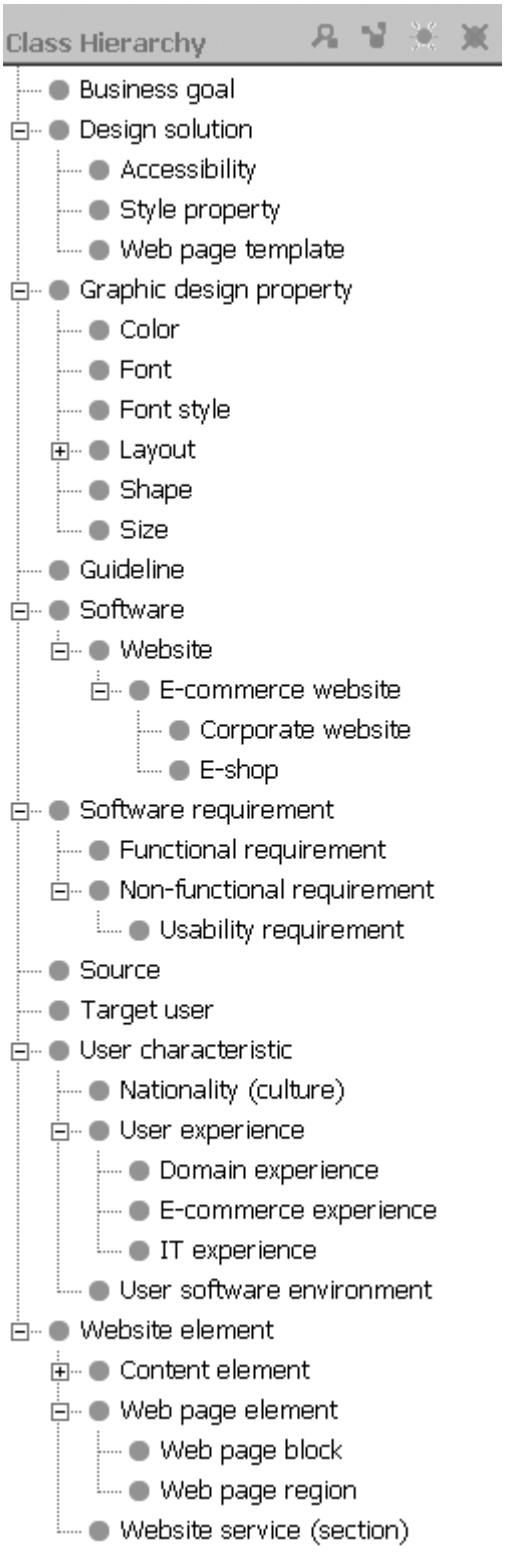

Figure 2. The hierarchy of classes in the ontology built with Protégé editor

2.4 Populating and using the ontology

2.4.1 Eliciting target user

First, an instance of *Target user* class was created, implying elder users and named as *TU1*. The slots were set in the following way:

- *name* = "Elder user"
- *age* = 60
- *context of use* = "From home computer, possibly without first-hand support from a younger, more advanced computer user"
- *education level* = "college"
- *gender* – not set
- *internet access* = "cable"
- *nationality (culture)* = Russian (default assignment)
- *software environment* = Internet explorer browser, 1024*768 screen resolution
- *user experience* = *IT experience* is low, *E-commerce experience* is low

Template Slots

Name	Cardinality	Type
age	single	Integer
context of use	multiple	String
education level	single	Symbol {none,high_school,college}
gender	single	Symbol {Male,Female}
internet access	single	Symbol {dial-up,cable,broadband,local_
nationality (culture)	single	Instance of Nationality (culture)
software environment	multiple	Instance of User software environment
user experience	multiple	Instance of User experience

Figure 3. Template slots for class *Target user* built with Protégé editor

Template Slots

Name	Cardinality	Type
applicability	multiple	Instance of :THING
related guideline	multiple	Instance of Guideline
reliability	single	Float
source	multiple	Instance of Source
tag	required multiple	Class with superclass :THING
text (english)	single	String
text (russian)	single	String

Figure 4. Template slots for class *Guideline* built with Protégé editor

2.4.2 Website content

Before designing the structure and visual appearance of web pages, the website content, i.e. its services and sections, must be defined. After a knowledge discovery session ran with a human domain expert, the rules inferring website content from business goals and target users' characteristics were contrived. First, we needed to create several instances of *Website service (section)* class in the ontology, so that some of the instances would be elected in the process of defining the slots of *E-commerce website* frame. The following instances of *Website service (section)* class were created for the current iteration:

WSS1. Homepage
WSS2. About Us
WSS3. Contact Us
WSS4. Q&A
WSS5. Forum
WSS6. Services
WSS7. Catalogue
WSS8. Shopping cart
WSS9. Back-office
WSS10. Payment service
WSS11. Delivery service
WSS12. Search
WSS13. Sitemap

Then, the following instances of *Business goal* class were created:

BG1. Presenting information about company or brand online.
BG2. Presenting information about offered goods or services online.
BG3. Selling goods or services online.

Finally, formal rules were contrived as an example:

R1. **if** *(BG1)* **then** {*WSS1*; *WSS2*; *WSS3*; *WSS4*; *WSS5*;} /* corporate website */
R2. **if** *(BG1* **and** *TU1)* **then** {*WSS1*; *WSS2*; *WSS3*; *WSS4*; *WSS13*} /* elder people are unlikely to need forum, but may benefit from extra aid in navigation */
R3. **if** *(BG3* **and** *TU1)* **then** {*BG1*; *BG2*; *WSS1*; *WSS2*; … *WSS13*;} /* e-shop */

2.4.3 Website design - HTML and CSS

As mentioned above and shown on Figure 1, web designers working with the ontology via a KBS should be able to obtain the formal knowledge summarized in HTML and CSS code. A possible way to implement this in a KBS would be with a WYSIWYG editor, allowing its user to:

- manage website content (instances of *Website service (section)* class) and web page regions (instances of *Web page region* class),
- position web page blocks (instances of *Web page block* class) in web page regions,
- manage applicable graphic design properties, such as color, font, font style, size, etc. (instances of respective classes) for headers, text, hyperlinks, tables and other content elements (instances of *Content element* class),
- check guidelines (instances of *Guidelines* class) relevant to the current design context: target users' characteristics, business goals, and selected objects – website elements, graphic design properties, etc.

The WYSIWYG editor would also display the current web page template (instance of *Web page template* class) with styles and design solutions applied to it, and, at the end, allow to generate program code (HTML and CSS) for the web page template.

2.4.4 Website design – guidelines

The sample guidelines were extracted from the current paper authors' previous works devoted to researching universal design and usability for elder people, as well as their special needs and preferences in physical, cognitive and emotional aspects. Two instances of *Guideline* class were created, *G1* and *G2*, linked to respective sources *S1* and *S2*, instances of *Source* class.
Setting *G1* slots:

- *applicability* = *TU1*
- *related guideline* – not set
- *reliability* – not set
- *source* = *S1*
- *tag* = *Accessibility*, *Size*, *Usability requirement*, *Content element*
- *text (english)* = "For elder users to match their younger counterparts' throughput in selection tasks, interface elements must be designed larger, about 1.42 times."

Setting *S1* slots:

- *name* = "Fitts' law, movement and selection throughput research for elder users"
- *related source* – not set
- *source reference* = (reference [24] of the current paper)

Setting *G2* slots:

- *applicability* = *TU1*
- *related guideline* – not set
- *reliability* – not set
- *source* = *S2*
- *tag* = *Business goal*, *Graphic design property*, *E-commerce website*, *Design solution*
- *text (english)* = "Web design factors that are related to "expressive" aesthetic dimension, such as original, sophisticated, fascinating, etc. may actually have negative influence on aesthetic impressions of the seniors."

Setting *S2* slots:

- *name* = "Research of elder people's aesthetic preferences for e-commerce websites"
- *related source* – not set
- *source reference* = (reference [25] of the current paper)

2.4.5 Knowledge discovery and information extraction

So far, knowledge discovery (KD) and information extraction (IE) for the developed ontology had been done without automation, with human domain experts and knowledge engineers. However, it should be noted that approaches to developing related computerized methods are made, most of them dealing with natural language processing techniques (see [18]). As disperse, informal and contradicting the sources for e-commerce web design may be, the goal of automating KD and IE should be set by knowledge engineers. For our web design support ontology, IE automation may lie in gradual transformation of guidelines into formal rules or axioms. That can be done when working knowledge base starts accumulating data on users' utilization of guidelines in actual designs and users' feedback, which may be gathered through voting for guidelines and designs, submitting comments and corrections, etc.

3. Conclusion

In the paper, we suggested an approach to support interface design and e-commerce web design in particular with domain ontology. The main idea of the approach is applying knowledge accumulated in HCI, usability engineering, web design and adjoining fields through specified target users' (personas') characteristics and project business goals. Somehow scattered knowledge in the above fields may be gathered in a frame-based ontology. The frame model for the knowledge base both allows organizing hierarchical structure for domain concepts (classes) and contributes to creating interaction scenario for user working with knowledge-based system. The scenario implements a dialogue aimed on specifying a purpose of a website being designed, its target users, etc., and the result is a web page design prototype accounting these factors and features. Frame model in this case allows attaining the necessary flexibility of the user-system interaction. Consequently, an ontology editor, Protégé, was used to build the hierarchy of classes, define relationships between them via slots, etc.
The developed ontology could be used not only as an expert system or in training web designers inexperienced in e-commerce domain, but also by skilled web and interface design engineers seeking to enhance the quality of their work by considering needs and requirements of a target user group as fully as possible. Ontology in this case may be used as a basis, because it contains easily comprehendible structure of the field. Secondly, the ontology could be used in scientific research, through providing ground for system analysis of the domain. It can be analyzed, for instance, for possible contradictions in models and rules or, vice versa, for highly important factors that need to be considered in web design. Then, by merging knowledge from various sources or other ontologies, the ontology would help establishing semantic equivalence of identical facts and concepts stated in different terms.
Finally, the ontology publishing online is planned, to accumulate empirical data of rules and guidelines applicability and significance, as well as to gather feedback from expert users. Operation of ontology by real users could also help assess if further knowledge formalization in ontology is welcomed and evaluate overall usefulness of knowledge representation methods in supporting web design activities and improving the quality of e-commerce software products.

References

[1] National Association of E-Trade Agents (Russia). *Press release issue (16.03.2009)*. Available at http://nauet.ru/press.php?p=1&sub=4&news=102. Accessed 01 Feb 2010.

[2] J. Nielsen, C. Snyder, R. Molich, S. Farrell, *E-Commerce User Experience* (Fremont, CA, USA: Nielsen Norman Group, 2001).
[3] J. Nielsen, IA Task Failures Remain Costly, *Alertbox, April 16, 2009*. Available at http://useit.com/alertbox/ia-failures.html. Accessed 01 Feb 2010.
[4] J. Nielsen, Usability for Senior Citizens, *Alertbox, April 28, 2002*. Available at http://useit.com/alertbox/seniors.html. Accessed 01 Feb 2010.
[5] R.L. Glass, *Facts and Fallacies of Software Engineering* (Reading, MA, USA: Addison Wesley, 2000).
[6] A. Cooper, *The Inmates are Running the Asylum* (SAMS, 1999).
[7] S. Douglas, M. Tremaine, L. Leventhal, C. Wills, & B. Manaris, Incorporating human-computer interaction into the undergraduate computer science curriculum. *Proc. of the 33th SIGCSE technical symposium on Computer science education*, 2002, 211-212.
[8] J. Nielsen, *Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier*, 1994.
Available at http://useit.com/papers/guerrilla_hci.html. Accessed 01 Feb 2010.
[9] J. Nielsen, Powers of 10: Time Scales in User Experience, *Alertbox, October 5, 2009*. Available at http://www.useit.com/alertbox/timeframes.html.
Accessed 01 Feb 2010.
[10] M.Y. Ivory & M.A. Hearst, The state of the art in automating usability evaluation of user interfaces, *ACM Computing Surveys (CSUR), 33*(4), 2001, 470-516.
[11] G. Brajnik, Using automatic tools in accessibility and usability assurance processes, *Lecture Notes in Computer Science, 3196*, 2004, 219-234.
[12] S. McConnell, *Code complete (2nd ed.)* (Redmond, WA, USA: Microsoft Press, 2004).
[13] J. Nielsen, Bridging the Designer–User Gap, *Alertbox, March 17, 2008*. Available at http://useit.com/alertbox/designer-user-differences.html.
Accessed 01 Feb 2010.
[14] B. Bygstad, G. Ghinea, E. Brevik, Software development methods and usability: Perspectives from a survey in the software industry in Norway, *Interacting with Computers, 20*(3), 2008, 375-385.
[15] U.S. Department of Health and Human Services, *Research-based web design & usability guidelines*, 2006.
Available at http://www.usability.gov/pdfs/chap.html. Accessed 01 Feb 2010.
[16] A. Osterwalder, Y. Pigneur, An e-Business Model Ontology for Modeling e-Business. *15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy*, Bled, Slovenia, 2002.
[17] T.R. Gruber. A translation approach to portable ontologies, *Knowledge Acquisition, 5*(2), 1993, 199-220.
[18] N.G. Zagoruiko, V.D. Gusev, A.V. Zavertailov, S.P. Kovalev, A.M. Naletov, N.V. Salomatina, ONTOGRID system for automation of domain ontologies building (in Russian), *Avtometriya Journal, 41*(4), 2005, 13-25.
[19] M. Minsky, A framework for representing knowledge, In *The Psychology of Computer Vision* (McGraw-Hill: P. Winston, 1975).
[20] R.C. Schank & R. Abelson, *Scripts, plans, goals and understanding* (Hillsdale, NJ: Erlbaum, 1977).
[21] O.M. Ovdey, G.Y. Proskudina, A review of ontology engineering tools (in Russian), *Russian Scientific Online Journal "Elektronnie biblioteki", 7*(4), 2004.
[22] M. Denny, *Ontology Tools Survey, Revisited*, 2004.
Available at
http://xml.com/pub/a/2004/07/14/onto.html. Accessed 01 Feb 2010.
[23] P. Krishnakumar, J. Raoul, Integrating the CLIPS Rule Engine with Protégé, *AIML 05 Conference, CICC*, Cairo, Egypt, 2005.
[24] M. Bakaev, T. Avdeenko, A formal research of older adults' physical and cognitive traits in movement and selection tasks for interface design. *In Proc. of IASDR 2009 Conference*, Seoul, Korea, 2009.
[25] M. Bakaev, K.H. Lee, H.I. Cheng, The aesthetic and emotional preferences of the elderly and the design factors for e-business web sites. *In The Eighth Pan-Pacific Conference on Occupational Ergonomics (PPCOE 2007)*, Bangkok, Thailand, 2007.