Specifications of Fuzzy Concepts with Evaluative Meaning in a Project Ontology during a Design of a System with Software

P Sosnin[1], E Sosnina[1] and A Kulikova[1]

[1] Ulyanovsk State Technical University, 32, Severny Venetc, Ulyanovsk, Russia

Abstract. The paper deals with ontological modeling the fuzzy concepts, uncontrolled use of which usually lead to problems with success in the development of systems with software. Typical concepts of this type are high-level requirements that are used in creating an architecture description of any system to be designed. In architecture practice, such requirements usually defined via constructive specifications that reflect viewpoints of stakeholders onto the system as a wholeness. Therefore, such requirements usually called as "concerns" play the roles of modalities which have a fuzzy content with estimation meaning that must be formed and controlled on the course of designing. The offered version of such work is implemented in the conceptual space, the kernel of which is a semantic memory of a question-answer type. Cells of such memory help to integrate symbolic, graphical and algorithmic specifications of concerns in corresponding articles in the project ontology. For scales of membership functions, it is suggested they reflections on levels that are oriented on models maturity.

1. Introduction

Over the past years, there were some attempts aimed at rethinking the basic concepts of software engineering, including the development of software intensive systems (Software Intensive Systems, SISs). Until now, the main reason for the innovative rethinking was and remains the problem of an extremely low degree of success [1] in designing the SISs. Almost all studies of this problem point to the very important role of negative manifestations of the human factor caused by lacks with understanding during personal and collaborative actions.

As one of the creations of Nature, understanding is the naturally artificial phenomenon that is responsible for the correct using of language means in human life. In cases, when it is necessary, the natural side of understanding manifests itself via managerial coordination of the perceptions (or imaginations) and corresponding descriptions. The artificial side corresponds to the use of means that help to model the internal processes and results of understanding beyond the brain for combining external modeling with internal processes for increasing the effectiveness of understanding. As it can be noted, a process of understanding and its result depend on a person who interacts with some case, applied means of modeling and a way of combining the internal and external processes aimed at understanding.

Designing the SISs is a source of heterogeneous and diverse cases in any of which it is necessary to constructively ensure the achievement of the necessary understanding. It should be implemented in the frame of the project language $L^P(t)$ that is created in the course of designing. Moreover, acts of understanding are implemented by designers in the current state of $L^P(t)$, thereby affirming that this state is adequate for the project to be developed. In this process, results of understanding had better register in the forms that are suitable for the reuse., because such forms will help to coordinate of understanding among stakeholders involved in the project.

Among very important forms registering of coordinated understanding, it needs to mark an architectural description of the designed system. Such description consists of a complex of architectural views any of which combines the certain block-and-line scheme with its description in the language $L^P(t)$. The quality of integrated architectural decisions essentially depends on the $L^P(t)$-kernel that is better to build as the project ontology $O^P(t)$ with notions without which reusable understanding is impossible.

The basic feature of any $O^P(t)$ is that its components must find objectivation in the corresponding project. This requirement extends to notions (concepts) directly or indirectly connected with the evaluation of the " project successfulness." Among these concepts, the important place occupies their set applied in the architectural description, first of all, a subset including "concerns" of stakeholders involved in the project. Usually, violations of requirement expressing the concerns lead to the unsuccessful results of designing. From this point of view, the greater part of concerns has both the semantic expressions and estimation values. It should be noted that any traditional concern has a name with fuzzy meaning, for example, extensibility, scalability, intelligibility.

Interests of this paper focus on ontological specifications of fuzzy concepts such as concerns from the side of their estimation meaning. Proposed decisions are oriented on their materialization in the instrumentally modeling environment OwnWIQA. This toolkit supports the designer activity at the conceptual stage, and it includes ontological maintenance of architectural decisions.

The remainder of the paper is as follows. The second Section presents the preliminary basis that includes some clarifications of considering the concerns in architectural descriptions and some features of creating and using the project ontologies in the OwnWIQA [2]. The third Section includes the review of the related works. The fourth Section focuses on our approach to the description of fuzzy concepts in the project ontology. Finally, the fifth Section has conclusions disclosing the features of the work with fuzzy concepts in designing the SISs.

2. Preliminary bases

2.1. Concerns in architectural descriptions and their specifications

As told above, on the course of developing the certain SIS, designers create the project language, the kernel of which contained the important concepts some of which can be qualified as fuzzy concepts that have the estimation value. This subset of concepts includes those that express the concerns used in architectural modeling the SIS to be designed. By the standard ISO/IEC/IEEE 42010:2011, concerns are defined as follows "interest in a system relevant to one or more of its stakeholders."

A system concern could pertain to any influence on a system in its environment including developmental, technological, business, operational, organizational, political, regulatory, or social influences. Generally, the typical kinds of concerns are presented in figure 1.

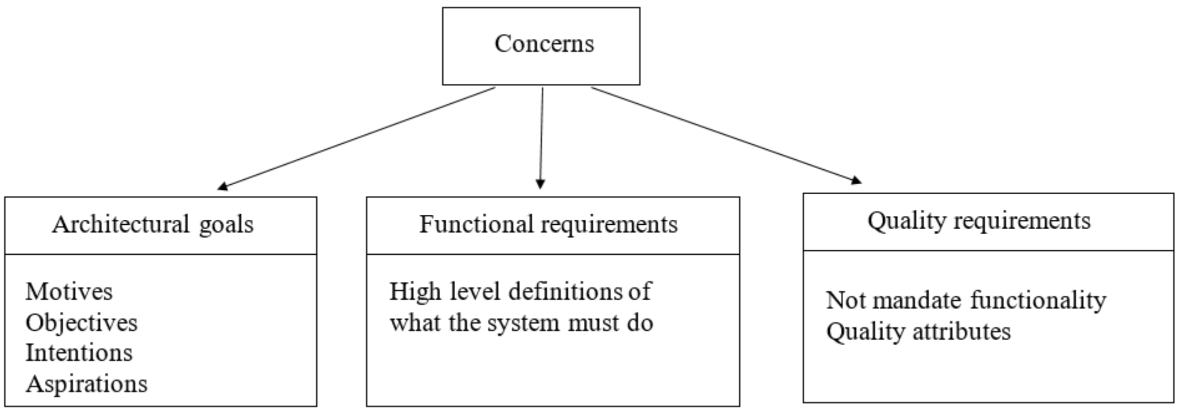

Figure 1. Kinds of concerns.

Understanding of stakeholders and their concerns is the basis of successful architectures because the diversity of stakeholders and their concerns create a rich environment and correspondingly lead to the complexities faced by architects who must consider them in architectural solutions. To underline similar conditions, let us present the list of concerns enumerated in [3]: acceptability, accessibility, accountability, accuracy, adaptability, administration, affordability, agility, assurance, auditability, authentication, autonomy, availability, backup, behavior, benefit, business alignment, business goals, business strategies, capacity, certification, communication, compatibility, completeness, complexity, compliance to regulation, conceptual integrity, concurrency, confidentiality, configurability, configuration management, consistency, continuity of operation, control, correctness, cost, credibility, customer experience, customizability, data accessibility, data integrity, data privacy, degradation, dependability, deployment, disaster recovery, disposability, distribution, documentation, durability, ease of learning, ease of use, economy of mechanism, effectiveness, efficiency, environmental protection, error handling, evolvability, extensibility, failure management, fault tolerance, feasibility, fidelity, flexibility, functionality, generality, implementability, information assurance, integrity, inter-process communication, interchangeability, interference, internationalization, interoperability, intuitiveness, known limitations, learnability, legal, licensing, localizability, logistics, maintainability, manageability, mobility, modifiability, modularity, monitoring, network topology, openness, operability, operating costs, optimizability, organization, performance, persistence, platform compatibility, portability, predictability, price, privacy, provability, quality of service, recoverability, regulatory compliance, reliability, repeatability, reporting, reproducibility, resilience, resource constraints, resource management, resource utilization, response time, responsiveness, reusability, robustness, safety, scalability, schedule, security, serviceability, simplicity, stability, state change, structure, subsystem integration, supportability, survivability, sustainability, system features, system properties, system purpose, technological constraints, testability, throughput, timeliness, traceability, trustworthiness, understandability, usability, usage, user-friendliness, vendor lock-in, versatility.

This list is open for the inclusion in it of other concerns (interests), but it can be taken as a basis for borrowing in any project (it is the main reason for including this list in our paper).

As told above, any unit of this list has the certain explication that helps to adopt it for the certain project and helps the architect to build the appropriate viewpoint in the diagrammatic form. All this information must find its expression in the corresponding article of the project ontology. We propose to extend this information to the additional specification of the concern, reflecting it from the side of the assessed value. In turn, this decision assumes the existence of a mechanism for assigning an estimated value for the concern, including the case when it has the type of a fuzzy concept. If such a mechanism is absent, it should be invented.

There are following approaches for the fuzzy specification of a concern that has an assessment value:

- The greater part of concerns corresponding to the quality requirement have formulae for computing the measures in the standard ISO/IEC/IEEE 52010:2011(former standard ISO 9126) and these measures can be used for defining the concerns as corresponding fuzzy variables.
- For some concerns, it is possible to use the standards that prescribe the normative of maturity (processes, people, project management and the others), for example, CMMI-1.3 Development, PMBOK 5 or P1M3.
- There are fuzzy concerns that have normative scales for expressing their estimation value, for example (informational) safety has the scale with seven evaluation assurance levels (standard ISO/IEC 15408).
- It needs to analyze the textual description of the chosen concern and to invent the specification of it as a corresponding fuzzy variable.

In any of the possible ways of specifying the concerns, it will need to build an appropriate membership function of the corresponding fuzzy variable. If for some concern this function is known then it can be used for operative monitoring the current estimative value of this concern on the course of designing. Thus at the end of designing it will be known which chosen concerns are achieved the planned value.

2.2. Features of project ontologies in the WIQA-environment
As told above, for the work with fuzzy concepts, we decided to use the toolkit OwnWIQA that includes the subsystem for creating the project ontologies and their combining in complexes, for example for a family of related projects (for a family of SISs). The conceptual model of such ontological structures is presented in figure 2.

The project ontology is created in the form of a *working dictionary* (corresponding with a certain project), which is divided into *groups* and *subgroups*. An ontology *concept (notion)* is a dictionary item that can be presented by its specification uploaded in the corresponding cell of the semantic memory of the question-answer type [4].

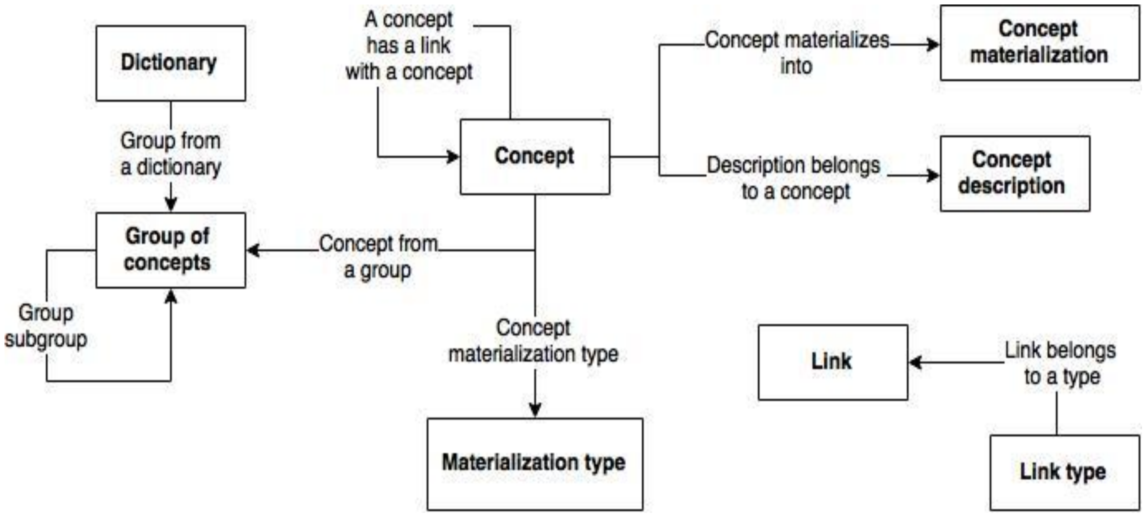

Figure 2. Framework specifying the creation of the project ontology.

Cells of such memory are adjusted on keeping a concept in a form, the typical structure of which is presented in figure 3 where the semantic meaning of any component is generally expressed by its name.

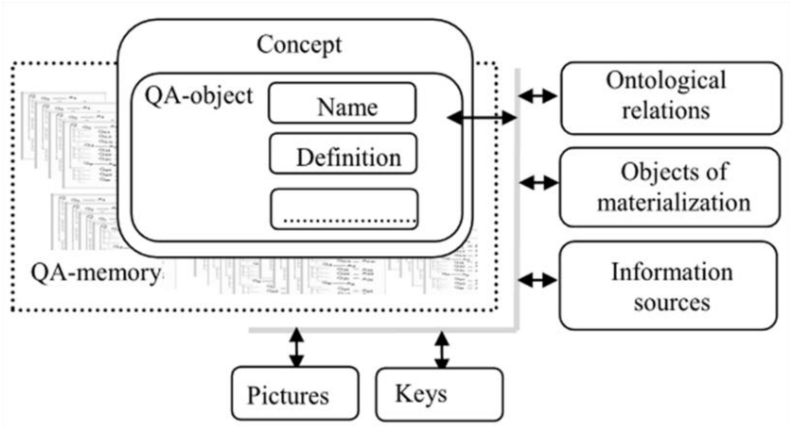

Figure 3. A typical structure for keeping a concept.

The scheme demonstrates that it has conditions for the attachment of a set of pictures (graphical units) that express the necessary ore useful information. Some of these pictures can be correspondent to the membership functions if the cell keeps fuzzy concepts, for example, a fuzzy concern in a graphical form. The toolkit OwnWIQA includes the graphical editor that can be used for creating and modifying similar forms.

3. Related Works

The first group of related works combines the papers focused on the place and role of concerns in the architectural description of systems with software. A deep analysis of the diversity of concerns and other novelties of the standard ISO/IEC/IEEE 42010:2011 is conducted in the paper [3]. Environmental aspects of architectural modeling are disclosed in the paper [5] where features of the context are focused on viewpoints that provide guidelines for framing and constructing architecture views. The study [6] deals with features of concerns and viewpoints in service-oriented applications. The real practice of architectural decisions is discussed in the industrial case study published in [7]. The current retrospection view on the theory and practice of architectural descriptions is presented in the paper [8]

Among papers that deal with the use of ontologies in designing the SISs, we mark ontologies of the applied type that usually is expanded using the other ontologies types. In accordance with the publication [9], the theory and practice of applied ontologies "will require many more experiences yet to be made." The specificity of project ontologies is indicated in some publications. In the technical report [10] the main attention is paid to "people, process and product" and collaborative understanding of interactions. As for developing the software systems and considering the ontological problems of software products, the use of ontology possibilities is investigated in the paper [11]. It needs to mark the paper [12], in which the project ontology is applied for the support of architectural recommendations

One more group of related papers focuses on the use of fuzzy concepts in architecture practice. In this group, first of all, we mark the paper [13] investigating the use of fuzzy measures in selecting the architecture tactics and the study [14] describing the fuzzy-based quantitative evaluation of architectures on the base architectural knowledge. In this group, we also mark the paper [15] that presents a fuzzy ontology-based approach to decision making in architectural design and the paper [16] describing the use of a maturity model in specifications of the architecture maintainability.

4. A way of ontological specifying the architectural concerns

4.1. The scale of estimation meanings

In creating the architecture of the certain system, any chosen concern should find the adequate specification and planned materialization. Therefore, it is expediently to bind with the considered concern its life-cycle. On the course of the life-cycle, the following states of the concern materialization are important:

1. The concern as one of the architectural requirements is adequately defined, and its specifications are coordinated with the planned estimation meaning.
2. The current state of the concern is coordinated with other concerns so that they are combined in an appropriate system of concerns.
3. For the concern, the corresponding view in the diagram form is drowned and registered. Such a view reflects the conditions, materialization of which will lead to objectifying the concern.
4. Conditions corresponding to the view are materialized with using the means of the corresponding viewpoint that is adjusted on achieving the planned meaning of the concern.

The integrated model of the concern states is prepared and registered for the reusable aims.

The indicated states are the controlled results of the process that is aimed at achieving the planned estimation meaning of the concern interpreted as the corresponding requirement. Moreover, such achievement is confirmed by the materialization of this requirement. Therefore, with the concern, it is expedient to bind a fuzzy attribute "an achieved state of the planned meaning of the concern" which opens the possibility for monitoring the state of the concern objectivation in the real-time of designing.

Such interpretation of estimating the concern objectivation is related with capability maturity models of processes in the software engineering. Typically, similar models are oriented on the scale with five levels of maturity. In the specialized applications of models, their levels have various interpretations. Among specialized models oriented on the architectural description, we mark the model described in [16]. This model has five levels of the maturity estimation – "drown," "modeled," 'documented, "traceable" and "reasoned." In the described way, for quality concerns, we also offer the use of five levels of estimation, but with interpretation described above in this subsection. Levels' version of the estimation simplifies defining the membership functions for concerns of the quality kind. It can also be applied for concerns of other kinds, but with an appropriate number of levels. Some templates forms for concerns not only the quality kind are presented in figure 4.

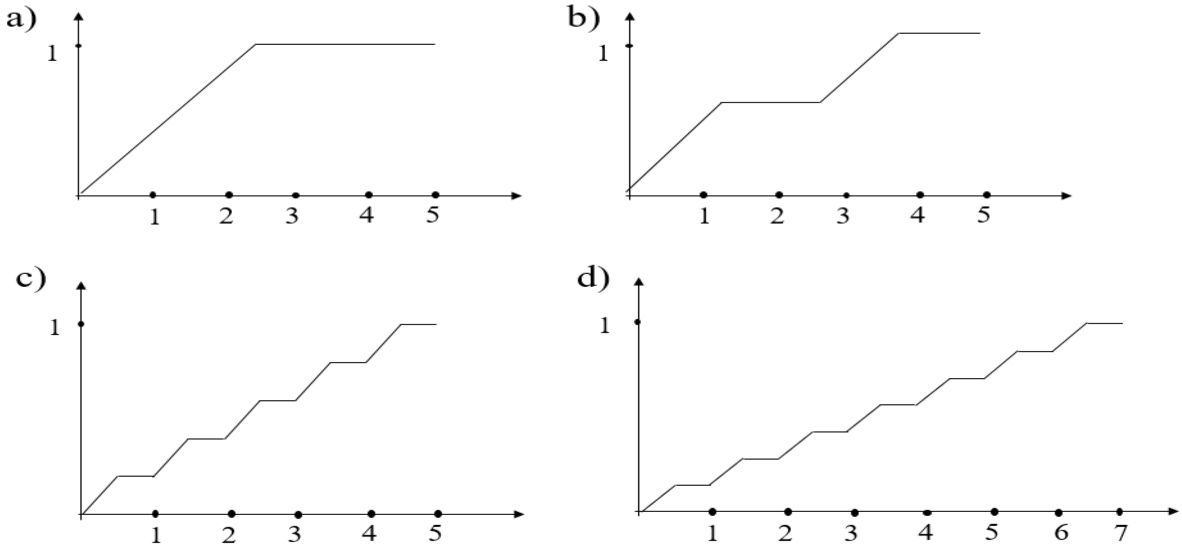

Figure 4. Template forms of the membership function.

Templates "a)" and "b)" are typical for concerns of the architectural goals. They are built and adjusted for the scale with five levels for corresponding with concerns of the quality kind (template "c)"). The fourth template "d)" is coordinated with seven evaluation assurance levels of the standard ISO/IEC 15408.

4.2. Some details of realization

To use and evaluate the concerns mentioned in the previous section while developing a certain project, we created a special section in the general ontology, i.e., the ontology which can be used as a basis for various projects.

Each concern usually undergoes expertise, which means that a stakeholder or an expert evaluates its meaning (degree) at various project phases. Such evaluation can be quite subjective and, therefore, the concern's value varies depending on the expert. Consequently, it is worthwhile to use fuzzy logic mechanisms to store information about the possible concerns which become linguistic variables and their values are defined as fuzzy.

Taking into account the abovementioned information, we decided to use the ontology structure of the WIQA environment and create the Concerns Section. Each ontology item has a name, a textual definition and several attributes such as:

- assignee (designer);
- date;
- view;
- references to the viewpoint;
- evaluative meaning, etc.

Figure 5 shows the way the Concerns Section is presented in the WIQA instrumental environment on the example of the "usability" concept.

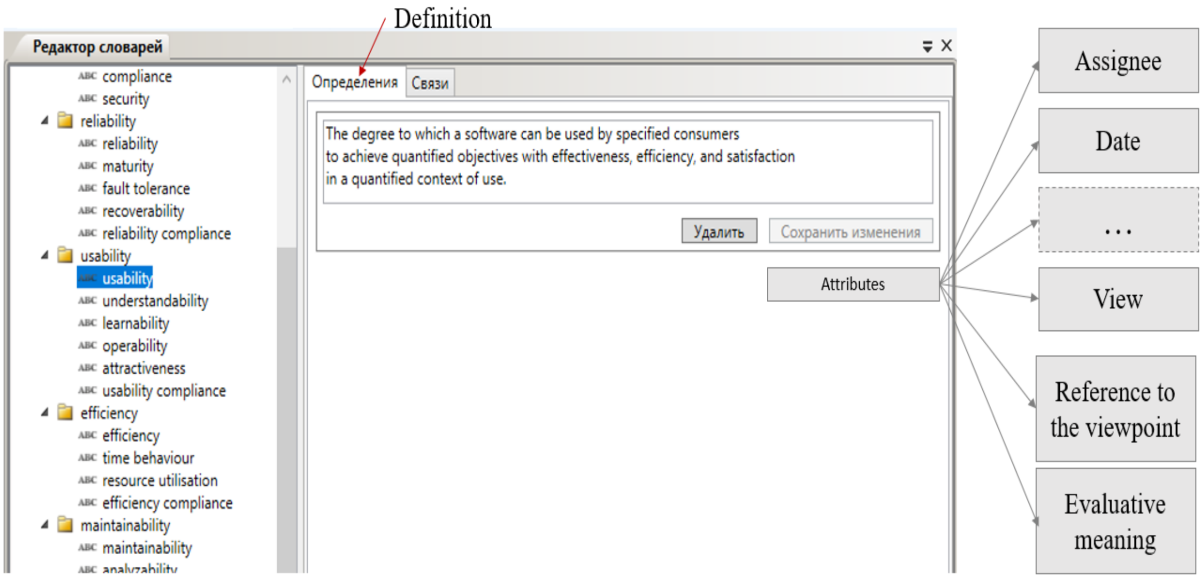

Figure 5. Concerns Section of the Project Ontology.

As it was indicated, except for the definition of this concept, its content is extended via basic and additional attributes that correspond to the semantic potential of the memory cell presented in figure 3. Let us clarify some of these attributes bound with architecture description:

1. The view corresponding to the concern is attached to it as a graphical file drawn with the use of the specialized graphical editor embedded to the OwnWIQA.
2. The reference to the viewpoint leads to the instruction that helps to materialize the view in the project. The instruction includes the structured list of the corresponding best practices, and this list is used in the process of monitoring the level of the concern objectivation.
3. Specification of each concern includes the appropriate membership function chosen from their collection, catalog of which is materialized as the specialized dictionary (figure 2). The membership function is registered as a graphical file attached to the concerned article.

It should be noted; the project ontology has the section for placing the ontology axioms, a set of which includes rules of consistencies of concerns. Providing the consistency is used in computing the estimative meanings of concerns named in rules.

4.3. Ontology search

Here is the algorithm used by the tool that finds ontology terms in the project documents and gets potential values out of these texts:

1) select a text T for analyzing;
2) split the text T into wordforms and get a set of wordforms {WF};
3) normalize the wordforms with the help of a morphological analyzer and get a set of normalized words $\{(WF, WF^N)\}$;
4) M = 0;
5) select the normalized wordform with the index M – WF^N_M;
6) IF WF^N_M is present in the concerns section of the project ontology (O^{PC}) THEN BEGIN;
7) add the words related to WF^N_M (WF_{M-3}, WF_{M-2}, WF_{M-1}, WF_{M+1}, WF_{M+2}, WF_{M+3}) to the evaluation attribute of the corresponding concept;
8) END;
9) IF there are more wordforms in the set $\{WF^N\}$ THEN BEGIN
10) M = M+1;
11) GOTO 5;
12) ELSE END.

The attributes added by the tool to the concepts of the project ontology can be considered as draft values of the fuzzy variables. That means that a designer should review these attributes and evaluate the concerns. This should be repeated at each design phase.

4.4. An example use of the proposed method

Let us consider evaluation *understandability* and *usability* of the user interface. In this case, these two concerns are deeply related to each other because when we talk about good *usability*, we mean that a user can easily *understand* how to use a software system when he or she looks at its interface. And if something is unclear, a user can take advantage of the project ontology to find out more information about the system.

Consequently, understandability can depend on whether there is enough information about the user interface in the project ontology or not. That means that when creating user interface prototypes, a designer should check if all its elements are represented in the project ontology and add new concepts to it when necessary. The required value of the understandability is considered as achieved when all the user interface elements can be found in the project ontology, i.e., can be easily understood by a user.

The procedure which verifies if a certain term is present in the project ontology and adds a concept if it is absent can be programmed in pseudocode language in the WIQA instrumental environment itself (as shown below). For convenience, we created a text file: "interface_elements.txt" which stores all the textual info from the user interface prototype.

```
    …
&i& := 0
    LABEL &LCycle& // loop index
   &conceptsNumber& := CountStringsInFile("C:\\WIQA\\interface_elements.txt")
   IF &i& >= &conceptsNumber& THEN goto &LOut&
       &word& := ReadFromFile(&i&, "C:\\WIQA\\interface_elements.txt")
            // read the string with the index i from the file with the normalized wordforms
       &wordID& := Onto_GetWordId(&dictID&, &groupID&, &word&)
       IF &wordID& != 0 THEN BEGIN
          // if the ID of the word is not equal to 0, that means that such word exists in the ontology
      &newWordID& := Onto_CreateWord (&groupID&, &word&)
      END
      &i& := &i& +1
      GOTO &LCycle&
       LABEL &LOut&
```

5. Conclusion

In developing the SISs, it is typical to use the fuzzy concepts, specifications of which are formed and coordinated on the course of the design process. This paper focuses on one kind of architectural requirements (concerns) that express the interests of active and potential stakeholders. The choice was caused by the essential influence of objectifying the concerns on the success of projects.

The offered way of constructive work with concerns is adjusted on its implementation in the instrumental environment OwnWIQA that has the subsystem for creating and using the applied ontologies. The basic feature of such ontologies is the place of any concepts in the semantic memory of the question-answer type. Cells of such memory allow not only register symbolic descriptions of concerns, but also combine with it other useful informational units, including textual, graphical and algorithmic constructs. In other words, memory cells are opened for uploading specifications of such complicated constructs as concerns.

The complexity of concerns is caused by the necessity as to declare them so to objectify in substantiated forms that are prepared for the reuse with different aims. The main of these aims is the confirmation that concerns found them objectivations. In the offered way, ontology specifications of each processed concern help to organize monitoring the process of achieving its planned meaning. For computing the current meaning, we offer using the membership functions, grading of which corresponds to the levels of professional maturity of the processes in the development of SISs. Such choice focuses on best practices that provide achieving the planned version of objectifying the concerns. Moreover. It helps to register the history of achieving the correspondent requirements, states of which are used in indicated monitoring.

6. References

[1] Chaos reports 1994-2018. Available at http://www. standishgroup.com

[2] Sosnin P 2018 Experience-Based Human-Computer Interactions: Emerging Research and Opportunities (IGI-Global) 296 p

[3] Hilliard R 2015 Lessons from the Unity of Architecting (Software Engineering in the Systems, Context) Chapter 10 pp 225–250

[4] Sosnin P and Pushkareva A 2017 Ontological Controlling the Lexical Items in Conceptual Solution of Project Tasks (LNCS, vol 10409) pp 31-46

[5] Bedjeti A Lago P Lewis G De Boer R D and Hilliard R 2017 Modeling Context with an Architecture Viewpoint (Proceedings of IEEE International Conference on Software Architecture)

[6] Gu Q and Lago P 2009 On service-oriented architectural concerns and viewpoints (*2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*, Cambridge) pp 289-292

[7] Dasanayake S Markkula J Aaramaa S and Oivo M 2015 Software Architecture Decision-Making Practices and Challenges: An Industrial Case Study (Proceedings od 24th Australasian Software Engineering Conference) pp 88-97

[8] Hasselbring W 2018 Software Architecture: Past, Present, Future. In: Gruhn V., Striemer R. (The Essence of Software Engineering. Springer) pp 168-184

[9] Garcia A C B Kunz J Ekstrom M and Kiviniemi 2003 A Building a Project Ontology with Extreme Collaboration and Virtual Design & Construction (CIFE Technical Report # 152, Stanford University)

[10] Fitsilis P Gerogiannis V Anthopoulos L 2014 Ontologies for Software Project Management: A Review, Journal of Software Engineering and Applications, vol 713 pp 1096-1110

[11] Eden A H and Turner R 2007 Problems in the Ontology of Computer Programs (Applied Ontology, vol 2 1, Amsterdam, IOS Press) pp. 13–36

[12] Bhat M Shumaiev K Biesdorf A Hohenstein U Hassel M and Matthes F 2017 An Ontology-based Approach for Software Architecture Recommendations 23rd Americas Conference on Information Systems (AMCIS 2017) vol 4) pp 3110-3120

[13] Alashqar A M El-Bakry H M and Abo Elfetouh A 2015 A Framework for Selecting Architectural Tactics Using Fuzzy Measures (International Journal of Software Engineering and Knowledge Engineering vol 27 3) pp 475–498

[14] Dhaya C and Zayaraz G 2012 Fuzzy based Quantitative Evaluation of Architectures using Architectural Knowledge (*Int Journal of Advance Science and Technology*) pp.137-154

[15] Di Noia T Mongiello M Nocera F and Straccia U 2018 A fuzzy ontology-based approach for tool-supported decision making in architectural design (Knowledge and Information Systems) pp 1–30

[16] Rathfelder C and Groenda H 2008 Towards an Architecture Maintainability Maturity Model (Softwaretechnik-Trends vol 28 4) pp 3-7

Acknowledgments

This work was supported by the Russian Fund for Basic Research (RFBR), Grant #18- 07-00989а, 18-47-730016 р_а, 18-47-732012 р_мк and the State Contract №2.1534.2017/4.6.