Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)

Volume 1

IDAACS'2013

The crossing point of Intelligent Data Acquisition & Advanced Computing Systems and East & West Scientists

September 12-14, 2013
Berlin, Germany

ORGANIZED BY

IEEE Ukraine Section I&M / CI Joint Societies Chapter

Research Institute for Intelligent Computer Systems Ternopil National Economic University
V.M. Glushkov Institute of Cybernetics National Academy for Sciences of Ukraine

University of Applied Sciences Hochschule für Technik und Wirtschaft (HTW) Berlin, Germany

TSB Technology Foundation Berlin

SPONSORED BY

IEEE Ukraine Section I&M / CI Joint Societies Chapter

Hochschule für Technik und Wirtschaft Berlin University of Applied Sciences

IEEE instrumentation & measurement society

The University of Maine

IEEE R8

River Publishers

IEEE Ukraine Section

With a population of over 3.5 million, Berlin is Germany's largest city and home to the federal government and parliament. In no other German city is there more progress and growth since German Reunification evident than here.

Berlin is not an industrial city, but a science, culture and media mecca with a distinct service sector in the German market. A vibrant multi-cultural orientation is characteristic of Berlin and makes it particularly attractive.

Berlin is the largest university city in Germany with three universities, four art academies, seven universities of applied sciences and a number of research institutions. The scope of academic discourse in Berlin is vast.

Additional copies may be ordered from:

IEEE Operations Center
445 Hoes Lane, P.O. Box 1331,
Piscataway, NJ 08855-1331 USA

IDAACS'2013 Organizing Committee
Research Institute for Intelligent Computer Systems
Ternopil National Economic University
3 Peremoga Square
46009 Ternopil, Ukraine
E-mail: orgcom@idaacs.net

IEEE Catalog Number: CFP13803-PRT

ISBN: 978-1-4799-1426-5

Table of Contents

VOLUME 1

VOLUME 2

The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
12-14 September 2013, Berlin, Germany

Decision Support System of Software Architect

Alexandr Harchenko [1], Ihor Bodnarchuk [2], Iryna Halay [1]
[1] National Aviation University, Kosmonavta Komarova ave. 1, Kyiv, 03058, Ukraine,
e-mail: harchenko_a1@voliacable.com
[2] Ternopil Ivan Pul'uj National Technical University, Ruska str., 56, Ternopil, 46000, Ukraine,
e-mail: bodnarchuk.io@gmail.com

Abstract—**Represented system is for supplying of architect with data, knowledge and methods needed to make decisions in the process of software systems (SS) design. It includes two subsystems: first one is subsystem of requirements specification and communication to SS, and second is architecture design subsystem. For requirements specification and communication formalism of basic protocols method and technic of expert pairwise comparisons are used. The selection of architecture decisions is represented as model of multi criteria hierarchic optimization, where modified Analytic Hierarchic Process (AHP) was applied. Main units of the system is implemented as formalized models or software units.**

Keywords—decision support system; software system architecture; optimization; requirements to the software system.

I. INTRODUCTION

Development of modern software products requires from designers to take to account significant amount of data, knowledge, factors to make effective decisions. Software systems are highly intellectual products, what complicates the process formalization of their design, and this, in own turn, does not allow to develop and use automation tools for these processes. Thus applying of formal methods like mathematical modelling, optimization, decision support theory to producing of models of design processes and creation on their basic the automation tools for decision support is actual at present.

Especially it is important on the stage of requirements specification and architecture design because these stages are the first and correction of wrong decisions made here leads to great losses. Nowadays automation systems have being using for design of requirements but it concerns to functional requirements only [1]. Talking about quality requirements some obstacles appear both during design process formalization and during requirements communication.

In [2] to solve these problems the SQFD method (Software Quality Function Deployment) was used. In the proposed system these problems were solved with applying of formalism of basic protocols method [3] and expert method of pairwise comparisons matrices.

Two standard approaches for design of software architecture are used most widely. First one consist in characteristics evaluation of selected sample of architecture with simulation method [4].

In the methods of other approach architecture characteristics are evaluated on the basic of expert questioning results. For example, in the ATAM method (Architecture Trade-Off Analysis Method) the risks that architecture does not meet conceptual requirements are evaluated [5]. The conceptual requirement of one stakeholder in the ATAM method is described with scenarios. Then the alternatives are analyzed on subject of supporting each of the scenarios. The disadvantages of both approaches are the limited number of aspects of software system use, that can be simulated together, and not sufficient formalization and high costs.

In represented system quality architecture evaluation among the set of alternatives is performed as solving of multi criteria hierarchic optimization problem [6]. Alternative architectures are compound with patterns that are stored in the repository as frames.

To solve the problem of optimization we used the Saaty's Analytic Hierarchic Process where for calculation of weight indices for alternatives the modified algorithm is used. It allowed to choose the architecture among bigger quantity of alternatives ($n > 9$), than in classical method, and to obtain better solutions.

II. SYSTEM STRUCTURE AND FUNCTIONALITY

The general look of the system is represented in the Fig. 1.

The system consists of data warehouse (DWH), where architectural patterns are stored, and program units which provide data input and their logical processing (OLTP), and units that perform informative processing – Data Mining.

This system implements a sequence of steps that allow you to get a decision on the choice of software system architecture that meets functional requirements and is optimal relatively to quality criteria.

The first step is elaboration of requirements to the system both functional and quality. The communication of requirements to the software system on requirements to the system's architecture is executed too. Elaborated requirements are specified in terms of the standard ISO\IEC 25010.

Alternate architectures that meet functional requirements are formed on the second step. These architectures are composed of standard patterns from the repository and according to the functional requirements and peculiarity of software domain. It corresponds with the technology of Microsoft ™ Corp.

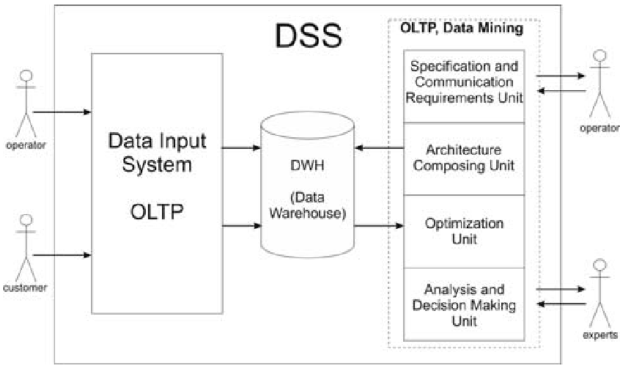

Figure 1. Structure of the system.

On the third step the comparative evaluation of architectures is performed relatively to the quality criteria of architecture. Here the method of multicriteria hierarchical optimization is used, and expert method of pairwise comparisons. The modified algorithm for calculation of weight indices for alternatives is applied that allows to expand the range of methods utilization for case of great quantity of alternatives ($n > 9$). Ranged according to the relative values of quality criteria architectures are passed to the analysis and decision making on selection of the best variant.

On the next, forth, step participating groups of experts analyze ranged regarding to the quality criteria alternatives, and trade-off decision is made for conflicts. Alternatives are ranged as well for the totality of quality criteria for the architecture and for the quality criteria of software system too. On the base of these results the final decision is made.

A. Requrements specification and communication unit.

This unit is a modification of software complex represented in [2]. But unlike of that one here for formalized representation of requirements in addition to the method of patterns of standard ISO 25020 also the formalism of basic protocols method is used [3]. It allowed to reduce the quantity of errors in the specification of requirements since formalization gives possibility to find out about 30% of all errors [3]. For the communication of quality requirements for the software system onto the quality requirements for the architecture, models of basic protocols and expert method of pairwise comparisons are used. Specification of requirements is an aggregate of properties that behavior of the software system must satisfy. These properties can be represented using the formalism of basic protocols.

Basic protocol is an expression like $\forall x(\alpha\langle u\rangle\beta)$, where x is a list of typed parameters, α and β – formulas of basic language, u – a process of the protocol. Formula α is a precondition and formula β is a postcondition of basic protocol. Postcondition express the state of the system after the execution of basic protocol. The basic protocol is treated as a formula of temporal logic that express the fact when (for certain values of parameters) the state of the system has a markup what satisfy the condition α, so the process u transfers the system to the state what meets the demands of condition β. Basic protocols formalize requirements to the software system. To write down α and β language closed to the language of specification is used. For definition of processes the language of algebra of processes is used like CCS, CSP or languages like UML, MCS or SQL. In [3] this formalism was used successfully for requirements verification to telecommunication systems.

In our case precondition α are requirements of quality of software system, specified in terms of standard ISO/IEC 25010 and postcondition β is requirements of quality of software system architecture. The process u is a process of communication (transformation) of α into β that is realized by means of expert technic of pairwise comparisons. Its sense consists in composing of matrix $B^s\left(b_{ij}^s\right)$, where b_{ij}^s expresses how the influence of i^{th} criteria of architecture quality exceeds the influence of j^{th} criteria on the realization of s^{th} criteria of software product quality. From built matrix the weight indices $w_i^s\left(i=\overline{1,n}\right)$ of architecture quality are calculated. Assigning priorities for quality criteria of software system $P_s\left(s=\overline{1,m}\right)$ we calculate integral weight indices

$$\overline{w_i} = \sum_{s=1}^{m} w_i^s \cdot P_s , \qquad (1)$$

and with constraint $\overline{w_i} \le \overline{w_t}$ we obtain criteria of architecture quality that are included in β.

Using of basic protocols formalism allowed to unify the notation of α and β what gave possibility to decrease quantity of errors in specifications and to formalize the procedure of requirements communication.

B. Architectures optimisation unit.

The structure chart of selection of architecture decision with accounting of indices of quality is depicted on the Fig. 2.

$K_i^1, i=\overline{1,m1}$ – criteria of quality of software system according to the standard ISO/IEC 25010;

$K_i^2, i = \overline{1, m2}$ – criteria of architecture quality;

$A_i, i = \overline{1, n}$ –alternative architecture decisions.

The list of quality in use criteria is defined and specified by software system developer together with customer in the unit mentioned above. Architecture quality criteria $\{K_i^2\}$ can be defined by means of communication of criteria $\{K_i^1\}$ in the mentioned above unit with accounting of recommendations of standard ISO/IEC 25010.

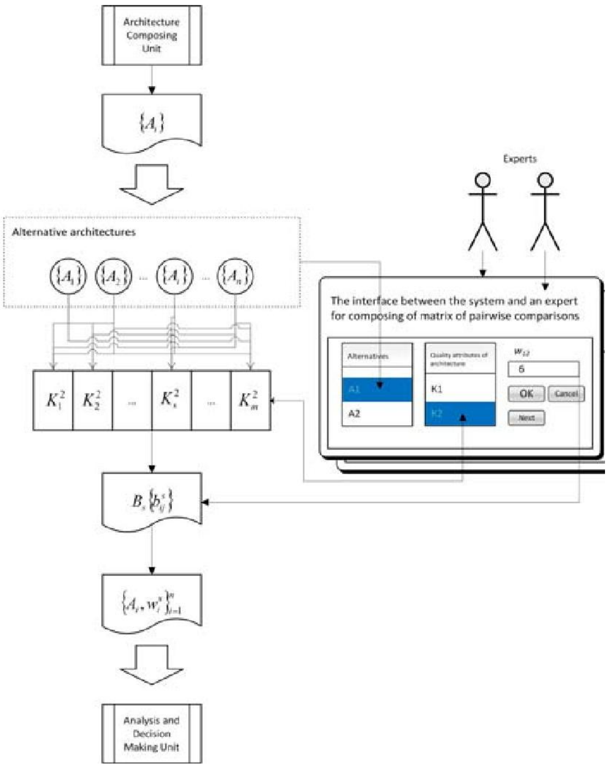

Figure 2. Hierarchical representation of the problem of architecture selection.

The architecture decision is chosen from the condition of optimization for the aggregate of criteria $\{K_i^1\}$, $\{K_i^2\}$. This is the problem of multicriteria hierarchical optimization and for the solution of such problems the Saaty's analytical hierarchic process (AHP) is used most often [7].

When AHP is used to solve such problems the weight indices w_i of alternatives (criteria) on each level are calculated by means of matrices of pairwise comparisons $B(b_{ij}^s)$ which are filled up by experts (here b_{ij} defines the supremacy of i^{th} criteria over j^{th}).

Indices of matrix must be consistent i.e. $b_{ij} = w_i / w_i \quad \forall b_{ij} \in B$. Weight indices in this case are calculated as eigenvector of matrix of pairwise comparisons which correspond to maximum characteristic number of matrix. But for considerable number of alternatives ($n > 9$) since influence on the experts of different factors the matrix $B(b_{ij})$ is inconsistent and its rank will differ from one i.e. the matrix will have some eigenvalues. So why in [6] for solution of problems of optimal architecture choosing for big quantity of alternatives ($n > 9$) the algorithm for calculation of alternatives' weights w_i in AHP on the base of condition of inconsistency minimization for matrix $B(b_{ij})$.

For possibility to choose the level of consistency when calculating we introduce the threshold value of consistency δ_t and measure of consistency we will write down as follows

$$\left| \frac{w_i}{w_j} - b_{ij} \right| \le \delta_t \cdot b_{ij}, \quad \delta_t \ge 0, \quad (2)$$

where δ_t is given threshold value.

Then weight indices w_i which minimize (2) we can find as solution of following problem

$$\min_{\{w_i\}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(w_i - b_{ij} w_j\right)^2 \quad (3)$$

$$a_i \le w_i \le c_i, \quad i = \overline{1, n}$$

$$-\delta_t \cdot b_{ij} \cdot w_j \le w_i - b_{ij} \cdot w_j \le \delta_t \cdot b_{ij} \cdot w_j; i, j = \overline{1, n}. \quad (4)$$

The problem (3), (4) is nonlinear programming problem and through the change of variables it can be brought to the equivalent linear programming problem:

$$\min_{\{w_i\}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(y_{ij}^+ - y_{ij}^-\right)$$

$$w_i \ge a, i = \overline{1, n}, \quad (5)$$

$$w_i - b_{ij} w_j = y_{ij}^+ - y_{ij}^-.$$

$$-\delta_t \cdot b_{ij} \cdot w_j \le w_i - b_{ij} \cdot w_j \le \delta_t \cdot b_{ij} \cdot w_j,$$

$$y_{ij}^+, y_{ij}^- \ge 0; i, j = \overline{1, n}. \quad (6)$$

The equivalence of problems (3), (4) and (5), (6) follows from the fact that vector of constraints at y_{ij}^+ and y_{ij}^- are linearly dependent and therefore any solution belongs to y_{ij}^+ or in y_{ij}^- and therefore the minimum (5) corresponds to the minimum (3). For each criterion the matrix $B\left(b_{ij}^{s}\right)$ is composed, where b_{ij}^{s} shows how i^{th} alternative exceeds j^{th} in the realization of s^{th} criterion. For obtained matrices the sets of weight indices $\left\{w_i^s\right\}, i=\overline{1,s}, s=\overline{1,7}$ are defined as solution of the problem of linear programming. For ranging of alternatives on the set of criteria it is needed to find their priorities. To define the priorities of quality indices of architecture $\left\{K_i^2\right\}$ for their influence on the realization of quality indices of software system $\left\{K_i^1\right\}$ experts fill up the matrix of pairwise comparison $B^s\left(b_{ij}^s\right)$, where the value b_{ij}^s defines how the influence of criterion K_i^2 exceed the influence of criterion K_j^2 on the realization of quality criterion K_s^1 of software system. After solving of the problem (5), (6) we will obtain the sets of priorities of criteria of architecture quality $\left\{P_i^{1s}\right\}$, $i=\overline{1,m2}$, $s=\overline{1,m1}$. Then the weight of alternate architecture A_i regarding the realization of criterion of software system quality K_s^1 will be calculated according to the formula:

$$J_i^{1s}=\sum_{j=1}^{m2} p_j^{1s}\cdot w_i^j,\quad i=\overline{1,n},\quad s=\overline{1,m1}, \tag{7}$$

w_i^j are weight indices defined on the previous stage.

Now we can range the alternatives $\{A_i\}$ according to the value $\left\{J_i^s\right\}$ for each $s=\overline{1,m1}$. That is we will obtain the sets of alternatives $\left\{A_i^s, P_i^{1s}\right\}, s=\overline{1,m1}$.

C. Analysis and decision making unit.

To make final decision for the selection of the alternative architecture the analysis of results outgoing from the optimization unit is performed.

Since some groups of specialists take part in the process of software system development with different priorities for each quality attribute so the priorities for every group are defined by forming of matrices of pairwise comparisons, and using AHP the priorities of criteria $\left\{P_i^s\right\}, i=\overline{1,k2}$ are found, where s is a number of group of experts. Compromise decision is made as geometric mean $P_{ij}^*=\sqrt[n]{P_{ij}^1\cdot P_{ij}^2\cdot\ldots\cdot P_{ij}^n}$ or as averaged with taking to account the index of competency for experts groups $P_{ij}^*=P_{ij}^{\alpha_1}\cdot P_{ij}^{\alpha_2}\cdot\ldots\cdot P_{ij}^{\alpha_n}$, ($\alpha_1,\alpha_2,\ldots,\alpha_n$ are indices of competency).

To achieve compromise during making the final decision and as well when the requirements to the software system is changed during its development the priorities of quality attributes may be changed. In this case for correcting of criteria priorities the expression is used

$$D'_{s,i,j}=\frac{\left|J_i-J_j\right|}{\left|w_i^s-w_j^s\right|}\cdot\frac{100}{P_s}. \tag{8}$$

Here $D'_{s,i,j}\left(s=\overline{1,m2}; i,j=\overline{1,n}; i\neq j\right)$ is minimal change of the priority P_k for the criterion of quality K_s that changes the order of adjacent architecture alternatives A_i and A_j on reverse. The minimal value of $D'_{s,i,j}$ shows that the priority P_s of the attribute K_s is critical for the changes of opinions in pairwise comparisons.

Using Eq. (8) for each criterion K_s we are available to find the interval ΔP_s in which experts can correct priorities P_s directly or by changing of values of pairwise comparisons without changing of order for the ranged alternatives $D_s^*=\min_{i,j} D_{s,i,j}=\Delta P_s, i=\overline{1,n}, s=\overline{1,k_2}$.

If it is needed to range the alternatives regarding to the global quality of software system we have to find the priorities of indices of software system quality $\left\{P_i^2\right\}$, $i=\overline{1,m1}$, applying the modified procedure of AHP.

We can find the weights of alternatives regarding to the realization of global criterion of software system quality for the value of index:

$$J_i^0=\sum_{s=1}^{m1} J_i^{1s}\cdot P_s^2,\quad i=\overline{1,n}. \tag{9}$$

Then the index of importance of the alternative A_i for the set of criteria will be

$$J_i=\sum_{j=1}^{m2} P_j\cdot w_j^i,\quad i=\overline{1,n}, \tag{10}$$

and ranging of $\{A_i\}$ is performed for values $\{J_i\}$.

Obtained alternatives, ranged regarding both to separate criteria and to their aggregate on the level of architecture quality and software system quality, are used for the final decision making for choosing of architecture.

III. CONCLUSIONS

Applying of proposed system for software design will allow due to using of formalized models, repository of ready-made architecture patterns, methods of optimization to increase the quality of software products, to decrease time of development and decrease costs.

Applying of the method of formalization of quality requirements will allow to decrease the quantity of wrong specifications approximately on 30% [3]. The method of communication of requirements to quality will allow reasonably to choose the criteria of architecture quality, and the algorithm created on this base allows to automate this procedure.

Applying of AHP gives possibility to choose the best architecture among the set of alternatives for the aggregate of quality criteria unlike to well-known methods ATAM, CBAM and others, where architecture is evaluated for one criterion.

Using of proposed algorithm for calculation of weight indices for alternatives in AHP on the base of minimum of inconsistency of pairwise comparisons matrix expands the limits of method application for bigger quantity of alternatives ($n > 9$), i.e. the selection of decision is made on total set of alternatives. And when using standard AHP the set of alternatives is divided on clusters with $n < 9$, and the problem is solved for each cluster separately, and then the best variants from each cluster are compared. So this way considerably increases time for obtaining of the solution. Thus, for the example discussed in [8] when n=15 the set of alternatives ought to divide on two clusters, and then the time to find the best alternative for standard algorithm will exceed more than twice the time for modified algorithm.

Applying of expression (8) when requirements are changed will allow to avoid repeated ranging of alternatives if values of changes lay in frames of defined intervals.

The method of composing of alternate architectures and using of the repository of patterns gives possibility to use effectively ready-made architectural patterns in the selection of optimal decision for wide range of applications.

REFERENCES

[1] Elizabeth Hull, Ken Jackson, Jeremy Dick, *Requirements Engineering*, Second edition, Springer Science+Business Media, 2005, 198 p.

[2] A. Harchenko, V. Yatcyshyn, I. Raichev, "A tool of quality requirements development and communication for software systems," in *Software Engineering*, vol. 2, 2010, pp. 29-34. (in Ukrainian)

[3] A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov, "Basic protocols, message sequence charts, and the verification of requerements specifications," *ISSRE 2004*, WITUL, Rennes, 4 November 2005, pp. 112–142.

[4] P. Clements, R. Kazman, M. Klein, Evaluating Software Architectures: Methods and Case Studies, Boston: Addison Wesley, 2002, 323 p.

[5] Len Bass, Paul Clements, Rick Kazman, *Software Architecture in Practice*, 2nd ed. Boston, MA: Addison-Wesley Professional, 2003, 528 p.

[6] A. Harchenko, I. Bodnarchuk, I. Halay, V. Yatcyshyn, "Software architecture design on the basic of method of hierarchic optimization," *Proceeding of the VIII International conference on MEMS Design*, Lviv – Polyana, 2012, pp. 39–40.

[7] T. Saaty, L. Vargas, *Decision Making with the Analytic Network Process*, N.Y.: Springer, 2006, 278 p.

[8] A. Harchenko, I. Bodnarchuk, I. Halay, "Stability of the solutions of the optimization problem of software systems architecture," *Proceedings of the VIIth International Scientific and Technical Conference CSIT 2012*, Lviv 2012, pp. 47-49.