Multi-criteria choice of the software package architecture for automating the analysis of the forest vegetation state

S A Potryasaev*, I Y Pimanov and A E Semenov

Laboratory of Information Technologies in System Analysis and Modeling, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 14th Line of Vasilievsky island 39, St. Petersburg 199178, Russian Federation

*Corresponding e-mail: spotryasaev@gmail.com

Abstract. The article describes a method for solving the problem of choosing the architecture of a software package (SP) of an information system for analyzing the state of forest vegetation, carried out using distributed information resources and calculation modules which usually belong to various organizations. The SP quality indicators can be assessed using both quantitative and qualitative scales. The proposed method is based on the joint application of the mathematical apparatus of the fuzzy sets and experimental design theories. It is shown that the predominant type of architecture for this class of tasks is the service-oriented architecture, which provides the best value of the integral indicator of the SP quality.

1. Introduction

The process of finding solutions to the overwhelming majority of environmental and forest management problems, the state of vegetation assessments, and many others, are connected to the development and application of information technology (IT). One of the main advantages arising from the use of IT is the ability to automate labor-intensive procedures for obtaining and processing large volumes of initial data required for analysis, including spatial data, as well as interpreting the results obtained and providing them to users in a simple visual form. The main trend in the development of IT is the use of unified data processing units and their combination, including the concept of "zero coding" on platforms capable of serving a large number of users. However, Environmental Science has somewhat different requirements to software; the latter is mostly used to build and run complex mathematical models that require data from heterogeneous distributed sources. A typical example is the InnoForestView project [1], which analyzes the negative consequences of the influence of anthropogenic and natural factors on the forests of border regions. To achieve the project goals, mathematical models and algorithms for processing heterogeneous data and information about the state of forest vegetation in the affected areas are being developed. Therefore, it became necessary to combine the results of calculations performed according to various algorithms and in various organizations involved in the implementation of the project. Under these conditions, when creating a unified distributed information system, the task of a reasonable choice of architecture and technology of the developed software package (SP) comes to the fore. To perform the task, it is required to carry out multi-criteria assessment, analysis and selection of the basic architecture of the information platform, on which special model-algorithmic, software and information support will be implemented.
Among the most common approaches to architectures of programs and software systems [2] suitable for solving the formulated problems, the following architectures can be singled out:

− monolithic architecture;
− modular architecture;
− component architecture;
− client-server architecture;
− service oriented architecture.

Applications with monolithic architecture [3, 4] are easy to implement, manage and deploy. They provide high consistency of a program code, explicit control of the computational process, and unified error control. This type of architecture is advisable for small information systems localized on one computing resource. Basically, these are ecological models used by a limited number of researchers to perform calculations based on data prepared in advance.

For more complex software systems, modular architecture is used [5]. Its underlying application decomposition simplifies the process of developing, maintaining, and then upgrading the software. The development of modular architecture has led to the emergence of the concept of "plug-in", i.e. an independently compiled software module, dynamically connected to the main program and designed to expand its functionality. In solving environmental problems, plugins are often used to pre-process some data types.

A similar approach is used in component architecture [6], which allows you to build more complex information systems based on the use of debugged "building blocks" of program code. The main idea behind the component approach is code reuse. This architecture places greater demands on plug-in versatility. The development of component architecture resulted in the emergence of software libraries with a wide range of universal functions. Component architecture, like monolithic architecture, is intended for building localized modeling software systems.

The foundations of network interaction of software modules were laid in client-server architecture [7]. In a narrow sense, the components of this architecture are considered to be a user client that implements the main logic of the task, and a server part, represented by a DBMS or a file server. In a broad sense, client-server architecture describes any interaction of two or more distributed software modules of the information systems being created. However, this interaction is most fully embodied in the ideology of service-oriented architecture.

Service Oriented Architecture (SOA) implements a modular approach to software development based on the use of distributed, loosely coupled replaceable components equipped with standardized interfaces for interoperability using standardized protocols [8]. This approach is used to build complex distributed information systems based on the integration of web services. In this case, integration is carried out at the protocol level, without the interacting parties’ understanding of each other’s internal structure. This provides the so-called weak "mesh" of modules (web services). According to ISO / IEC / IEEE 24765, the term “coupling” describes the degree of interdependence between software modules, the strength of the relationships between modules, and the degree to which different subroutines or modules are interdependent. The use of services allows you to combine heterogeneous computing modules into a single complex application that operates on a variety of hardware (servers). It is this situation that is typical for information systems created during the implementation of interdisciplinary projects, including international projects involving assessments of the state of forest vegetation on the basis of heterogeneous data and carried out in several organizations separated in geographic space.

Service-oriented architecture can be presented by both the "classic" form and micro-services format. At the same time, the "classic" service-oriented approach has already taken root in a certain segment of scientific information systems, while the micro-service architecture has recently been rapidly gaining popularity in commercial applications [9].

Improving the usability of micro-services development and operation has led to the abandonment of some of the basic capabilities that were originally present in the "classic" service-oriented architecture. Specifically, SOA's protocol-level integration, or so-called “contract decoupling”, allows a service and its consumers to evolve separately while maintaining a consistent description of the interaction interface.
Micro-services architecture does not “support contract decoupling”, although this is one of the main features of a service-oriented architecture. This architecture is suitable for building research information systems operating in a continuous mode, as well as in real time.

2. Methods and materials

A rational choice of a SP architecture for the distributed assessment of the forest vegetation state, including that carried out in the framework of international projects [1], involves a multicriteria comparative analysis of various possible SP configurations. Moreover, the criteria used can be assessed using both quantitative and qualitative scales. The analysis showed that to solve such problems, it is advisable to use the approach used to solve weakly structured multicriteria problems [10]. The essence of this approach is in the combined use of methods of verbal analysis of decisions (simple and complex supporting situations of the survey) and methods of converting qualitative indicators into quantitative indicators based on the joint application of the mathematical apparatus of the theory of fuzzy sets, relations and measures, as well as the theory of the experimental design.

The procedure that implements this approach assumes, as the first step, the formation of a set of indicators by which the assessment and selection of the specific architecture of a software complex is carried out. These indicators include:

Indicator of modularity. This indicator is specifically used for options of information system creating within the framework of projects for assessing the forest vegetation state carried out by various organizations, when it is necessary to combine the results generated by separate distributed calculation modules. Modularity is assessed based on the concepts of "cohesion" and "coupling" used in the ISO / IEC / IEEE 24765 standard [11]. Connectivity can be assessed on a scale from 1 (the case of software libraries, in which the functions in modules are combined thematically) to 5 (for web services, where a certain task is solved entirely at the expense of internal functions). Likewise, in the case of engagement, the highest score is given if the modules can function independently of each other.

Indicator of permissible heterogeneity. For each component, the most appropriate means of its implementation are selected. In addition, if there are requirements for the openness of a software package, it is allowed to embed third-party software solutions into it. Specified properties of the complex imply the functioning of its components in various, sometimes incompatible software environments. The indicator is formed as the sum of particular indicators characterizing acceptable options for implementing an application for a number of popular programming languages and operating systems.

Performance indicator. The presence of time-consuming tasks and the need for their prompt solution require high computational efficiency. Numerically, this indicator is estimated by the time it takes to solve a test mathematical problem by implementation of the same code, but organized using different architectures.

Multi-user indicator. The inherent interdisciplinarity of environmental modeling may require a range of experts and researchers to work together. In this case, a software package should provide convenient means of multi-user work. The indicator has the lowest value for single-user mode, and the highest, for architecturally unlimited number of users.

Scalability metric. It evaluates the ability of the architecture to quickly increase the overall performance of a software package when the load increases. At the same time, in the absence of the load, the software package should not consume a large amount of computing power. To obtain a numerical estimate of the indicator, the ratio of the time of solving a thousand test mathematical problems on one server to the time of solving them by a hardware-software complex of three identical servers is estimated.

The algorithm for multicriteria selection based on the described composition of indicators is as follows. Let the set of architectures for building software systems be evaluated by a set of the above quality indicators \( F = \{ F_1, F_2, \ldots, F_m \} \), each of which is a linguistic variable. In this case, the linguistic variables \( F_j = \) "Modularity, Heterogeneity, Performance, Multi-user mode, Scalability" can take values from a variety of simple and compound terms \( T(F_j) = \{ \text{"low"}, \text{"below average"}, \text{"average"}, \text{"above} \).
average", "high"}. For a qualitative interpretation of the resulting indicator that evaluates the generalized quality of a particular architecture, the linguistic variable "Efficiency" is used, which can take on the values \( T(F_{\text{res}}) \) {"bad", "below average", "average", "above average", "good"} [12, 13]. In the most general form, the decision maker's knowledge about the relationship between the individual quality indicators \( F = \{F_1, F_2, \ldots, F_m\} \) and the resulting indicator \( F_{\text{res}} \) can be represented by production models of the form:

\[
P_j : \, \text{IF} \, \, F_1 = A_{1j} \, \text{and} \, \, F_2 = A_{2j} \, \text{and} \, \ldots \, \text{and} \, \, F_m = A_{mj}, \, \text{THEN} \, \, F_{\text{res}} = A_{f_{\text{res}}},
\]

where \( A_{ij} \in T(F_i), A_{f_{\text{res}}} \in T(F_{\text{res}}) \) are the terms of the corresponding linguistic variables. The bipolar scale \([-1, 0, +1]\) is used as a general scale for all values of the indicators, and terms \( A_{ij}, A_{f_{\text{res}}} \) can be set using fuzzy numbers (L-R) of the type (figure 1).

**Figure 1.** Linguistic variable terms on the scale \([-1, +1]\).

The extreme ("minimum" and "maximum") values of the linguistic variable \( F_i \) are marked "-1" and "+1". To build a resulting indicator \( F_{\text{res}} \), according to the theory of experimental design, an orthogonal plan of an expert survey is built, the elements of which are the extreme values of individual effectiveness indicators \( \{F_1, F_2, \ldots, F_m\} \).

Further, to solve the problem of a reasonable choice of SP architecture, the following steps are performed:

**Step 1.** Formation of a set of linguistic scales for each indicator and the resulting quality indicator of the compared architectures of software systems. Conversion of all indicators to the scale \([-1, +1]\).

**Step 2.** Creation of an orthogonal survey plan for experts and conducting an expert survey (answers to questions of the production rules).

**Step 3.** Construction of the resulting quality indicator for comparable architectures.

One of the advantages of the proposed methodology is modification of the procedure for obtaining expert answers to the questions contained in production rules. So, for example, in the case of a standard approach to performing an expert survey with the number of quality indicators exceeding 5, the number of questions asked is above 32, which, as a rule, leads to inconsistency in expert statements due to peculiarities of human thinking. In the methodology discussed, it is proposed to conduct an abbreviated expert survey. Production rules, in which all quality indicators, except one, take "low" or "high" values, are called simple rules for interviewing an expert, or simple reference situations. The number of such situations corresponds to the number of particular indicators of the SP architecture quality. Composite rules (complex support situations) are represented using simple support situations with production rules of a special kind. It is proposed to estimate the resulting indicator in complex support situations by building a constructive parametric \( \lambda \)-fuzzy Sugeno measures [14] on a finite set of simple support situations. The Sugeno measures reflect the estimate of the resulting indicator in a complex rule. The resulting estimates of complex rules are used to check the decision maker's statements for consistency. So, for example, when answering a complex rule, the relative indicator obtained on the basis of the Sugeno measures is greater than the specified error value, then it is concluded that the expert gave an incorrect answer. The revealed contradictions are presented to decision makers for their analysis and elimination.
3. Results and Discussion
As a result of the described methodology implementation for choosing a SP architecture for the forest vegetation state assessment, a final table with normalized values of individual quality indicators was made (table 1).

| Architecture of software systems          | Normalized values of quality indicators |
|------------------------------------------|----------------------------------------|
|                                          | Modularity | Heterogeneity | Efficiency | Multi-user mode | Scalability |
| Monolithic architecture (mon)            | 0.2        | 0.1           | 1.0        | 0.1             | 0.1         |
| Modular architecture (mod)               | 0.4        | 0.1           | 1.0        | 0.1             | 0.1         |
| Component architecture (com)             | 0.6        | 0.3           | 0.9        | 0.1             | 0.1         |
| Client-server architecture (csa)         | 0.8        | 0.8           | 0.4        | 0.5             | 0.4         |
| Service-oriented architecture (soa)      | 1.0        | 1.0           | 0.3        | 1.0             | 1.0         |

The calculation of the resulting indicator values according to the proposed methodology requires filling in the expert survey matrix using extreme values (−1\(i_F\) − «low», +1\(i_F\) − «high») of the indicators \(F_i\) (Modularity, Heterogeneity, Performance, Multi-user mode, Scalability). Generalized expert estimates for simple support situations are presented in table 2. It is in these estimates that experts’ ideas about the suitability of a particular architecture for creating a specific software package are taken into account, expressed through combinations of indicator values.

| Rule | \(F_1\) | \(F_2\) | \(F_3\) | \(F_4\) | \(F_5\) | \(F_{res}\) |
|------|---------|---------|---------|---------|---------|-------------|
| 1    | High    | Low     | Low     | Low     | Low     | 0.2         |
| 2    | low     | High    | Low     | Low     | Low     | 0.3         |
| 3    | Low     | Low     | High    | Low     | Low     | 0.3         |
| 4    | Low     | Low     | low     | High    | Low     | 0.8         |
| 5    | Low     | Low     | Low     | Low     | High    | 0.6         |

To estimate the resulting indicator in complex reference situations, taking into account the opinions of an expert in simple reference situations, the parameter −1<\(\lambda^*\)<∞ of the Sugeno fuzzy measures is calculated by solving the equation

\[
\frac{(1+0.2\lambda)(1+0.3\lambda)(1+0.3\lambda)(1+0.8\lambda)(1+0.6\lambda)-1}{\lambda} = 1.
\]

Taking into account \(= -0.9596\), expert estimates of the resulting indicator in difficult reference situations and the resulting indicator of the quality of the SP architecture are calculated according to the proposed methodology. The calculated values of the resulting architecture efficiency indicator will be equal to: \(F_{res}(mon) = 0.711\), \(F_{res}(mod) = 0.754\), \(F_{res}(com) = 0.813\), \(F_{res}(csa) = 0.891\), \(F_{res}(soa) = 0.931\).

Thus, as a result of implementation of the proposed multi-criteria assessment procedure, a reasonable choice was made in favour of service-oriented architecture. Since for the SP analysis of the forest vegetation state, the multi-user mode is important, as well as the scalability of the SP, the result looks reasonable. In addition to justifying the choice of architecture, the proximity of the numerical values of the resulting indicator is of great importance. Thus, client-server and service-oriented architectures differ insignificantly in the resulting indicator, which means that they are interchangeable.
4. Conclusion
The result of choosing a SP architecture does not mean the absolute objective superiority of service-oriented architecture over other options. Numerical assessments of particular indicators characterizing the quality of alternative architectures, as well as a generalized indicator of their effectiveness, allow, situationally and reasonably, to switch to other architectures of the information platform for solving the corresponding types of environmental problems.

The proposed procedure for the multicriteria choice of a SP architecture can be applied once again when obtaining more detailed information both on specific implementations of individual modules of the hardware and software system, and on the indicators with the help of which their quality is assessed. In this case, a more detailed description of both the architecture and the corresponding features of its implementation will be obtained.

Acknowledgments
The studies on the topics presented were carried out within the framework of the budget theme No. 0073–2019–0004, and “Innovative information technologies for analysis of negative impact on the cross-border region forests (KS1309 InnoForestView)” project of the South-East Finland – Russia CBC Programme 2014-2020.

References
1. Project KS1309 Innoforestview - Innovative information technologies for analysis of negative impact on the cross-border region forests. Available at: https://innoforestview.site/ (accessed: 28.02.2021)
2. Sovetov B Ya, Vodyakho A I, Dubenetsky V A and Tsekhansky V V 2012 Architecture of information systems [Arhitektura informatsionnyh sistem - in Russian] (Moscow: Publishing center "Academy") p 288
3. Greenfield J, Short K, Cook S, Kent S and Krupey J 2007 Software development factories: stream assembly of typical applications, modeling, structures and tools] (Moscow: ID Williams LLC) p 592
4. Gouigoux J P and Tamzalit D 2017 From Monolith to Microservices: Lessons Learned on an Industrial Migration to a Web Oriented Architecture Proceedings of IEEE International Conference on Software Architecture Workshops (ICSAW’2017) pp 62–65
5. Kalandarev A 2017 Modern web architecture. From monolith to microservices System administrator [Sistemnyi administrator - in Russian]. No 1-2 pp 80–83
6. Richards M 2015 Software Architecture Patterns O'Reilly Media, Inc. p 47
7. Oluwatosin H S 2014 Client-Server Model IOSRJ Comput. Eng. vol 16, No 1, pp 2278-8727
8. Papazoglou, M P and Van Den Heuvel W 2007, Service oriented architectures: Approaches, technologies and research issues, VLDB Journal, vol 16, No 3, pp 389-415
9. Bakshi K 2017 Microservices-based software architecture and approaches // Proceedings of the IEEE Aerospace Conference. pp. 1–8.
10. Pavlov A N, Pavlov DA, Pavlov A A and Slin’ko A A 2017 The Technique of Multi-Criteria Decision-Making in the Study of Semi-Structured Problems, Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017). April 2017. – (Switzerland: Springer International Publishing), vol 2: Cybernetics and Mathematics Applications in Intelligent Systems. pp 131-140
11. McConnell S 2010 Perfect code. 2nd ed. Code Complete (Moscow: Russian edition), p 896
12. Kopytov E A, Pavlov A N and Zelentsov V A New Methods of Calculating the Genome of Structure and the Failure Criticality of the Complex Objects' Elements Transport and Telecommunications, 2010, vol 11, No 4 pp 4-13
13. Pavlov A N, Pavlov A A, Pavlov D A and Slin’ko A A 2017 Combined method of multicriteria choice of management decisions based on models of knowledge representation and experiment planning. *Proceedings of the A.F. Mozhaisky Military and Space Academy* Issue 656 pp 9-17

14. 1986 Fuzzy Sets in Control Models and Artificial Intelligence ed D A Pospelova (Moscow: Science) p 312