

Development of a web application for applying the analytic hierarchy process

B. E. Arbuzov, B. L. Tsengel, E. A. Osipenko, A.V. Dyshkantuk

Abstract—The article is devoted to the software implementation of solving the choice problem using the Analytic Hierarchy Process (AHP) method developed by T. Saaty, utilizing the React library. The formulation of the choice problem to determine the best alternative for a system is described, as well as the issues that arise in its solution, and the software implementation of the AHP method, based on a developed web application, is presented. The process of program development is also described in detail, including the creation of an interface for evaluating criteria and alternatives. The advantages of using this method for calculating decision assessments and the JavaScript library React for developing the user interface that implements the AHP are discussed. Measures have been proposed to increase the accuracy of the results obtained using the AHP, and to improve the performance of the web application developed based on the React library.

Keywords—choice problem, decision-making, analytic hierarchy process, inversely symmetric matrix, web application, JavaScript, React, virtual DOM, webhook.

I. INTRODUCTION

In the practice of systems engineering [1], one of the most complex tasks is the choice of a system variant based on a set of characteristics. The complexity of the problem is determined primarily by the subjectivity of both the formulation of the problem itself (the goal is determined by the decision-maker - hereinafter referred to as the DM) and the subjective choice of the preferred decision-making criteria, as well as the target indicators.

Another complexity of the problem is the difficulty of obtaining correct estimates of the parameters of the variants, as the judgments of the DM are usually evaluated qualitatively (semantically) with difficult-to-formalize values, which, taking into account again the opinion of the

DM, must be translated into numerical characteristics, usually on a scale no higher than the ratio scale [2].

Even when the choice problem is correctly formulated in the mathematical sense (analytically), it is not possible to avoid the subjectivity of the DM and the difficulty in formalizing the parameters when solving it.

The verbal formulation of the choice problem is as follows: there are several variants of a system (processes, objects, etc.), each of which has corresponding parameters and structure. It is necessary to select from these variants the system with the best combination of parameters in terms of efficiency and cost.

II. FORMULATION OF A PROBLEM

The mathematical (set-theoretic) formulation of the problem of choosing a system variant based on a set of characteristics is as follows:

Given:

The set of possible system variants, the structure of which consists of the corresponding subsystems and elements:

$$W = \left\{ \langle z_k, \xi_{\kappa_j} \rangle \mid j \in \overline{1, K_j}, \kappa \in \overline{1, K} \right\} \quad (1)$$

For simplicity, three levels of the hierarchical structure of the system are presented. In applied problems of systems engineering, there may be more, which, accordingly, will increase the number of variables in the tuple.

$$z_k = \begin{cases} 1 - k \\ 0 - k \end{cases} \quad (2)$$

1 – the k -th functional subsystem is included in the structure of the system's appearance, and 0 – the k -th functional subsystem is not included in the structure of the system's appearance.

$$\xi_{ij} = \begin{cases} 1 - j \\ 0 - j \end{cases} \quad (3)$$

1 – the j -th type of element is included in the composition of the 1 – the k -th functional subsystem, 0 – the j -th type of element is not included in the composition of the 1 – the k -th functional subsystem.

Parameters of subsystems (or requirements for them), i.e. their target characteristics:

Manuscript received June 21, 2024.

B. E. Arbuzov, student of the Saint-Petersburg State Marine Technical University, Saint Petersburg, Russia (e-mail: boggzon@gmail.com). SPIN-code: 7748-7127.

B. L. Tsengel, student of the Saint-Petersburg State Marine Technical University, Saint Petersburg, Russia (e-mail: tsengel01@bk.ru). SPIN-code: 6072-3568.

E. A. Osipenko, candidate of Philological Sciences, associate professor of the Department of Foreign Languages, Saint-Petersburg State Marine Technical University, Saint Petersburg, Russia (e-mail: elena.osipenko12061976@gmail.com). SPIN-code: 6023-7064.

A. V. Dyshkantuk, Doctor of Technical Sciences, Professor of the Department of Computer Engineering and Information Technologies, Saint-Petersburg State Marine Technical University, Saint Petersburg, Russia (e-mail: dyshkantuk@yandex.ru).

$$R_{TC} = \langle R_1, R_2, \dots, R_n \rangle \quad (4)$$

Parameters of the system operating conditions (independent of the DM):

$$L = F(\tau, \omega, \gamma) \quad (5)$$

Set of parameters of supporting (interacting) systems:

$$H_{SUP} = \{H_1, H_2, \dots, H_k\} \quad (6)$$

Functional parameters of the system that determine its effectiveness:

$$G = F(R_{TC}, L, H_{SUP}) \quad (7)$$

Constraints:

$C \leq C_{SET}$ – cost constraints;

$V \leq V_{SET}$ – constraints on the feasible system parameters (structural constraints);

$M \geq M_{REQ}$ – constraint on the quality (effectiveness) of the system operation;

$T \leq T_{SET}$ – constraint on the time parameters of the system, which can be represented on an absolute scale (t_0 – system operational time) or complex indicators (K_R – readiness factor, K_{OR} – operational readiness factor, etc.) specified on a ratio scale.

Find:

Such a rational system variant, in which the required values of its effectiveness will be achieved, with the minimum costs for its operation (1), where C is the cost of the system operation (implementation of the target purpose), W – is one of the considered system variants, $G(W)$ – are the functional parameters of the system that determine its effectiveness, G^{REQ} – is the required level of system parameters that allows its functionality to be implemented:

$$W^* = \arg \min_{G(W) \in G^{REQ}} C(W) \quad (8)$$

Such a problem formulation is optimization-based, but the functional, i.e. the parameters that influence the result of the problem solution, are the prerogative of the DM, and if some of them (or all) do not have a measurable numerical value, then judgments about their priority and values are also developed based on the opinion of the DM.

There are many methods to give «objectivity» to the subjective opinions of the DM, which are referred to as expert assessment methods.

One of the widely used methods for solving the problem in the presented formulation is the Analytic Hierarchy Process (hereinafter AHP).

AHP, developed by Thomas Saaty [3], is a mathematical tool for the systematic comparison and evaluation of alternative options based on their relative importance in the hierarchical structure of goals or criteria. This method is based on the principle of pairwise comparisons, where experts assess the relative importance of each pair of alternatives, which then allows the hierarchy of priorities to be identified and informed decisions to be made.

The AHP is considered to be a universal method due to the fact that it includes not only mathematical, but also

psychological aspects of the problem being studied. This is because the AHP helps to visually and simply represent a complex problem in the form of a hierarchy of simple steps, compare alternatives to each other, and assign a quantitative assessment to each of them [4].

The AHP is noted to have wide application in tasks related to multi-criteria decision making, strategic planning, forecasting, and even conflict resolution [5]. The use of this method is described as providing decision-makers with the ability to determine priorities and allocate resources in accordance with key success factors, which contributes to the development of effective strategies. The AHP is also said to allow for the assessment of probable scenarios based on a comparative analysis of influencing factors, and provides a structured approach to the analysis of conflict situations, the identification of key contradictions, and the search for compromised solutions.

The methodology for applying the AHP begins with the creation of a hierarchical structure that includes the goals, main criteria, and alternatives. An example of such a model is provided in Figure 1.

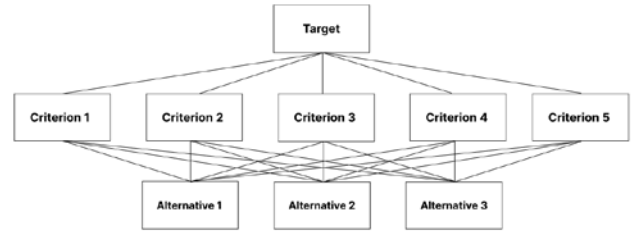


Fig. 1. Hierarchical structure model

At the top level, the goal is presented, which is evaluated using a number of criteria placed on the next level of the hierarchy. Each of these criteria has its own characteristic, evaluated either quantitatively or qualitatively (semantically). The collective interaction of these criteria in the modeling process allows the best solution to be chosen from the alternatives presented at the lower level of the hierarchy.

In the example presented above, the model is applied to a situation where a person wants to purchase a fitness club membership. The most significant criteria for the fitness center are described as the following: distance, size of the facility, attendance, price, and equipment. The names of the fitness centers serving as candidate solutions in this example are «Profitness», «Skulptors», «Legend».

III. DEVELOPMENT

The software implementation of the method is described as a web application that was developed in JavaScript using the React library, which is intended for the development of user interfaces. React library uses the virtual object model of the document (thereinafter virtual DOM), which structures the document, such as HTML or XML, in the form of a tree of objects, where the first level of the hierarchy is the document itself, and the nodes of the tree are the elements and attributes of the document [6]. React uses the virtual DOM to improve performance when updating the user

interface. All changes are first made to the virtual DOM, and then the virtual DOM is compared to the real DOM to determine the minimum number of changes required to update the interface. The JSX extension allows the creation of elements for displaying the DOM in React using syntax similar to XML, resulting in a highly understandable interface description [7].

Since React is not described as a standalone structure, it is said to be highly compatible and flexible, allowing it to be integrated with other libraries and frameworks. These tools have the ability to take on other tasks related to the web application, such as state management, route handling, and server-side rendering [8].

The web application interface is presented in Figure 2, where the user is able to enter the number of characteristics with their names, and similarly enter the evaluated systems that were presented in the example model above.



Fig. 2. Web application interface

In the code (Figure 3), the concept of webhooks [9] is used to dynamically track the entered characteristics and systems, through which the system names, their characteristics, and the quantity entered by the user are stored in arrays. By default, five characteristics and one system are assigned – these are the minimum scales.

```
const Data = () => {
  const [characteristicCount, setCharacteristicCount] = useState(5);
  const [characteristicNames, setCharacteristicNames] = useState(Array(5).fill(''));
  const [systemCount, setSystemCount] = useState(1);
  const [systemNames, setSystemNames] = useState(Array(1).fill(''));
}
```

Fig. 3. Webhooks example

After entering the names and clicking on the "Enter second-level matrix" button, an inversely symmetric matrix will be created, reflecting the relationships of each criterion to every other, which is filled out by the user based on their subjective judgments. After clicking the "Calculate normalized marks" button, the page will display a list of assessments for all criteria, in which the criterion with the highest assessment can be considered as the most significant (Figure 4).

	Distance	Size	Attendance	Price	Equipment
Distance	1	2	3	2	0.20
Size	0.50	1	0.50	3	4
Attendance	0.33	2.00	1	0.30	1
Price	0.50	0.33	3.33	1	3
Equipment	5.00	0.25	1.00	0.33	1

Calculate normalized marks:

- Normalized mark 1: 0.23
- Normalized mark 2: 0.24
- Normalized mark 3: 0.14
- Normalized mark 4: 0.22
- Normalized mark 5: 0.16

Fig. 4. Interface with second level matrix

Table 1 represents an inversely symmetric matrix, where A1, A2, ... An are a set of «n» elements and w1, w2, w3, ... wn are the corresponding weights or intensities.

Table I. Inversely symmetric matrix

	A1	A2	A3	...	An
A1	$\frac{w1}{w1}$	$\frac{w1}{w2}$	$\frac{w1}{w3}$...	$\frac{w1}{wn}$
A2	$\frac{w2}{w1}$	$\frac{w2}{w2}$	$\frac{w2}{w3}$...	$\frac{w2}{wn}$
A3	$\frac{w3}{w1}$	$\frac{w3}{w2}$	$\frac{w3}{w3}$...	$\frac{w3}{wn}$
...
An	$\frac{wn}{w1}$	$\frac{wn}{w2}$	$\frac{wn}{w3}$...	$\frac{wn}{wn}$

The assessments of the components are obtained as follows: the product for each row is found by multiplying the corresponding elements, and the nth root is extracted from the resulting product, where n is the number of elements in the row and a_i is the assessment of the component.

$$\sqrt[n]{\left(\frac{w1}{w1}\right)\left(\frac{w1}{w2}\right)\left(\frac{w1}{w3}\right)\dots\left(\frac{w1}{wn}\right)} = a_i \quad (9)$$

In the program code, to calculate the assessments, an empty resulting array is first created, in which the assessments will be stored. Then, in a double loop, a pass is made through the entire matrix, and the calculation of the assessments, described above and implemented in the program code in Figure 5, is performed. The calculated assessments are saved in the resulting array.

```
const calculateNormalizedMarks = () => {
  const marks = [];
  let sum = 0;
  for (let i = 0; i < characteristicCount; i++) {
    let product = 1;
    for (let j = 0; j < characteristicCount; j++) {
      product *= parseFloat(matrix[i][j]) || 0;
    }
    const mark = Math.pow(product, 1 / characteristicCount);
    sum += mark;
    marks.push(mark);
  }
  const normalized = marks.map(mark => (mark / sum).toFixed(2));
  onNormalizedMarksChange(normalized);
  setNormalizedMarks(normalized);
};
```

Fig. 5. Section of code with assessments calculations

The next step is the normalization of the assessments. Formally, this stage is described as follows: the sum of all the assessments calculated in the previous stage is found, and each of these assessments from the previous stage needs

to be divided by the found sum of all the assessments, where x_i is the assessment of the priority vector, and \sum is the sum of the assessments of all the components.

$$\frac{a_i}{\sum} = x_i \tag{10}$$

Thus, normalization is achieved, and the sum of the normalized assessments will give one.

The code fragment shown in Figure 6 uses the higher-order function map to iterate through each element of the array from the previous stage. For each element, it divides the element by the sum of all the elements. The program then uses the map function again, along with the previously described JSX extension, to display the normalized values in the user interface.

```
<h3>Normalized marks:</h3>
<ul>
  {normalizedMarks.map((mark, index) => (
    <li key={index}>{`Normalized mark ${index + 1}: ${mark}`}</li>
  ))}
</ul>
```

Fig. 6. Assessments normalization in code

The next step is the filling of the third-level matrices. In these matrices, all the systems are compared pairwise for a specific characteristic. In the given example, five third-level matrices will be created (Figure 7), as there are five characteristics in the example.

Characteristic 1

	Profitness	Skulptors	Legend
Profitness	1	2	3
Skulptors	0.50	1	1
Legend	0.33	1.00	1

Characteristic 2

	Profitness	Skulptors	Legend
Profitness	1	0.40	0.20
Skulptors	2.50	1	2
Legend	5.00	0.50	1

Characteristic 3

	Profitness	Skulptors	Legend
Profitness	1	1	1
Skulptors	1.00	1	0.5
Legend	1.00	2.00	1

Characteristic 4

	Profitness	Skulptors	Legend
Profitness	1	0.5	0.5
Skulptors	2.00	1	0.5
Legend	2.00	2.00	1

Characteristic 5

	Profitness	Skulptors	Legend
Profitness	1	0.33	2
Skulptors	3.03	1	1
Legend	0.50	1.00	1

Calculate normalized marks for all matrix Calculate global marks

Fig. 7. Interface with third-level matrices

These matrices, just like the second-level matrix, are filled out by the DM based on their subjective judgments. The calculation of the normalized assessments for the third-level matrices is carried out in the same way as for the second-

level matrix. After clicking the “Calculate normalized marks for all matrix” button, the calculated assessments will appear (Figure 8). These assessments allow the identification of the best system for a specific characteristic.

Calculate normalized marks for all matrix Calculate global marks

Global marks:

- System 1: 0.27
- System 2: 0.35
- System 3: 0.37

Fig. 8. Global assessments

As a result, the best system is system number 3, that is, the «Legend» fitness club. Global assessments are calculated by multiplying the elements of a row with the corresponding components of the normalized assessments of the second-level matrix, the products are summed, and the final assessments Y_i are obtained:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{matrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = Y_1 \\ \dots = Y_2 \\ \dots = Y_3 \\ \dots = Y_4 \end{matrix} \tag{11}$$

The columns are the transposed matrices obtained after calculating the third-level matrices, the components x_i are the normalized assessments of the second-level matrix. The implementation of the code for calculating the global assessments is presented in Figure 9.

```
const handleCalculateGlobalClick = () => {
  const calculatedGlobalMarks = [];
  if (normalizedMarks.length === 0 || normalizedMarks3.length === 0) {
    console.log("Normalized marks have not yet been calculated.");
    return;
  }
  for (let i = 0; i < systemNames.length; i++) {
    let mark = 0;
    for (let j = 0; j < characteristicCount; j++) {
      mark += parseFloat(normalizedMarks[j]) * parseFloat(normalizedMarks3[j][i]);
    }
    calculatedGlobalMarks.push(mark.toFixed(2));
  }
  setGlobalMarks(calculatedGlobalMarks);
};
```

Fig. 9. Section of code with global assessment calculations

The calculations are performed in the «Matrix3» component (Figure 10). Therefore, the normalized scores of the second-level matrix are passed to this component through input parameters as props. In the parent component, these data are passed as props to «Matrix3», where they are received and used.

```
<Matrix3 normalizedMarks={normalizedMarks} characteristicCount={characteristicCount}>
```

Fig. 10. «Matrix3» component

Thus, this type of component-based application architecture significantly simplifies its refactoring, making changes, and provides the ability for further expansion. Decomposing into separate components increases the flexibility and scalability of the entire application.

IV. CONCLUSION

As a result, using the JavaScript library React, a web application was developed that implements the AHP to determine the best solution to the choice problem. The analysis process was also illustrated with an example involving five characteristics and three systems. Based on the development process, the following conclusions can be drawn:

1. The AHP is universal, as it can be applied in a wide range of domains to solve various problems. However, the effectiveness of using AHP directly depends on the user's level of expertise in the specific area of its application.

Thus, to achieve maximum effectiveness, the use of AHP requires certain qualifications and a deep understanding of the subject area.

2. In addition to the user's level of expertise, the accuracy of the results obtained using the AHP is also influenced by the number of factors being evaluated. A larger number of factors allows for a more detailed assessment of the solution. However, as the number of factors and evaluated systems increases, it may be necessary to increase computational power to process the growing amount of data.

Thus, when applying AHP, it is necessary to strike a balance between the accuracy of the assessment and the available computational resources.

3. Using React for developing large-scale web applications has several advantages. Thanks to the virtual DOM, React allows efficiently updating only the parts of the interface that have changed, which significantly improves the performance of the application. Additionally, the component-based architecture of React promotes modularity and scalability of the code, which simplifies the development and maintenance of complex projects.

However, the large size of the JavaScript package required for React to work can slow down the loading of the application, especially on devices with limited capabilities or with a slow internet connection.

4. In the case of small web applications, the use of React can also be justified. Its component model and JSX syntax simplify the development and maintenance of even simple projects. However, the overhead associated with the virtual DOM can be more noticeable in small projects where performance is not critical. Thus, the choice of React for small projects requires careful analysis and comparison with other frameworks.

The developed web application is a compact and convenient solution for accessing the AHP [10]. The intuitive interface is said to significantly simplify the use of this method. This implementation can be further applied by a wide range of users – both by enterprises and individuals for personal purposes.

REFERENCES

- [1] GOST R 57193-2016 Systems and software engineering - System life cycle processes. - Moscow: Standartinform, 2016.
- [2] Kolechik, M.M. Qualimetry // M.M. Kolechik. Uchebnoe posobie. 2nd ed., stereotype. - Moscow: MGIU, 2004. - 200p.: ill.
- [3] Saaty, T.L. Decision Making. Analytic Hierarchy Process // T.L. Saaty. - Moscow: Radio i Svyaz, 1993.
- [4] Kurbanova E.R. Hierarchy analytic method: characteristics // Forum molodyh uchenyh. – 2019. – No. 5 (33). – P. 749-752.
- [5] Volokobinsky M.Yu., Pekarskaya O.A., Razi D.A. Decision making based on the hierarchy analysis method // Finansy: teorija i praktika. – 2016. – Vol. 20, No. 2. – P. 33-42.
- [6] Arbuzov B.E., Tsengel B.L., Osipenko E.A. «Using the React library in developing web application» // Sbornik materialov X Mezhdunarodnoy nauchno-prakticheskoy zaochnoy konferentsii «ETAP-2023». – P. 276-281.
- [7] Beteev K.U., Muratova G.V. The concept of Virtual Dom in the React.js library // Inzhenernyj vestnik Dona. – 2022. – No. 3 (87). – P. 170-180.
- [8] Gorbachev A. A., Gorbacheva E. S. Comparison of the classical web application implementation process and the React library approach // Molodoj issledovatel' Dona. – 2020. – No 1 (22). – P. 28-31.
- [9] Webhook // OkoCRM. – URL: <https://okocrm.com/glossary/vebhuk/> (date of request: 04.06.2024).
- [10] Saaty-hierarchy-analysis-method // GitHub. URL: <https://github.com/TsengelBair/Saaty-hierarchy-analysis-method> (date of request: 08.06.2024).