

*Розглядається проблема скорочення витрат на розробку інформаційних систем і програмних продуктів. Запропоновано замінити персонал ІТ-проектів в ряді процесів і робіт з розробки програмного продукту, які повторюються, інтелектуальними інформаційними технологіями за умови економічної доцільності подібної заміни. Як апарат побудови таких технологій запропоновано використовувати апарат штучних нейронних мереж.*

*Серед основних напрямів автоматизації процесів розробки інформаційної системи виділено задачу пошуку повторно використовуваних функцій для реалізації функціональної вимоги, яку висунуто до системи. Для розв'язання даної задачі запропоновано модифікувати формальний опис та структурну схему нейрону ADALINE. Суть модифікації полягає у використанні мереж фреймів для формальних описів повторно використовуваних функцій та функціональної вимоги до інформаційної системи. Порівняння цих формальних описів дозволяє виявляти повторно використовувану функцію, яка в максимально можливому ступені відповідає висунутій функціональній вимозі. Результатом вирішення задачі пошуку є формальний опис системної функціональної вимоги. Цей опис формується на основі формальних описів знайденої функції та висунутої функціональної вимоги.*

*Виходячи з даного представлення задачі пошуку, було розроблено спеціальний алгоритм навчання модифікованого нейрону. Суть цього алгоритму полягає в пошуку опису повторно використовуваної функції, який максимально подібний формальному опису функціональної вимоги до системи.*

*За результатами модифікації елементів було модифіковано структурну схему нейрону ADALINE. Запропонований підхід та розглянуто основні особливості архітектурних рішень з реалізації модифікованої структурної схеми*

*Ключові слова: функціональна вимога, функція, ADALINE, дублювання, асоціатор, алгоритм навчання, репозиторій*

# THE ADALINE NEURON MODIFICATION FOR SOLVING THE PROBLEM ON SEARCHING FOR THE REUSABLE FUNCTIONS OF THE INFORMATION SYSTEM

**Saif Q. Muhamed**  
PhD, Lecturer\*

E-mail: saifkassimm@gmail.com

**Mohammed Q. Mohammed**  
PhD, Lecturer\*

E-mail: mqmh82@gmail.com

**M. Evlanov**  
Doctor of Technical Sciences, Associate  
Professor\*\*

E-mail: maksym.ievlanov@nure.ua

**G. Kliuchko\*\***  
E-mail: halyna.kliuchko@nure.ua

\*Department of information technology and business  
University Information Technology  
and Communications  
Al-Nidhal str., Baghdad, Iraq, 00964

\*\*Department of information control system  
Kharkiv National University of Radio Electronics  
Nauky ave., 14, Kharkiv, Ukraine, 61166

## 1. Introduction

At present, the market for IT services is quite unstable. The Gartner study shows that the cost of IT services has significantly decreased over 2015–2016. Some growth in expenses for IT services observed in 2017 did not eliminate the uncertainty of the market dynamics [1]. It should be noted that the software segment remains the largest segment in the IT market today [1]. Therefore, one of the main problems requiring attention from consumers and providers of IT services is the problem of reducing the IT service development expenses.

One of the most important expense items in the IT projects of software product development is the staff expense item. This type of expenses includes, particularly, the expenses on hiring the staff to participate in the IT project, the IT salaries and emoluments, training, and upgrade of the IT

project staff skills. It should also be noted that the existing project management practices recommend that the project personnel be divided into groups of permanent employees and employees hired to participate in a particular project [2]. At the same time, existing models of maturity of IT companies assume improvement of software writing processes and the IT project management in the direction of ensuring repeatability and standardization of software products [3, 4]. This approach enables an assumption of the possibility of substitution of intelligent information technologies for the IT project personnel in a number of repeatable processes and software development works provided that such a replacement is economically viable. Such a solution to the problem of reducing expenses on staff participation in the development of information systems and software products for various purposes is relevant from theoretical and applied points of view.

---

## 2. Literature review and problem statement

---

At present, the problem of using intelligent ITs for automating the process of development of software products for various purposes is one of the most urgent problems of scientific studies in the IT field. At the same time, various neural nets (NN) are considered as the main tools of artificial intelligence suitable for solving this problem. For example, recommendations on application of various methods of artificial intelligence (including NN) in testing new software products are considered in [5]. The issues of NN application for estimating size of the developed software are considered in [6]. Application of NN for recognition of software design patterns based on the open source code of the product is considered in [7].

However, most researchers consider the works connected with classification and identification of reusable components of software products as the main field in which application of intelligent ITs is justified most of all. A review of the studies carried out in the field of NN application for identifying reusable components of software products and an original NN model for solving this problem are given in [8]. Classification of software components based on their technical description is considered in [9]. Application of NNs for solving the problem of estimating reusable components is considered in [10] based on the indicators of modularity, interface complexity, supportability, flexibility, and adaptability.

A similar problem is also solved in the process of selecting the repository-stored web services to form an internet environment that supports the reusable software components for both service providers and consumers. Since the potential of web services for service-oriented computations is universally recognized, the demand for an integrated infrastructure facilitating opening and publication of services is ever growing. In this connection, it was suggested in [11] to use the NN based methods of intellectual analysis of descriptions of individual web-services in a repository for solving the problem of searching for similar web-services. Application of NNs for solving the problem of classification of web-services as reusable, stand-alone software components applicable in satisfying the needs independently or in combination with other web services is considered in [12].

However, the solutions discussed in [6–12] require conversion of textual descriptions of software components, their templates and web services into sets of the numerical data necessary for the NN functioning. In addition, in the course of application of the results described in [11, 12], a necessity of reverse conversion of the numerical result of the NN operation to a web-service description arises in a number of cases. Such conversions are unique for each particular NN, they significantly complicate their implementation and are not applied anymore in any information system (IS) and software product development tools. This often leads to higher costs and, consequently, refusal to create and operate NNs as the elements of intellectual ITs for development of ISs or software products.

It should be noted that the initial data for the neural nets include their formal descriptions in the course of solving problems of classification, identification and search for reusable web services or software components. Such descriptions can be represented in the following ways:

a) a special model based on quantitative metrics for estimating the reuse of software components [10];

b) representation of a component or service as a set of formal descriptions of their functions or interfaces (including application of WSDL and similar formal languages) [9, 11];

c) a formal component description based on the diagrams of the UML class [6].

The latter method is especially promising since it allows one to solve the problems of classification, identification and search for reusable components already in the course of identifying and analyzing the requirements to the products being created. In this case, an additional problem arises: the problem of formal description of the requirements to the product or the IS realizable by reusing the existing components. It is suggested to use knowledge-oriented description of requirements. Particularly, application of ontologies in the engineering of requirements is considered in [13]. Study of knowledge mining and management for formation of the requirements specification is proposed in [14]. Application of the patterns based on frame networks to represent functional requirements at a knowledge level was suggested in [15].

The above analysis allows us to draw a conclusion on the prospects of the studies aimed at solving the problems of classification, identification and search for reusable components in the course of elucidation and analysis of requirements to the products being created. Such studies will significantly improve accuracy of cost estimation and reduce the time spent on development of information systems and software products.

---

## 3. The aim and objectives of the study

---

This work objective was to modify the model of the neuron enabling solution of the problem of search for a description of a reusable function for implementation of the functional requirement to the IS. This makes it possible to reduce the cost of development of IS and software products by excluding staff from the search for reusable functions.

To achieve this goal, solution of the following tasks was considered:

- to modify the mathematical model of the neuron used to solve the problem of searching for the description of a reusable function;

- to modify the neuron block diagram as applied to the specifics of solution of the problem of searching for the description of a reusable function;

- to consider the features of practical implementation of the proposed modifications.

---

## 4. Results of the modification of a mathematical model of mADALINE neuron

---

Currently, there is a fairly large number of neuron types. However, in order to verify the possibility of using them in the problem of searching for the description of a reusable function to implement the functional requirements to the IS, it is necessary to use the neurons most simple in their implementation. Therefore, a comparative analysis of the main existing types of neurons was made. Proceeding from this analysis, it was recommended to apply a neuron of the ADALINE type [16]. Neurons of this type can be used both as elementary neurons in the NN composition and independently in the problems of pattern recognition, signal processing and implementation of logical functions.

Block diagram of the ADALINE neuron is shown in Fig. 1 [17, 18].

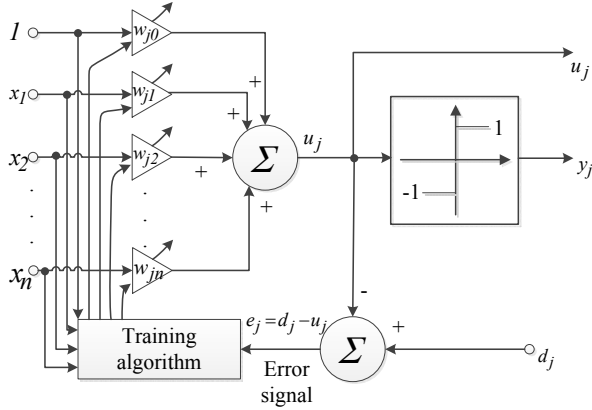


Fig. 1. Block diagram of the ADALINE neuron

The ADALINE neuron consists of two main parts: the adaptive linear associator and the nonlinear activation function. It has  $n+1$  inputs and two outputs: an analog,  $u_j$ , and binary,  $y_j$ . Besides, there is an additional input to which the training signal  $d_j$  is sent showing what should be a desired neuron response to each particular set of input signals. The analog output is the weighted sum of inputs,  $x_j$

$$u = \sum_{j=1}^n w_j x_j + \theta, \quad (1)$$

where  $w_j$  is the weight factor for the input  $x_j$ ,  $j=1, \dots, n$ ;  $\theta$  is a constant. The binary output,  $y_j$ , can take values  $+1$  or  $-1$  depending on polarity of the analog signal,  $u_j$ . The output signal,  $u_j$  is compared with the external training signal,  $d_j$ . The resulting error signal,  $e_j = d_j - u_j$ , enters the training algorithm which reconstructs the weights  $w_j$  so as to minimize some error function,  $e_j$ , called the training criterion. A quadratic function is most often used as such a function. This makes it possible to use for training not only the algorithm synthesized by Widrow and Hoff specially for the ADALINE neuron but also a number of recurrent procedures for adaptive identification [17, 18].

To apply the ADALINE neuron in solving the problem of search for a description of a reusable function to implement a functional requirement to the IS, the following should be determined:

- a formal description of the input signals,  $x_j$ ;
- a formal description of the training signal,  $d_j$ ;
- a formal description of the weight factors,  $w_j$ ;
- a formal description of the output analog signal,  $u_j$ ;
- a model of adaptive linear associator;
- a concrete type of the nonlinear activation function,  $y_j$ ;
- a concrete type of the training algorithm.

In general, the problem of searching for a description of a reusable function to implement a functional requirement to the IS requires comparison of formal descriptions of reusable functions and a formal description of the functional requirement to the IS by the Consumer of the IT services. Next, it is proposed to consider representation of the previously implemented  $j$ -th functional requirement to the IS at the level of knowledge,  $K_j^{f_{ib}}$  as the input signals,  $x_j$ . It is proposed to consider representation of the  $i$ -th functional requirement to the IS at the level of the Consumer's knowledge,  $K_j^{f_{ib}}$  as a training signal  $d_j$ . It is proposed to consider a system-wide representation of the  $j$ -th functional requirement to the IS at

the level of knowledge,  $K_j^{f_{is}}$ , as a description of the output analog signal,  $u_j$ . Formal descriptions of  $K_j^{f_{ib}}$ ,  $K_i^{f_{iv}}$  and  $K_j^{f_{is}}$  in the form of frame networks are given in [19].

It should be noted that the representations of requirements at the level of knowledge  $K_j^{f_{ib}}$ ,  $K_i^{f_{iv}}$  and  $K_j^{f_{is}}$  are the pluralities of numerical and symbolic data. The patterns of designing these representations are considered in [15]. However, the ADALINE neuron can process just numeric data. Conversion of representations,  $K_j^{f_{ib}}$  and  $K_i^{f_{iv}}$ , into vectors of numerical signals requires rather complicated algorithms. Therefore, it was proposed to modify the ADALINE neuron so that it can process the representation data based on their parameters computed during operation of the adaptive linear associator. The modified ADALINE neuron will be denoted hereinafter as mADALINE.

The objective of solving the problem of searching for a reusable function description is to find such a function the reuse of which would require minimal costs for adapting the found function to the features of the functional requirement. This objective can be represented as a search for such a description of a reusable function that duplicates to the greatest possible extent description of the functional requirement to the IS. Therefore, to develop a formal description of an adaptive linear associator, it is proposed to use the description of the problem of identifying duplicate functional requirements considered in [20, 21].

In general, the problem of identifying duplicate requirements can be formally described as follows. Let there be a frame net,  $Arch_{base}$ , consisting of a set of  $K_j^{f_{is}}$  representations. Each  $K_j^{f_{is}}$  representation is a collection of objects,  $ob$  (frames, interfaces, and links) description of which is given in [19]. The set of required descriptions of IT services,  $IT_{acm}$ , is a partition of the  $Arch_{base}$  network such that

$$\{IT_{acm_1}, \dots, IT_{acm_k}\} = Arch_{base}$$

and

$$IT_{acm_i} \neq \emptyset \wedge IT_{acm_i} \cap IT_{acm_j} = \emptyset \text{ for } 1 \leq i, j \leq k,$$

where  $k$  is the number of the IT services generated by the IS. Each IT service,  $IT_{acm_j}$ , has the following characteristics:

- $D(IT_{acm_j})$  is a set of unique objects;
- $Occ(ob, IT_{acm_j})$  is the number of occurrences of the  $ob$  object in the description of the IT service,  $IT_{acm_j}$ ;
- $S(IT_{acm_j}) = \sum_{ob \in D(IT_{acm_j})} Occ(ob, IT_{acm_j})$ ;
- $W(IT_{acm_j}) = |D(IT_{acm_j})|$ ;
- $H(IT_{acm_j}) = S(IT_{acm_j}) / W(IT_{acm_j})$ .

The cost function which makes it possible to estimate the degree of duplication of representations,  $K_j^{f_{is}}$ , generally takes the following form [20, 21]:

$$Profit(IT_{acm}, r) = \frac{\sum_{j=1}^k \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r \times |IT_{acm_j}|}}{\sum_{j=1}^k |IT_{acm_j}|}, \quad (2)$$

where  $|IT_{acm_j}|$  is the number of system-wide representations of functional requirements at a level of knowledge  $K_j^{f_{is}}$ , describing the IT service,  $IT_{acm_j}$ ;  $r$  is the repulsion factor,  $r \geq 1$ .

Then the problem of synthesizing versions of description of the created IS architecture can be formulated as follows: for the given  $D(Arch_{base})$  and  $r$ , find the partition:  $IT_{acm}$ :

$Profit(IT_{acm}, r) \in [Profit_{max} - \epsilon; Profit_{max}]$ , where  $Profit_{max}$  is the maximum value of the function (2),  $\epsilon$  is the permissible error value. In a case when the value of  $\epsilon$  cannot be determined using expert opinion methods, it is recommended that  $\epsilon = 0.1 \times Profit_{max}$  [20, 21].

In contrast to the problem of identifying duplicate requirements, the problem of searching for a reusable function description to implement a functional requirement to the IS requires a pairwise comparison of the  $K_i^{fv}$  representation with each of the  $K_j^{fb}$  representations at the disposal of the IT service provider. To describe this comparison, transform the cost function (2) to the following form:

$$Profit(IT_{acm}, r) = \frac{1}{\sum_{j=1}^k |IT_{acm_j}|} \times \sum_{j=1}^k \left( |IT_{acm_j}| \times \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \right). \quad (3)$$

With respect to the above, the parameters of expression (3) take the following values:

- a) the number of IT services being studied (functions of the created IS)  $k=1$ ;
- b)  $IT_{acm_j} = (K_i^{fv}, K_j^{fb})$ , since we describe one function by two compared requirements, hence  $|IT_{acm_j}| = 2$ ;
- c)  $r=1$  (each functional requirement to the IS is implemented as an independent module of the created IS and does not assume any further decomposition during the design of the IS and development of information and software support of the IS).

Then the function (2) for describing the adaptive linear associator of the mADALINE neuron will take the following form:

$$u_j = Profit(IT_{acm_j}) = \frac{S(IT_{acm_j})}{W(IT_{acm_j})} \rightarrow 2. \quad (4)$$

The value of function (4) will be 2 if the description of the reusable function  $K_j^{fb}$  exactly corresponds to the description of the Consumer's functional requirement,  $K_i^{fv}$ , to the IS. Therefore,  $d_j=2$  for any possible pair  $K_i^{fv}$  and  $K_j^{fb}$ .

The formal description of the weight factors,  $w_j$ , should be changed most strongly. For the proposed modification of the mADALINE neuron,  $w_j$  will represent not a vector but a single coefficient the formal description of which will have the following form:

$$w_j = K_j^{fb} \cup K_i^{fv}. \quad (5)$$

Then expression (4) can be written as follows:

$$u_j = Profit(w_j) = \frac{S(w_j)}{W(w_j)} \rightarrow 2. \quad (6)$$

However, such a formal description of the weight factor is only of value when solving the task of searching for the description of a reusable function to implement the functional requirement to the IS. As shown in [19], to describe the results of solving this problem, it is more expedient to use the system-wide representation of the functional requirement to IS at the level of knowledge  $K_j^{fs}$ . This representation can be formed on the basis of the result of computation of  $w_j$  as follows:

$$K_j^{fs} = K_j^{fb} \cup (w_j \setminus K_j^{fb}). \quad (7)$$

Such a system-wide representation of  $K_j^{fs}$  can be used in the future as a basis for forming a specification for reworking

of a reusable function taking into account features of the implemented Consumer's functional requirement to the IS of the IT services.

In a general case, solution of the problem of searching for description of the reusable function for implementation of the functional requirement to the IS is reduced to the choice of one of the following alternatives:

- a) recognition of the absence of reusable functions similar to the description of the Consumer/s functional requirement for the IS;
- b) searching for the description of one reusable function that is as close as possible to the description of the Consumer/s functional requirement  $K_i^{fv}$ , to the IS;
- c) searching for descriptions of two or more reusable functions which are as similar as possible to the description of the Consumer's functional requirement  $K_i^{fv}$ , to the IS.

Then description of the nonlinear activation function,  $y_j$ , of the mADALINE neuron can be performed based on the Heaviside's unit function. For the problem under consideration,  $y_j$  will have the following form:

$$y_j = \begin{cases} 0, & Profit(w_j) < 2 - \epsilon(k); \\ 1/2, & Profit(w_j) = 2 - \epsilon(k); \\ 1, & Profit(w_j) > 2 - \epsilon(k), \end{cases} \quad (8)$$

where  $\epsilon(k)$  is the permissible error value at the  $k$ -th training step.

Expression (8) requires paying special attention to the form of the instruction algorithm. In the course of solving the problem of searching for the description of a reusable function, not only numeric but also symbolic data are processed. Therefore, conventional algorithms (for example, the normalized least-squares algorithm or the Widrow-Hoff algorithm) are not suitable for training the mADALINE neuron.

For the simplest algorithm of training the mADALINE neuron in the course of solving this problem, an algorithm of searching for the maximal existing similarity is proposed. It consists of the following steps.

*Step 0.* Set the number of iterations  $k=0$  and the maximum permissible error value  $\epsilon(k)=0.1$ .

*Step 1.* Take  $j=1$ .

*Step 2.* If  $\epsilon(k) > 2 - Profit(w_j, k)$ , then take  $\epsilon(k) = 2 - Profit(w_j, k)$  and  $k=k+1$ .

*Step 3.* Take  $j=j+1$ . If  $j \leq n$ , then go to Step 2.

*Step 4.* If  $y_j=0$ , then take  $\epsilon(k) = \epsilon(k) + 0.1$ . Otherwise terminate the algorithm.

*Step 5.* If  $\epsilon(k) \leq 1$ , then go to Step 1. Otherwise, terminate the algorithm.

This algorithm will make it possible to find description of the reusable function even if the degree of its similarity to the functional requirement to the created IS is minimal.

---

## 5. Results of the modification of a block diagram of the mADALINE neuron

---

The results of modification of the mADALINE model represented by expressions (5)–(8) require a change of the block diagram of the given neuron as shown in Fig. 1. This change will determine in the future the main features of implementation of the mADALINE neuron in the form of a

software module of the intelligent IT for IS creation or modification for various purposes.

In the course of modification of the mADALINE neuron block diagram, the following features should be considered:

a) the library of representations of previously implemented functional requirements to the IS at the level of knowledge  $K_j^{fib}$  is a repository in which a finite number of these representations are stored;

b) the representation of the  $j$ -th Consumer's functional requirement to the IS at the level of knowledge  $K_i^{fv}$  enters the mADALINE neuron from the outside;

c) the value of  $u_j$  calculated from formula (6) and the value of  $K_j^{fis}$ , calculated by formula (7) should be transferred to the external user only in the case when  $y_j > 0$ .

Taking into account these conditions, the block diagram of the mADALINE neuron will have the form shown in Fig. 2.

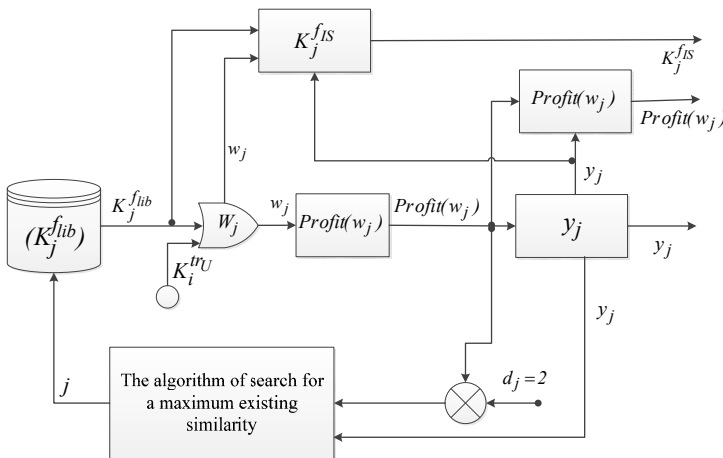


Fig. 2. Block diagram of the modified mADALINE neuron

### 6. Analysis of the practical implementation of the mADALINE neuron

The proposed block diagram of the mADALINE neuron determines the basic approach to the implementation of neurons of this type. This approach involves creation and operation of a special repository of functions available for reuse in subsequent IT projects of creation or upgrade of IS. This approach determines the following method of implementation of the mADALINE blocks shown in Fig. 2:

a) the blocks of data sampling from the  $w_j$  repository, formation of  $w_j$  and  $K_j^{fis}$  are executed using tools of the SQL language or equivalent languages;

b) the blocks of calculating the values of the  $Profit(w_j)$ ,  $y_j$  parameters and the block realizing the algorithm of searching for the maximum existing similarity are executed as software elements.

In general, the providing part of the mADALINE neuron can be implemented in the form of one of the following architectural solutions:

a) an independently existing functional module that can be integrated into the IS or IT of automated design of IS and software products for various purposes;

b) a separate web-service that can be integrated into existing service-oriented IS to implement the function of searching for web-services that maximally correspond to the given description.

To implement the offered modifications of the mADALINE neuron, it is expedient to use the intellectual IT of accelerated IS development considered in [22]. This IT is intended for automation of functioning of the following processes of the IS life cycle:

a) the process of determining the requirements of stakeholders;

b) the process of requirements analysis;

c) the process of designing the IS architecture.

A set of patterns of representation of functional requirements forms the basis of the intelligent IT of accelerated IS development. These patterns make it possible to represent the knowledge extracted from the requirement descriptions in the form of frame networks.

A fragment of the data diagram of the intelligent IT for accelerated IS development used in implementation of the mADALINE neuron is described in [22].

The following tables are used to implement the mADALINE neuron:

- APP\_REQUIREMENT.Requirement table which contains data on implemented functional requirements ( $K_j^{fib}$ ), available for reuse;

- APP\_REQUIREMENT.Implemented\_attribute table which contains data on the elements of frames describing the functional requirements available for reuse;

- APP\_ATTRIBUTE.Attribute table which contains data on the elements of the frames describing all functional requirements;

- APP\_ATTRIBUTE.Data\_type table which contains data on the types of data of the frame elements;

- APP\_ATTRIBUTE.Attribute\_type table which contains data on the types of frame elements;

- APP\_ATTRIBUTE.Attribute\_synonym table which contains data on possible synonyms for the frame element names;

- APP\_REQUIREMENT\_VERS.Analytical\_requirement\_version table which contains representations of the Consumer's functional requirements,  $K_i^{fv}$ ;

- APP\_REQUIREMENT\_VERS.Attribute\_in\_requirement table which contains data on the elements of the frames used for description of the Consumer's functional requirements,  $K_i^{fv}$ ;

- APP\_REQUIREMENT\_VERS.Analytical\_attribute\_version table which contains data on the elements of the frames used to describe all Consumer's functional requirements  $K_i^{fv}$ .

The  $w_j$  generation block is implemented as a usual SELECT query that selects descriptions of the Consumer's requirement frame elements and one of the analyzed requirements available for reuse. The following tables are used in this query:

a) APP\_REQUIREMENT.Requirement;

b) APP\_REQUIREMENT\_VERS.Analytical\_requirement\_version;

c) APP\_REQUIREMENT.Implemented\_attribute;

d) APP\_REQUIREMENT\_VERS.Attribute\_in\_requirement;

e) APP\_REQUIREMENT\_VERS.Analytical\_attribute\_version;

e) APP\_ATTRIBUTE.Attribute;

g) APP\_ATTRIBUTE.Attribute\_synonym.

The APP\_ATTRIBUTE.Attribute\_synonym table data are used in searching for possible synonyms for the names of the frame elements available for reuse. If this synonymous name coincides with the name of the frame element from the description of the Consumer's requirement, then the name of the frame element available for reuse is replaced by a synonymous name.

As a result of this query execution, a cursor is generated. It contains records on the frame elements describing both the Consumer's demand and the next analysis query available for reuse. The number of records in this cursor is the  $S(w_j)$  value. The number of records in the cursor formed as a result of executing the same SELECT query with an additional DISTINCT option enables calculation of the  $W(w_j)$  value. This query option can also be used in formation of a system-wide representation of the functional requirement to the IS at the level of knowledge,  $K_j^{fs}$ .

The  $S(w_j)$  and  $W(w_j)$  values enable calculation of the values of the  $Profit(w_j)$  and  $y_j$  parameters. These parameters are also used in implementation of the algorithm of searching for the existing maximum similarity. It should be noted that in implementation of this algorithm, the  $j$  parameter is the pointer of the current record in the APP\_REQUIREMENT.Requirement table and the  $n$  parameter is equal to the number of records in this table.

The considered version of the mADALINE neuron implementation makes it possible to fully automate solution of the problem of searching for the reusable function description. All actions in solution of this problem by these means are performed without analyst's participation.

Preliminary experiments were carried out on the application of the mADALINE neuron implementation in solving the problem of searching for the reusable function description. In order to evaluate the effect of using the mADALINE neuron in the course of solving the problem of searching for the reusable function description, experiments were carried out for the following information systems discussed in [22]:

a) functional problems of the League of Ukrainian Clubs of Smart Games Register information and analytical system (hereinafter, LUC Register IAS) were considered as a source of reusable functions;

b) the Virtual Bulletin Board information system was considered as a source of functional requirements for which search for reusable functions was necessary.

The data on the functional requirements of the LUC Registry IAS and the services implementing these requirements were used

as descriptions of reusable functions. In total, records of 15 such requirements were used in the experiments.

As shown in [22], in creation of the Virtual Bulletin Board information system, 6 descriptions of requirements to the LUC Registry IAS based on the fragments of APP\_CORE, APP\_USER, APP\_PRIVILEGE, APP\_PAGE, APP\_PUBLICATION, APP\_PERSON and APP\_GEOGRAPHY frame networks were reused. The decision on the reuse of the requirement descriptions was made based on the results of applying the method of forming representation of the  $i$ -th functional requirement to the IS at a level of the Supplier's knowledge described in [19]. This method enables search for descriptions of the functional requirements coinciding with new introduced requirements and already implemented by the information system developer.

As a result of the experiments, application of the mADALINE neuron has made it possible to find reusable descriptions of the same functions of the LUC Register IAS that were recommended with the use of the method of forming the representation of the  $i$ -th functional requirement to the IS at the Supplier's knowledge level. The main parameters that make it possible to compare the results of solving the problem of searching for reusable functions by applying the indicated neuron and the method are given in Table 1.

Table 1

The results of experiments on comparing solutions of the problem of searching the reusable functions with application of the mADALINE neuron and the method of forming the representation of the  $i$ -th functional requirement to the IS at the level of the Supplier's knowledge

Comparison parameter	mADALINE	The method of forming the representation of the $i$ -th functional requirement to the IS at the Supplier's knowledge
The problem of search for reusable function of creation of shared publications		
The time spent by the tool for solving the problem	0.9 s	1.5 s
The number of the reusable descriptions found	1	1
Necessity of human participation in the choice of the reusable description	No	No
The problem of search for the reusable function of the system user's data registration		
The time spent by the tool for solving the problem	1.3 s	1.7 s
The number of the reusable descriptions found	1	2
Necessity of human participation in the choice of the reusable description	No	No
The problem of search for the reusable function of the system user's location registration		
The time spent by the tool for solving the problem	1.8 s	2.3 s
The number of the reusable descriptions found	1	3
Necessity of human participation in the choice of the reusable description	No	Yes
The problem of search for the reusable function of system administration		
The time spent by the tool for solving the problem	0.8 s	1.2 s
The number of the reusable descriptions found	1	1
Necessity of human participation in the choice of the reusable description	No	No
The problem of search for the reusable function of the user's navigation in the system		
The time spent by the tool for solving the problem	0.9 s	1.3 s
The number of the reusable descriptions found	1	1
Necessity of human participation in the choice of the reusable description	No	No
The problem of search for the reusable function of delimitation of the user's right of the system access		
The time spent by the tool for solving the problem	0.9 s	1.2 s
The number of the reusable descriptions found	1	1
Necessity of human participation in the choice of the reusable description	No	No

The main disadvantage of the method of forming the representation of the  $i$ -th functional requirement to the IS at the level of the Supplier's knowledge is that he finds all possible descriptions of the implemented functional requirements to the LUC Registry IAS in which information from the requirement to the Virtual Bulletin Board information system is present. Therefore, there is a necessity of participation of a person who must solve the problem of choosing one of the descriptions found which will be further recommended for reuse.

The use of the mADALINE neuron to solve this problem makes unnecessary human participation. This becomes possible because during solution of the problem, the mADALINE neuron finds description of the LUC Registry IAS reusable function as much as possible coinciding with the analyzed functional requirement of the Virtual Bulletin Board information system.

To verify behavior of the proposed implementation of the mADALINE neuron on a large number of records of the implemented requirements available for reuse, additional studies are necessary.

---

### 7. Discussion of results of modification of the model and the block diagram of the mADALINE neuron

---

The offered results of modification of the mADALINE neuron completely exclude the analyst from the process of searching for reusable functions or web-services that satisfy the functional requirement. In accordance with the block diagram of the mADALINE neuron shown in Fig. 2, human participation is only required in preparation of representation  $K_i^{rv}$  of the Consumer's functional requirement to the created IS at the level of his knowledge. All other operations of search for a reusable function maximally similar to the description of the requirement are performed by the mADALINE neuron in an automatic mode.

The main feature of the proposed modification of the mADALINE neuron is the necessity of a formal description of both reusable functions and the presentation of the functional requirement to the IS. Although a large number of studies are carried out in this direction (e. g. [13–15]), the problem of formal description of requirements (including on the knowledge level) is still far from being resolved.

Among shortcomings of the proposed modification, it is necessary to emphasize considerable computational costs

for execution of the algorithm of searching for the existing maximum similarity. In the proposed algorithm version, one iteration requires  $n$  scans of the repository records ( $K_j^{fub}$ ). If the descriptions of the reusable functions do not completely correspond to the representation of the functional requirement  $K_i^{rv}$ , to the IS, the number of records in the repository will increase to  $10 \times n$ . Therefore, the issues of reducing computational complexity and increasing accuracy of the mADALINE training algorithm are a very promising study direction.

---

### 8. Conclusions

---

1. Formal descriptions of the input signals,  $x_j$ , the training signal,  $d_j$ , the weight factors,  $w_j$ , and the output analog signal,  $u_j$ , of the ADALINE neuron were modified. Based on the results obtained, a mathematical model of the adaptive linear associator, formal description of the nonlinear activation function,  $y_j$ , and the training algorithm for the ADALINE neuron were modified. In the course of modification, the features of solving the problem of searching for the description of a reusable function for implementation of a functional requirement to the IS were taken into account. The proposed solutions make it possible to use the capabilities of the modified mADALINE neuron in processing the representations of functional requirements and reusable functions in the form of frame networks.

2. Modification of the ADALINE neuron was proposed as applied to the peculiarities of solving the problem of searching for the description of a reusable function for implementation of the functional requirements to the IS.

3. An approach to the implementation of the proposed version of modification of the neuron block diagram was described. Two versions of architectural solutions for implementation of the providing part of the mADALINE neuron were proposed. The peculiarities of implementation of the mADALINE neuron as an element of intellectual IT for accelerated IS development were considered. The proposed version of implementation makes it possible to completely exclude the analyst from the process of solution of the problem of searching for the reusable function description. This will reduce the cost of developing or modifying the IS by reducing the labor costs of the analysts who previously participated in the search for the reusable function descriptions to realize functional requirements to the IS.

---

### References

1. IT (world market) // TAdviser. The State. Business. IT. URL: [http://www.tadviser.ru/index.php/Статья:ИТ\\_\(мировой\\_рынок\)](http://www.tadviser.ru/index.php/Статья:ИТ_(мировой_рынок))
2. Project Management Body of Knowledge. 5th ed. Newton Square: Project Management Institute, Inc., 2013. 586 p.
3. Biryukov A. Five steps to excellence // Director of Information Service. 2011. Issue 04. URL: <http://www.osp.ru/cio/2011/04/13008116/>
4. Terehov A. Modern software quality models // Interface. URL: <http://www.interface.ru/fset.asp?Url=/misc/qs.htm>
5. Gulia P. Nature inspired soft computing based software testing techniques for reusable software components // Journal of Theoretical and Applied Information Technologies. 2017. Vol. 95, Issue 24. P. 6996–7004.
6. Panyangam B., Kiewkanya M. Software Size Estimation in Design Phase Based on MLP Neural Network // Advances in Intelligent Systems and Computing. 2017. P. 82–91. doi: 10.1007/978-3-319-60663-7\_8
7. Software design pattern recognition using machine learning techniques / Dwivedi A. K., Tirkey A., Ray R. B., Rath S. K. // 2016 IEEE Region 10 Conference (TENCON). 2016. doi: 10.1109/tencon.2016.7847994

8. Chythanya N. K., Rajamani L. Neural Network Approach for Reusable Component Handling // 2017 IEEE 7th International Advance Computing Conference (IACC). 2017. doi: 10.1109/iacc.2017.0030
9. Automated architectural component classification using concept lattices / About N. A., Arevalo G., Falleri J.-R., Huchard M., Tibermacine C., Urtado C., Vauttier S. // 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture. 2009. doi: 10.1109/wicsa.2009.5290788
10. Singh C., Pratap A., Singhal A. Estimation of software reusability for component based system using soft computing techniques // 2014 5th International Conference – Confluence The Next Generation Information Technology Summit (Confluence). doi: 10.1109/confluence.2014.6949307
11. Hwang Y. Facilitating service matchmaking based on semantic similarity // Information (Japan). 2015. Vol. 18, Issue 11. P. 4443–4458.
12. Syed Mustafa A., Kumara Swamy Y. S. Web Service classification using Multi-Layer Perceptron optimized with Tabu search // 2015 IEEE International Advance Computing Conference (IACC). 2015. doi: 10.1109/iadcc.2015.7154716
13. Applications of ontologies in requirements engineering: a systematic review of the literature / Dermeval D., Vilela J., Bittencourt I. I., Castro J., Isotani S., Brito P., Silva A. // Requirements Engineering. 2015. Vol. 21, Issue 4. P. 405–437. doi: 10.1007/s00766-015-0222-6
14. Serna M. E., Bachiller S. O., Serna A. A. Knowledge meaning and management in requirements engineering // International Journal of Information Management. 2017. Vol. 37, Issue 3. P. 155–161. doi: 10.1016/j.ijinfomgt.2017.01.005
15. Levykin V., Ievlanov M., Neumyvakina O. Developing the models of patterns in the design of requirements to an information system at the knowledge level // Eastern-European Journal of Enterprise Technologies. 2017. Vol. 5, Issue 2 (89). P. 19–26. doi: 10.15587/1729-4061.2017.110586
16. Kliuchko H. G. Research of models and methods for determining the configuration variant of the information system // 21 International Youth Forum "Radio Electronics and Youth in the 21st Century". Kharkiv: KhNURE, 2018. Vol. 6. P. 133–134.
17. Rudenko O. G., Bodyanskiy E. V. Fundamentals of the theory of artificial neural networks. Kharkiv: Teletekh, 2002. 317 p.
18. Rutkovskaya D., Pilinskiy M., Rutkovskiy L. Neural networks, genetic algorithms and fuzzy systems. Moscow: Goryachaya liniya-Telekom, 2004. 452 p.
19. Ievlanov M. Methods of presenting formulated requirements to the information system at the level of knowledge // Eastern-European Journal of Enterprise Technologies. 2015. Vol. 4, Issue 3 (76). P. 4–11. doi: 10.15587/1729-4061.2015.47535
20. Ievlanov M. Development of the model and method of selecting the description of rational architecture of information system // Eastern-European Journal of Enterprise Technologies. 2016. Vol. 1, Issue 2 (79). P. 4–12. doi: 10.15587/1729-4061.2016.60583
21. Evlanov M., Vasilcova N., Panfyorova I. Modeli i metody syntezu opysu ratsionalnoyi arhitektury informatsionoyi systemy // Visnyk natsional'noho universytetu «L'vivs'ka politekhnika». Seriya: Informatsiyi systemy ta merezhi. 2015. Issue 829. P. 135–152.
22. Levykin V. M., Ievlanov M. V., Kernosov M. A. Pattern planning of requirements to the informative systems: design and application: monograph. Kharkiv, 2014. 320 p.