

BUILDING SOFTWARE OF INTELLIGENT HUMIDITY SENSOR

S. V. Artemova, P. A. Podkhvatilin, R. O. Sheludyakov

*Department «Design of Electronic and Microprocessor-Based Systems», TSTU;
sartemova@yandex.ru*

Key words and phrases: genetic algorithms; optimization of structural construction; software.

Abstract: The article discusses an approach to solving the problem of the structural construction of software by the example of intelligent humidity sensor, which represents an information-measuring system (IMS). The optimal structure of positioning of the developed algorithms used by IMS modules has been defined. The target function is the aggregate function of the intercom feature system that was maximized. In order to design the system with the required properties, we introduced the following limitations: on the single shot inclusion of algorithm in to set of program modules, duplication of memory by the information element, the complexity of the data interface between the individual modules. The solution to the problem was found using genetic algorithms. A similar approach was used for the design of software both of data measuring and data control systems.

The possibility of practical solution to the problems of structural construction are given in the theory of complexity of problems of selection, which show that the problem of even moderate size belongs to the class of NP-complete problems, which can be solved only approximately [3]. Therefore, most practical problems of structural construction can be solved using approximate (or heuristic) methods. These methods don't guarantee an optimal solution, but they lead to results that are close to optimal, with an acceptable expense of computing resources. Let's consider the structural construction of the software on the example of small Information-Measuring System (IMS) – an Intelligent Humidity Sensor (IHS). IHS allows indirect measurement of moisture of the material directly in the drying process in real time [1]. At the core of the functioning of IHS are neural networks.

An important step in designing IHS is creating of software systems that allows information-measuring system to have the following properties: loose coupling of elements (decomposition should be done such a way that the flow of information through the communication was minimal); testability (the ability to determine whether the proper functioning), diagnosability (the possibility of finding faults), ease of maintenance and operation (the minimal requirements to qualification of personnel), protection from unauthorized access, efficiency (economic efficiency in the operation), modifiability, functional extensibility, the ability to change the configuration, the

maximum duration of the life cycle (with no significant obsolescence), the minimum time for start-up [2].

The following modules are used as core modules of IHS at designing of software and information systems (Fig. 1): Intelligent Interface (**II**) module, supervisor module (**SV**), interrogation of Sensors Module (**IS**) and a Module for Implementation of Measurement Procedures (**MIMP**). Functioning of IHS is based on using of DataBase (**DB**) and Knowledge Base (**KB**).

The II module includes:

- cognitive graphics, which enables to perceive the process and search results visually, by establishing relationships between the images appearing on the screen and the cognitive processes of creative thinking;
- integrated development environment, which allows developers to make adjustments in the functional software modules and knowledge base;
- linguistic processor, which implements an intelligent dialogue of users with IHS on a limited natural language. The important operation of intellectual dialogue is the function of adding information which is necessary for IHS in the database.

Supervisor module is the main module of IHS, managing the processing, transmission and storage of information in IHS, and also the working of hardware and software tools.

The basic standard functional software modules of information-measuring subsystem of IHS include:

- interrogation of sensors module;
- implementation of measurement procedures module.

We used a series of algorithms for the implementation of IHS, which must be placed in modules so that the IMS possesses the above mentioned properties. We suggest solving optimization problem of structural construction using evolutionary methods, in particular, genetic algorithms. A table of arrangement of algorithms for the modules (Table 1) and the graph of interaction between algorithms through the data elements (Fig. 2) are given as the input data. Table 1 shows the various variants of distribution algorithms in modules. For example, the algorithm of preliminary calculations of input values can be placed in following modules: II, SV, IS, MIMP.

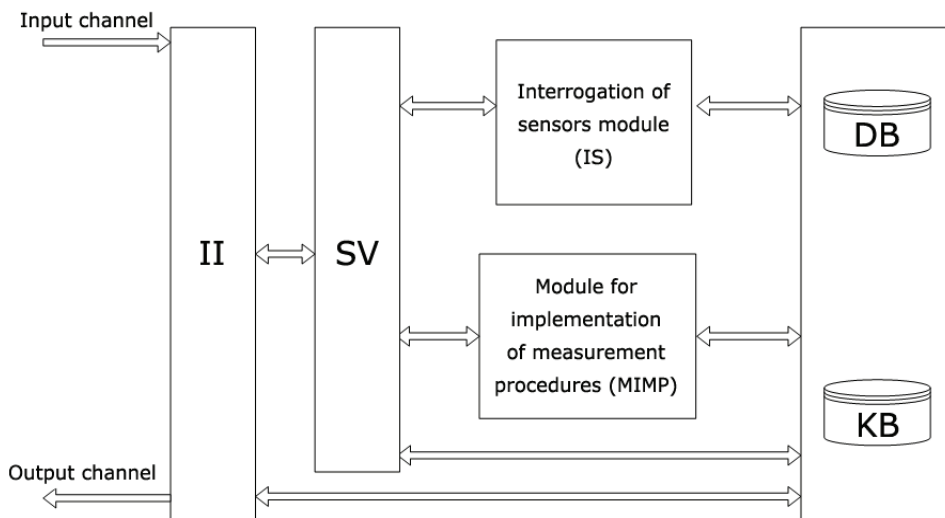


Fig. 1. Information model of IHS

Variants of distribution algorithms in modules

Number of algorithm	Name of algorithm	Name of module
1	A preliminary calculation of input values	II, SV, IS, MIMP
2	Normalization of input values	II, SV, IS, MIMP, DB, KB
3	Working with cognitive graphics	II
4	Denormalization of calculated output values	II, SV, IS, MIMP
5	Functioning of the database	SV
6	Functioning of the knowledge base	II
7	Survey of sensors	IS
8	Implementation of measurement procedures	MIMP
9	Functioning of the linguistic processor	II
10	Implementation of hardware tools	SV, IS
11	Implementation of software tools	SV, IS
12	Making decisions in the event of sensor failure	II, SV, IS, MIMP, KB
13	Implementation of different models of measuring procedures	KB, MIMP

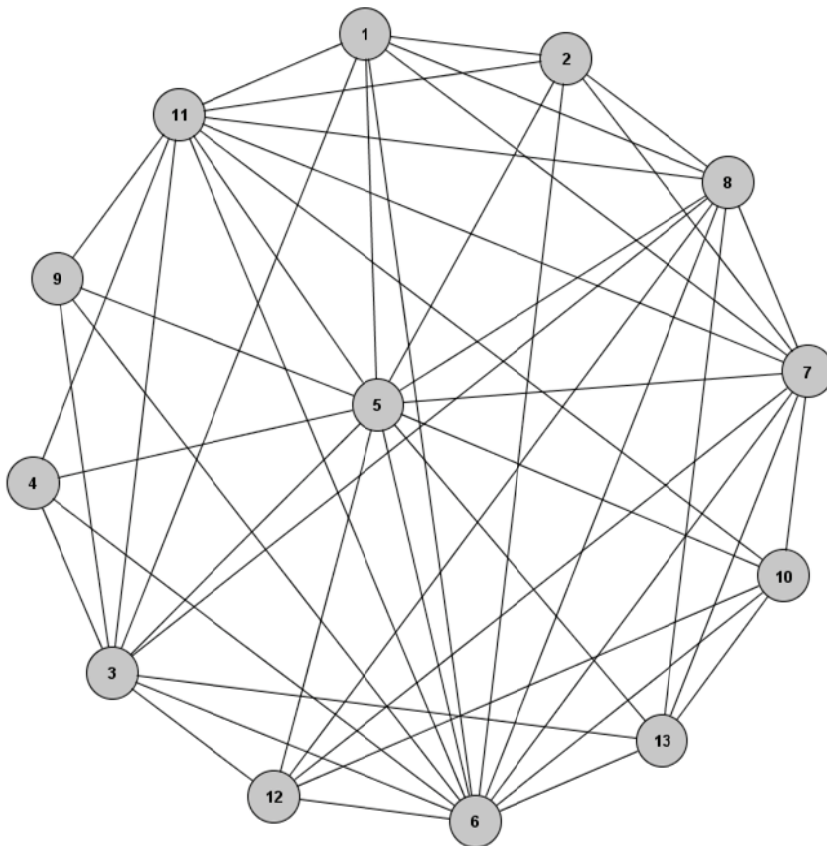


Fig. 2. Graph of the interaction of algorithms via information elements

<i>i</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
P	2	2	1	2	2	1	3	4	1	2	2	1	4

Fig. 3. Row vector **P**

We need to define an optimal placement of algorithms in modules using a genetic algorithm.

The solution of the problem is proposed to conduct using the technique proposed in [3].

1. We form the row vector **P**. These values are genes in the subsequent solution of the problem using genetic algorithms.

2. We describe the connection matrix **D**, “1” at the intersection of row and column of corresponding constructs means that there is symmetrical relationship between them. The elements of the main diagonal are considered to be “0” (constructive not linked with itself). The link is described only once for elements above the main diagonal because of emerging duality, the other elements equal to zero.

3. Let’s create a matrix of alternatives **X**. “1” is in relevant column on the intersection with the row whose number is equal to the number of the module, which construct belongs to a particular solution. The others elements are equal to “0”. The values of the matrix are linked with vector **P** by logical functions. The limitations of uniqueness of structural elements (the sum of the elements of columns is equal to “1”) and volume limitations of modules (the sum of elements of the rows is not greater than the capacity of a specific module which sets by the user) are superimposed on this matrix in the problem-solving process.

4. We form the matrix $\mathbf{Y} = \mathbf{X} \times \mathbf{D}$ and $\mathbf{W} = \mathbf{Y} \times \mathbf{X}^T$. The number of internal links equals to the sum of elements of main diagonal of matrix **W**. We need to increase the number of internal links based on the problem, because it means reducing the number of external links, i.e. to determine a more appropriate solution to the problem. This function is targeted and subjected to maximization by genetic algorithm.

Let’s form the row vector **P** according to this method (Fig. 3), the number of its elements corresponds to the number of distributable algorithms ($i = 13$). The immutable cells are the result of the restrictions painted gray. The elements of **P** are genes, and their values will be chosen in the solution process to achieve the optimum value of the target function. The range of their changes is limited. The element values (number) indicate that the corresponding algorithm belongs to the module with number, which is contained in cell. The modules are numbered (Fig. 4).

Next we will form the matrix **X** (Fig. 5). It has size $i \times j$, where i is the number of algorithms, and j is the number of modules. If the i -th algorithm is included in the j -th module, then cell i, j is “1”, otherwise cell is “0”. The data come from Fig. 3. The table is built in Excel automatically using logical functions (IF function). Cells, that are constantly equal to “0” as a result of the restrictions, painted gray. The service row “Sum” shows that the sum of the corresponding columns is equal to “1”, where 1 is the

<i>j</i>	Name of module
1	II
2	SV
3	IS
4	MIMP
5	KB
6	DB

Fig. 4. Numbering of modules

restriction for the ownership of the algorithm to just one module. We can see and control (limit) the fullness of modules via service column “Sum” (the sum of the corresponding rows).

Let’s fill the matrix **D** ($i \times i$) which shows the relationship between the algorithms (the relationship is symmetric, non-directional) (Fig. 6). If the modules are connected, then the intersection of the corresponding row and column is “1”, otherwise – “0”. The matrix is formed on the basis from the input data.

X	<i>i</i>													Sum
	<i>j</i>	1	2	3	4	5	6	7	8	9	10	11	12	
1	0	0	1	0	0	1	0	0	1	0	0	1	0	4
2	1	1	0	1	1	0	0	0	0	1	1	0	0	5
3	0	0	0	0	0	0	1	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	1	0	0	0	0	1	2
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sum	1	1	1	1	1	1	1	1	1	1	1	1	1	

Fig. 5. Ownership of the *i*-th algorithm to the *j*-th module

D	<i>i</i>												
	<i>i</i>	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	1	1	1	1	0	0	1	0	0
2	0	0	0	0	1	1	1	1	0	0	1	0	0
3	0	0	0	1	1	1	0	1	1	0	1	1	1
4	0	0	0	0	1	1	0	0	0	0	1	0	0
5	0	0	0	0	0	1	1	1	1	1	1	1	1
6	0	0	0	0	0	0	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	1	0	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	1	1	1
9	0	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 6. Relationship between the algorithms

We also use the service matrices \mathbf{P}^T and \mathbf{X}^T , which are transposed matrices \mathbf{P} and \mathbf{X} . Matrix \mathbf{Y} is determined by multiplying of \mathbf{X} on \mathbf{D}

$$\mathbf{Y} = \mathbf{X} \times \mathbf{D}.$$

Matrix \mathbf{W} is determined by multiplying of \mathbf{Y} on \mathbf{X}^T

$$\mathbf{W} = \mathbf{Y} \times \mathbf{X}^T.$$

The target function is the sum of the elements of main diagonal of the matrix \mathbf{W} and will be maximized by changing the values of the genes from vector \mathbf{P} using genetic algorithms

$$f = \sum_{\substack{i=1 \\ j=1}}^{13} \mathbf{W}_{ij}, \quad i = j.$$

Table 2

Distribution algorithms in modules

Number of algorithm	Name of algorithm	Name of module
1	A preliminary calculation of input values	SV
2	Normalization of input values	SV
3	Working with cognitive graphics	II
4	Denormalization of calculated output values	SV
5	Functioning of the database	SV
6	Functioning of the knowledge base	II
7	Survey of sensors	IS
8	Implementation of measurement procedures	MIMP
9	Functioning of the linguistic processor	II
10	Implementation of hardware tools	SV
11	Implementation of software tools	SV
12	Making decisions in the event of sensor failure	II
13	Implementation of different models of measuring procedures	MIMP

We use the following settings for the implementation of the process of selecting a suitable combination: population size is 50; length of chromosome is 8 bit; type of chromosomes is continuous; probability of crossover is 0.9; mutation probability is 0.01; degree of renovation is 0.98; elitism strategy is on; diversity operator is on; required to find is maximum of function f .

We use the following limitations to solve this problem:

– restriction of range for the chromosomes

$$1 \leq P_i \leq 6, \quad i = \overline{1, 13};$$

– restriction of the uniqueness of the algorithms

$$\sum_{i=1}^6 X_i = 1, \quad i = \overline{1, 13};$$

– placing restrictions for specific algorithms (from the condition):

$$1 \leq P_1 \leq 4, \quad 1 \leq P_4 \leq 4, \quad 2 \leq P_{10} \leq 3, \quad 2 \leq P_{11} \leq 3, \quad 1 \leq P_{12} \leq 5, \quad 4 \leq P_{13} \leq 5.$$

The solution to the problem of distribution of algorithms in functional software modules as a result of optimization using a genetic algorithm is presented in Table 2.

The solution, not contradicting with common sense of placement algorithms in the software modules, was found in the process of optimization of constructing of software for IHS using a genetic algorithm. Thus, we can conclude that this approach can be applied to the design of information systems and software both for data measuring and data control systems, which have a large number of modules and algorithms in its composition.

This work was partially supported by grants from the Russian Foundation for Basic Research No. 14-08-00198-A and No. 12-08-00352-A.

References

1. Artemova S.V. *Programmnye produkty i sistemy*, 2013, no. 3, pp. 272-276.
 2. Denisenko V.V. *Komp'yuternoe upravlenie tekhnologicheskim protsessom, eksperimentom, oborudovaniem* (Computer control of technological process, experiment, equipment), Moscow: Goryachaya liniya – Telekom, 2009, 608 p.
 3. Norenkov I.P. *Osnovy avtomatizirovannogo proektirovaniya* (Fundamentals of CAD), Moscow: Izdatel'stvo MGTU im. N. E. Baumana, 2002, 336 p.
-

Построение программного обеспечения интеллектуального датчика влажности

С. В. Артемова, П. А. Подхватилин, Р. О. Шелудяков

*Кафедра «Конструирование радиоэлектронных и микропроцессорных систем»,
ФБГОУ ВПО «ТГТУ»; sartemova@yandex.ru*

Ключевые слова и фразы: генетические алгоритмы; оптимизация структурного построения; программное обеспечение.

Аннотация: Рассмотрен подход решения задачи структурного построения программного обеспечения на примере информационно-измерительной системы (ИИС). Определена оптимальная структура размещения разработанных алгоритмов по используемым модулям ИИС. В качестве целевой использована суммарная функция внутренних связей системы, которая максимизирована. Чтобы проектируемая система обладала требуемыми свойствами, введены следующие ограничения: однократность включения алгоритма в совокупность программных модулей, дублирование занимаемой памяти информационным элементом, сложность информационного интерфейса между отдельными модулями. Решение задачи найдено с использованием генетических алгоритмов. Подобный подход применен для проектирования информационного и программного обеспечения как в информационно-измерительных, так и в информационно-управляющих системах.

Список литературы

1. Артемова, С. В. Решение задачи структурного построения программного обеспечения интеллектуального датчика влажности / С. В. Артемова // Програм. продукты и системы. – 2013. – № 3. – С. 272 – 276.
 2. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. – М. : Горячая линия – Телеком, 2009. – 608 с.
 3. Норенков, И. П. Основы автоматизированного проектирования : учеб. для вузов / И. П. Норенков. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. – 336 с.
-

Konstruktion der Software des intellektuellen Sensors der Feuchtigkeit

Zusammenfassung: Es ist das Herangehen der Lösung der Aufgabe der strukturellen Konstruktion der Software am Beispiel des Informationssystemsystems (IMS) betrachtet. Es ist erforderlich, die optimale Struktur der Unterbringung der entwickelten Algorithmen nach den verwendeten Modulen vom IMS, zu bestimmen.

Als zweckbestimmte Funktion wird die summarische Funktion der inneren Beziehungen des Systems, die maximiert wird, verwendet. Damit das entworfene System über die geforderten Eigenschaften verfügte, werden die folgenden Beschränkungen eingeführt: auf die Einmaligkeit des Einschlusses des Algorithmus in die Gesamtheit der Programmmodule, auf das Dublieren des eingenommenen Gedächtnisses vom informativen Element, auf die Komplexität des informativen Interfaces zwischen den abgesonderten Modulen. Die Lösung der Aufgabe befindet sich unter Ausnutzung der genetischen Algorithmen. Das ähnliche Herangehen kann man für die Projektierung der Informations- und Software, wie der Informationsmeß-, als auch Informationssteuersysteme verwenden.

Construction du logiciel du capteur intelligent de l'humidité

Résumé: Est examinée l'approche de la solution du problème de la construction structurelle du logiciel à l'exemple du système informationnel de mesure (SIM). Il est nécessaire de définir la structure optimale du placement des algorithmes suivant les modules SIM employés. Est utilisée la fonction sommaire des liaisons du système qui est maximisée. Sont introduites les limitations sur l'insertion des algorithmes dans l'ensemble des modules, sur le doublage de la mémoire par un élément informatique, sur la complexité de l'interface entre les modules. L'approche pourrait être utilisée pour la conception du logiciel des systèmes informationnels de mesure et ceux de commande.

Авторы: *Артемова Светлана Валерьевна* – кандидат технических наук, доцент кафедры «Конструирование радиоэлектронных и микропроцессорных систем»; *Подхватилин Павел Александрович* – студент; *Шелудяков Роман Олегович* – магистрант кафедры «Конструирование радиоэлектронных и микропроцессорных систем», ФГБОУ ВПО «ТГТУ»

Рецензент: *Муромцев Дмитрий Юрьевич* – доктор технических наук, профессор, заведующий кафедрой «Конструирование радиоэлектронных и микропроцессорных систем», ФГБОУ ВПО «ТГТУ».
