

# Algorithm for the integration of software modules based on the ontological approach

Khamza Eshankulov, Gavhar Turdiyeva, Makhsuma Ismoilova, Guli Murodova and Rano Murodova

Bukhara State University, 11, M. Iqbol Street, Bukhara, 200114, Uzbekistan  
vivente\_2006@mail.ru

**Abstract.** In this paper, an algorithm for integrating software modules in the creation of a software package for information monitoring and decision-making of continuous production stages has been developed. The software complex consists of several modules. Modules installed in the client-server hardware, such as authentication, reading data from a video surveillance device, reading data from electronic scales, data entry, integration, information monitoring, decision-making modules. The data recorded by integrating the data is stored in a database on the server. Each program acts as a module and transmits data to another module. An integration algorithm was developed using an ontological approach to integrate software modules. This builds an ontology for each software module. Classes are developed based on ontology, and application modules are integrated based on class attributes. Metadata is stored in each class and attribute. Through the core class, metadata is received and transmitted from the program modules.

**Keywords:** Information monitoring, Integration, Software modules, Ontologies, Architecture, Metadata structure, Decision-making.

## 1 Introduction

Today, the society is rapidly developing computerization and automation, the creation of software tools and complexes for information monitoring and decision-making. [1-2].

Until recently, the tasks entrusted to specialists were designed and implemented through hardware and software tools and software packages [3-4].

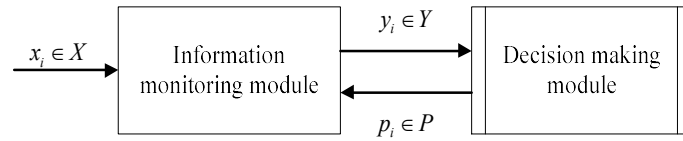
Lack and availability of information monitoring systems in manufacturing enterprises is also different, the requirements for information monitoring systems are different, bringing flexible software packages to the level of demand is becoming a problem [5-6].

---

<sup>1</sup> Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

When creating a software package for information monitoring and decision-making of continuous production stages of manufacturing enterprises, the problem of integrating software modules into one system arises.

In this case, it is necessary to monitor the information through the data sets  $X_i = \{x_1, x_2, \dots, x_n\} \in X$  at the stages of continuous production through computer systems installed on the basis of computer networks of the enterprise and to make decisions  $p_i \in P$  on the basis of monitoring parameters  $Y_i = \{y_1, y_2, \dots, y_{n1}\} \in Y$  from the stage of continuous production  $s_i \in S$  (Figure 1).



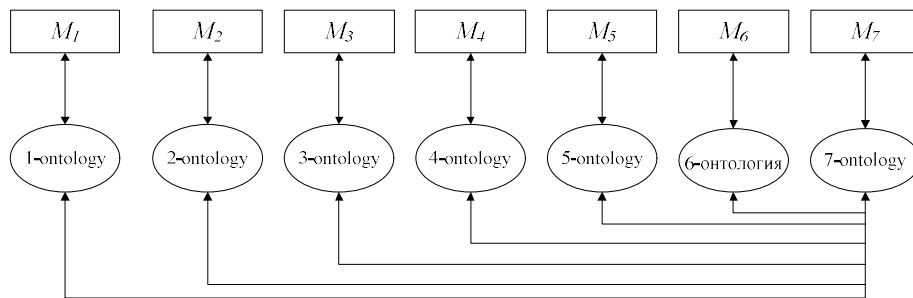
**Fig. 1.** Information monitoring and decision making scheme.

The software complex is built on the basis of client-server architecture [7]. In this case, the model of the information monitoring module works on the basis of an algorithm developed on the basis of the Petri network [8]. The decision-making module is created based on the frame knowledge base [9-10].

In order to create a complete software complex, the task was to implement the problem of integration of several software modules on the basis of an algorithm developed through an ontological approach [11-12].

## 2 Materials and methods

The integration scheme of the software modules through the ontological approach is presented in the form in Figure 2. It builds ontologies to store the metadata used in the integration of each module, collects data using the integration module, and stores it in a server database.



**Fig. 2.** Scheme of integration of information monitoring and decision-making software modules:  $M_1$  - authentication module;  $M_2$  - video reading module from video surveillance device;  $M_3$  - electronic balance data reading module;  $M_4$  - data input module;  $M_5$  - decision module;  $M_6$  - information monitoring module;  $M_7$  - integration module.

In the integration algorithm through the ontological approach, integration is done through the comparison of classes and the interrelationships between class attributes.

The ontology is formally expressed as follows:

$$O = \{\{T\}, \{R\}, \{F\}\}, \quad (1)$$

In this:

- $T$  - A set of concepts, terms, and classes based on them that are defined in ontology  $O$  for the subject area. For example, the weight of the oil seed, the date of acceptance, the state number of the machine, the type of oil seed;
- $R$  - a set of relationships between classes in a subject area. For example, the data exchange relationship between the data entry class and the electronic balance class;
- $F$  - a set of functions that connect many attributes and relationships that are considered in the ontology  $O$ . For example, procedures that read data from electronic scales and store them in the data structure.

In the integration process, the identification, attribute, and relationship of each class to other classes is described as follows:

$$T = \{\{I\}, \{A\}, \{P\}, \{C\}\}, \quad (2)$$

In this:  $I$  - class identifier;  $A$  - set of attributes;  $P$  - a set of senior classes;  $C$  - a set of lower-level terms.

The set of attributes is defined as follows:

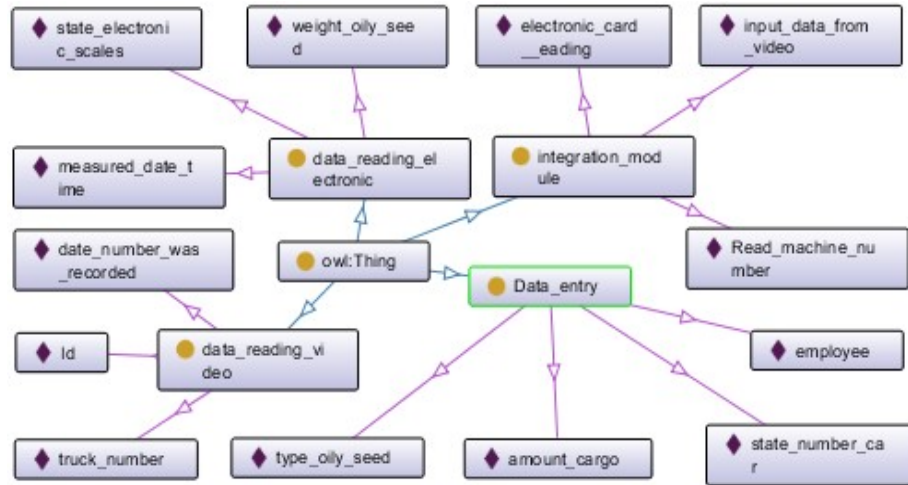
$$A = \{N, T_p, K, V\}, \quad (3)$$

Here:  $N$  - attribute name;  $T_p$  - attribute type;  $K$  - attribute key;  $V$  - attribute value.

A seven-layer architecture is proposed to integrate the modules of the software package. Using a seven-layer model is a convenient way to semantically link programming modules to each other. This model includes the following layers:

- Authentication module ontology;
- Ontology of data reading module from video surveillance device;
- Ontology of data reading module from electronic scales;
- Ontology of data input module;
- Ontology of information monitoring module;
- Decision module ontology;
- Integration module ontology;

In the ontologies developed for each integration module, metadata is stored and information is exchanged through the integration module. The ontology classes and attributes of the module are constructed to store metadata (Figure 3).



**Fig. 3.** Scheme of integration of software modules through an ontological approach.

### 3 Results and Discussion

Semantic metadata is used to describe ontological objects (Figure 4). Metadata is structured data that describes the content of data sources or databases.

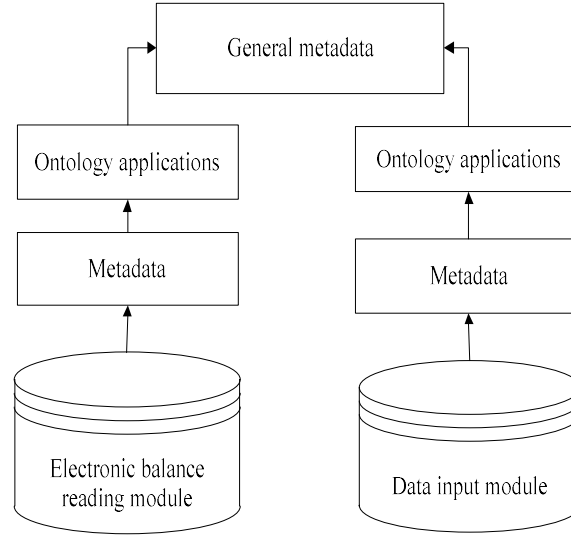
Formally, metadata can be expressed as follows:

$$M = \{T_i\} \cup \{Ex_i\}, \quad (4)$$

In this:  $T_i$  - many terms in ontology related to the system  $i$ ;  $Ex_i$  - a set of copies of terms in ontology  $O$ ;

There are three types of metadata in the software package:

- Systematic - service information required for the operation of information system functions;
- Structured - The information monitoring system is designed to describe general information about the location, name, size of objects;
- Semantic - used to describe the meaning of an object within a system and its relationship to other objects.



**Fig. 4.** Metadata structure.

The software package should not only have a structure, but also be expressed strictly on the basis of certain semantic rules. These rules are used to process and modify permissible data. Semantic relationships make it possible to integrate one module with another through permitted data.

The ontology of each module is represented by object-oriented programming language classes based on semantic relevance for the integration of information monitoring and decision-making modules, for example:

```

//class Readbumber {
String yukmashinaRaqam;
Int32 id;
DateTime raqam_qaydetilgan_vaqt }

```

The Readbumber class recognizes truck numbers based on the attributes id, number\_registered\_time, yukmashina\_Number, and saves them in an XML file.

In ontology  $O$  the semantic relationship is taken as the predicate defined in  $\overline{O}$ , and this is determined by the set of related  $Z = \{z^1, z^2, z^3\}$ .

If there is a semantic connection  $z$  in  $O$  ontology, then  $z(O)$  is represented in appearance.

The process of integrating data reading and data input modules from electronic scales is as follows:

$O_1 = \{\{T_1\}, \{R_1\}, \{F_1\}\}$  - ontology  $U_1$  of data reading module from electronic scales;

$O_2 = \{\{T_2\}, \{R_2\}, \{F_2\}\}$  - ontology  $U_2$  of data input module;

The connection between ontologies known from practice refers to the connection between the classes to which they belong, and this can be as follows:

- equivalence:  $z^1 : map(T_1) = T_2$ , if  $S(T_1, T_2) \geq b$ , where  $b$  is the default value of the semantic  $S(T_1, T_2)$  relationship,  $T_1$  is used to create a map of class  $O_2$  in ontology. This means that the class of integrated values in the first programming module also exists in the second ontology.
- generalization:  $z^2 : map(T_1) = T_2$ ,  $T_2 = \{T_{2i}\}$ , if  $q < S(T_1, T_{2i}) < b$ , where  $b$  is the default value of the semantic  $S(T_1, T_2)$  relationship,  $T_1$  is used to create a map of class  $O_2$  in ontology. This means that the class of integrated values in the first programming module also exists in the second ontology.
- identification:  $z^3 : map(T_1) = T_2$ ,  $T_1 = \{T_{1i}\}$ , if  $q < S(T_{1i}, T_2) < b$ , where  $b$  is the default value of the semantic  $S(T_1, T_2)$  relationship,  $T_1$  is used to create a map of class  $O_2$  in ontology. This means that the class of integrated values in the first programming module also exists in the second ontology.
- partially equivalent:  $z^4 : map(T_1) = T_2$ , if  $q < S(T_1, T_2) < b$ , where  $b$  is the default value of the semantic  $S(T_1, T_2)$  relationship,  $T_1$  is used to create a map of class  $O_2$  in ontology. This means that the class of integrated values in the first programming module also exists in the second ontology. It was found that the intersection of a set of attributes of classes  $T_1$  and  $T_2$  has common attributes ( $A^2 \cap A^1 \neq \emptyset$ ). There is a class  $T$  that is the superclass of  $T_1$  and  $T_2$ .
- differentiation:  $z^4 : map(T_1) = \emptyset$ ,  $\exists T_1, \forall T_2 \in O_2, S(T_1, T_2) \leq q$ ,  $q$  is the limiting value that is not equivalent.

The integration of data reading and data input modules from electronic scales can be expressed as follows:

$$S = \langle O_1, O_2, U_1, U_2, Z \rangle \quad (5)$$

Based on the above rules, the integration algorithm is proposed as follows: алгоритми қуйидагича таклиф қилинди:

*Step 1.*  $O_1$  ontology,  $T_1$  classes and  $A_1$  attributes are formed for  $U_1$  programming module.

*Step 2.*  $O_2$  ontology,  $T_2$  classes and  $A_2$  attributes are formed for  $U_2$  programming module.

*Step 3.* the semantic connection is determined by  $S(T_1, T_2)$ .

*Step 4.* If there is no semantic connection, the algorithm terminates, otherwise it moves to the next step.

*Step 5.* Integrates through semantic linking.

*Step 6.* The results are checked.

*Step 7.* The algorithm completes its work.

## 4 Conclusion

In this paper, ontologies are built for the integration of modules within the information monitoring and decision-making software package, and software modules are integrated at the data stage through an ontological approach. As an example, an algorithm for integrating modules for reading data from electronic scales, data entry, reading data from a video surveillance device has been developed.

## References

1. Peter Göhner, Michael Weyrich: Agent-based Concepts for Manufacturing Automation. Multiagent System Technologies: 12th German Conference, Germany, Proceedings, 77-92 (2014).
2. Global Industrial Automation Market: Overview. PRNewswire, 162 (2017)
3. Al-Ahmari A., Ridway K.: An integrated modelling method to support manufacturing systems analysis and design. Computers in Industry, 38(3), 225-238 (1999).
4. Shekhtman M.B.: Trends in the development of automation systems and new requirements for the training of automation specialists, Journal Industrial ACS and controllers, 4,. 10-14 (2008).
5. Romero D., Vernadat F.: Enterprise information systems state of the art: Past, present and future trends. Computers in Industry, 79, 3-13 (2016).
6. Kantorowitz Eliezer, Lyakas Alexander: Use-case components for interactive information systems. Science of Computer Programming, 56, 5-21 (2005).
7. Khamza Eshankulov, Hakim Eshonkulov: Architecture of information monitoring system of oil and fat producing enterprise. International Journal of Advanced Research in Science, Engineering and Technology, 7, 2, 13031-13036, (2020).
8. Muminov B.B., Eshankulov Khamza: Modelling Asynchronous Parallel Process with Petri Net. International Journal of Engineering and Advanced Technology (IJEAT), 8, 5S3, 400-405 (2019).
9. Muminov B. B., Eshankulov H. I.: Development of a decision-making system module (FSSM) and its frame model in the manual "Oil Monitoring". Tashkent, 254 (2019)
10. Ergashev A.A., Khusenov M.Z.: Using a frame model to describe knowledge. Bukhara State University of Scientific Information, 4, 92-95 (2019).
11. Albers Albert, Kurrle Armin: Using semantic metadata for continuous development of requirements and goals in the Smart Mobility domain. An empirical study. Procedia CIRP, 70, 168-173 (2018).
12. Lin H. K., Harding J. A.: A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. Computers in Industry, 58, 428-437 (2007).