EPJ Web of Conferences **248**, 02005 (2021) https://doi.org/10.1051/epjconf/202124802005
MNPS-2020

Information-Active Systems Design Tool: "IS-2" Integrated Environment

Andrey Gavrilov, *Galina* Volkova, and *Olga* Novoselova*

Department of information technologies and computing systems, Institute of information technology, Moscow State University of Technology "STANKIN", RU-127055, Moscow, Russia

Abstract. The article describes the tool for designing information-active systems - the integrated environment "IS-2": the concept of the methodology automation of intellectual labor (MAIL), the method of modeling the visual integrated environment, its structure, and features of the software solution. The Methodology Automation of Intellectual Labor (MAIL) offers an industrial way to create information-active, intelligent, automated and other systems. Their design involves the formation of models at three stages - initial modeling, conceptual modeling, and infological modeling. To automate the design process according to MAIL, a visual integrated environment (VIS) that supports this process is being developed at MSUT "STANKIN". A formal description of the structure of the models, the structure of creating models process, the method of modeling VIS has been completed. The features of the integrated environment "IS-2", obtained in the process of its development such as the formation of a universal (library of universal components used in the development of modules) component and specialized, are described. The composition of the versions is described including graphical editors for forming constituent models and their linking, means of intermediate documentation of models in the form of diagrams and specifications, and prototypes of modules for supporting analytical processing and synthesis of models.

1 Introduction

The methodology of the automation of intellectual labor (MAIL) defines an industrial way of creating various systems. This methodology involves dividing the creation process into stages such as initial modeling, conceptual modeling, infological modeling, datalogical modeling (logical and physical) and implementation. The implementation of any stage of the MAIL consists in the application of an invariant set of procedures for the formation of models at two levels of complexity: generalized, in the case, when the generated model describes a set of design tasks, and local, when the model describes just one design task. Since local and generalized models have a similar structure, their execution procedures are algorithmically identical [1].

Thus, in order to automate the design process of information-active, automated, informational and other systems in accordance with MAIL, a set of software tools is needed to support all the stages (initial, conceptual and infological modeling) of their implementation for any structure of each model. The approach to the automation of procedures by methodology used earlier made it possible to quickly obtain ready-to-use tools for supporting individual procedures at various stages of the design of information-active and automated systems in a short time. However, with the increase in their number, the development of complete projects containing models from different stages has become more difficult. This occurred due to the lack of interaction and information transfer between these software tools. In addition, for the developed tools, different implementation platforms were often used, which made it difficult to further debug and develop them.

To solve the problem of constructing a project of system (automated, informational, information-active, etc.), a visual integrated system/environent (VIS) is being developed, which will include the interface of support tools for various procedures of the methodology and an automated method for their formation such as visual modeling. To reduce the time of creation of both various VIS modules and the entire system, a method for modeling the environment has been developed, taking into account the structural features of models, their formation and execution [2].

2 The structure of information-active systems models

MAIL offers modeling methods for various stages of the life cycle of information-active or automated systems (AS). Each method contains a formal description of two model representations - universal and for subject problems. The model representation for subject tasks (ST) consists of models formed at two levels of abstraction - an object level model and a specific level model. Since the VIS is supposed to be used to form models of object-level subject tasks, its structure should be based on the structure of these models. [1-3]

* Corresponding author: olga.n10@yandex.ru

Since the structures of the models are identical, on their basis a unified model of the object level of the ***n***-th object problem of the ***i***-th stage of creation was compiled:

$$M2_i(n) =< EL2_i(n), ST2_i(n), DN2_i(n), FN2_i(n), R2_i^{SDF}(n) > \quad (1)$$

where $EL2_i(n)$ is the set of elements of the ST model, $ST2_i(n)$ is the set of static relations on the elements of the object-level model (static component), $DN2_i(n)$ is the set of dynamic constraints on the elements of the model (dynamic component), $FN2_i(n)$ is the set of functional relations on the elements of the model (functional component), $R2_i^{SDF}(n)$ is the linkage of the components. Each component includes two structures: the main and the derivative for the static component, a system of constraints on the elements of the main and derivative of static structures, and a system of functional relations on the elements of the main and derivative of static structures.

From the point of view of a graphical description of structures for object-level models of the ***n***-th subject task, all components can be formally represented by graphs: static, dynamic, functional and meaningful (component linking graph):

$$M2_i(n) \to \{G_i^{tr}(n)\}, \quad G_i^{tr}(n) = \left(V_i^{tr}(n), E_i^{tr}(n)\right), \quad (2)$$

where the indices ***t, r*** determine the type of the graph and its components:

$$t \in ('ST', 'DN', 'FN', 'R') \quad (3)$$

denote the type of the component represented by the graph, $r \in [1,2]$ determine the structure index.

From the point of view of the tabular description of structures, each model of object-level for the n-th subject task is fixed in the form of specifications, which can be formally represented by relational relations. Each structure of the model can contain specifications for describing structure elements, for describing binary and / or ternary relationships on structure elements.

$$M2_i(n) \to \{R_i^{tre}(n)\}, \quad (4)$$

$$R_i^{tre}(n) \subset D_i^{tre1} \times D_i^{tre2} \times \ldots \times D_i^{trem}, \quad (5)$$

$$R_i^{tre}(n) = \{r_i^{tren}: r_i^{tren} = \left(d_i^{tre1-x}, d_i^{tre2-y}, \ldots, d_i^{trem-z}\right)\}, \quad (6)$$

where the indices ***t, r*** determine the type of specification describing the ***tr***-th structure, the index ***e*** defines the index of the specification: ***e*** = 1 for a table with elements, ***e*** = 2 for a table with binary links, ***e*** = 3 for a table with ternary links, ***e***> 3 for additional specifications.

3 Process structure

The structure of creating models process can be formally presented as follows:

$$Y = (P, TP), P = \cup_u P_u, \; P_u \cap P_k = \emptyset, \; u \neq k, \quad (7)$$

$$P_u = \{p_{uv}\}, \quad (8)$$

$$TP \subset P \times P, \quad (9)$$

where *P* is a set of actions, *TP* is a set of links between actions that describe the structure of actions. P_u - a set of actions of the ***u***-th level of complexity, p_{uv} - the ***v***-th action at the ***u***-th level of complexity in general. In this case, actions at each difficulty level can be defined as follows:

$$\overline{p_{1\iota(s)}} = p_{1i}\&(p_{0s}, p_{1i}) \quad (10)$$

- ***i***-th stage of ***s***-th process,

$$\overline{p_{2J(s\iota)}} = p_{2j}\&\left(p_{1i}, p_{2j}\right)\&(p_{0s}, p_{1i}) \quad (11)$$

- ***j***-th procedure of the ***i***-th stage of the ***s***-th process,

$$\overline{p_{3k(s\iota J)}} = p_{3k}\&(p_{2j}, p_{3k})\&\left(p_{1i}, p_{2j}\right)\&(p_{0s}, p_{1i}) \quad (12)$$

- the ***k***-th operation of the ***j***-th procedure of the ***i***-th stage of the ***s***-th process,

$$\overline{p_{4l(s\iota Jk)}} = p_{4l}\&(p_{3k}, p_{4l})\&(p_{2j}, p_{3k})\&\left(p_{1i}, p_{2j}\right)\&(p_{0s}, p_{1i}) \quad (13)$$

- ***l***-th action of the ***k***-th operation of the ***j***-th procedure of the ***i***-th stage of the ***s***-th process.

4 Visual Integrated Environment (VIS) modeling method

The method of modeling a visual integrated environment consists in its sequential presentation at various levels of detail as a set of fundamentally different structural elements. [2]

Taking into account the structure of models of information-active systems and procedures for their processing, the visual integrated environment can be represented as a set of models of various levels of abstraction:

$$VIS =< VIS1, VIS2, VIS3, R^{123} >, \quad (14)$$

Where $R^{123} = \{R^{12}, R^{23}\}$ - is a set of interrelationships between elements of VIS models of different levels of abstraction. (see fig. 1)

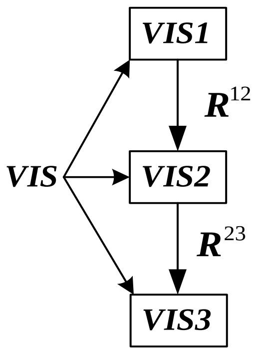

Fig. 1. Representation of the VIS implementation model taking into account the levels of abstraction

For each action $\overline{p_{4l(sijk)}}$, there can be many versions of the support tools implemented as functional modules

$$M^{sijkl} = \{m_f^{sijkl}\} \qquad (15)$$

Based on this, the visual integrated environment can be represented as a system of functional modules:

$$\mathrm{VIS1} =< \mathrm{M}, \mathrm{R}^1 >, \qquad (16)$$

$$\mathrm{M} = \bigcup_{s,i,j,k,l,f} \mathrm{M}^{sijkl} \qquad (17)$$

where ***M*** is a set of functional modules, $R1 \subset M \times M$ is a set of relationships between functional modules.

For a set of modules ***M*** that implement all processes, information, functional and interface components can be distinguished that implement the corresponding parts of the modules:

$$M^{sijkl} \rightarrow < \mathrm{INF}^{sijkl}, \mathrm{FUN}^{sijkl}, \mathrm{INT}^{sijkl}, \mathrm{R}^{sijkl} >, \qquad (18)$$

$$\mathrm{R2}^{sijkl} \subset \mathrm{INF}^{sijkl} \times \mathrm{FUN}^{sijkl} \times \mathrm{INT}^{sijkl}. \qquad (19)$$

Thus, at the second level of abstraction - the level of logical components, the visual integrated environment can be represented as a set of components and their relationships:

$$VIS2 =< INF, FUN, INT, R2 >. \qquad (20)$$

Each component at the file system level is represented by its elements of the corresponding types.

$$INF^{sijkl} =< TI^{sijkl}, DI^{sijkl}, FI^{sijkl}, R2^{sijkl} > \qquad (21)$$

for an information component, which is implemented by information volumes, directories, files. The set of files of the functional and interface components are formed in the same way.

Based on the foregoing, the VIS model at the third level of abstraction - the file system (physical), is formally defined as a set of volumes, directories and files:

$$VIS3 =< T, D, F, R3 >, \qquad (22)$$

$$\mathrm{R3} \subset \mathrm{T} \times \mathrm{D} \times \mathrm{F} \times \ldots \times \mathrm{T} \times \mathrm{D} \times \mathrm{F}, \qquad (23)$$

$$\mathrm{R3} = \mathrm{R3}_1 \bigcup \mathrm{R3}_2 \qquad (24)$$

At the same time, $R3_1$ determines the storage structure of the VIS, i.e. hierarchical connections between the elements of the file system, and $R3_2$ - the structure of functioning, i.e. the relationship between files within a specific version of the VIS.

The identical structure of the models at each stage, an invariant set of modeling procedures made it possible to identify invariant parts in the implementation of functional modules - universal files that implement identical information, functional or interface parts of the modules. For example, for information component files of various modules:

$$FI^{sijkl} \bigcap FI^{smjkl} = UFI^{jkl}, \qquad (25)$$

where UFI^{jkl} is the set of universal information files that implement the ***l***-th action of the ***j***-th procedure for the ***k***-th component of the model at various stages ***i*** and ***m***.

The set of such files that implement the universal part of the logical components of the VIS must be assembled into a separate library - a separate constructive element of the environment, which makes it possible to reduce the implementation time of its modules. The considered method for modeling VIS allows it to be represented in the matrix form shown in Fig. 2. Each area defines the class of the VIS module - its place within the entire system. The class of the module determines its functional purpose, input and output data. The colored sections show an example of module classes that can be built on generic components: red - modules that support a certain type of model, blue - support processing of a certain component, green - support a certain procedure. Data structures, processing algorithms and presentation forms can be invariant.

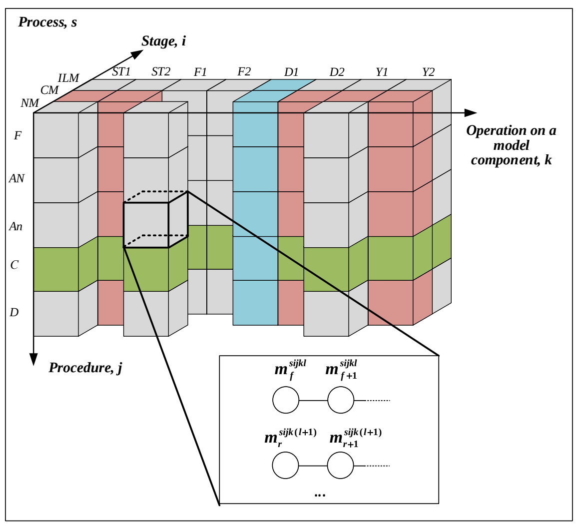

Fig 2. The structure of the visual integrated environment.

Symbols. Modeling procedures: F - formation, AN - analysis, An -analytical processing, C - synthesis, D - documentation. Stages: NM - initial modeling, CM - conceptual modeling, ILM - infological modeling; components of the models: ST1 - basic static structure, ST2 - derived static structure, F1 - functional structure of the first kind, F2 - functional structure of the second kind, D1 - dynamic structure of the first kind, D2 - dynamic structure of the second kind, Y1 - linking ST1, D1 and F1, Y2 - linking CT2, D2 and F2.

5 VIS implementation

When developing a software environment that ensures the integration of the means to support the procedures of the MAIL, the main task is to ensure information flows between the means that can operate simultaneously on different computers. It should also be borne in mind that a single modeling procedure can have multiple support tools, depending on the configuration the integrated environment itself. A schematic diagram of information interaction between classes of such tools within one stage of creating information-active and automated systems is shown in Fig. 3.

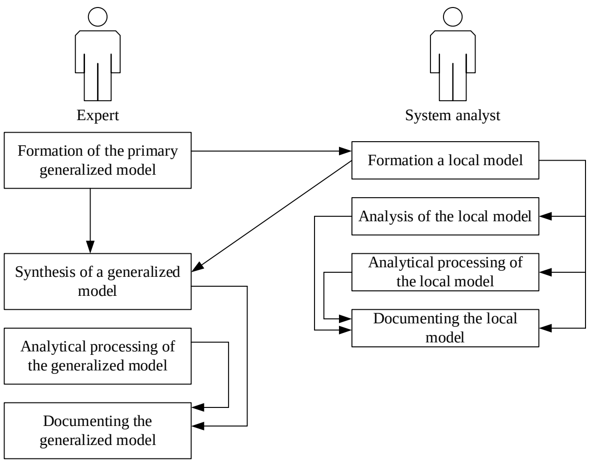

Fig 3. Information flows between classes of integrated environment modules

Based on this, a visual integrated environment (VIS) was developed in the form of the "IS-2" software package (see Fig. 4-5), which contains functional modules supporting the procedures of the MAIL stages. This software package "IS-2" provides a graphical method for developing models and automatic generation of specifications based on the constructed diagrams. The complex manages the operation of modules, as well as organizes their information interface. Since the models have a similar composition and are processed by an invariant set of procedures, during the development of the VIS, the unification of software solutions was made - the general information, functional and interface aspects of various modules were highlighted. Later they were moved to a single repository of universal components. [4, 5]

Created at the Department of Information Technologies and Computing Systems of the MSUT "STANKIN" implementation - the software complex" IS-2 "allows you to form models at various stages of system design (initial, conceptual, infological), to perform analytical processing of models, including restructuring the algorithm for solving the task. Also "IS-2" can form a generalized model for a complex of subject tasks, integrating algorithms for solving subject tasks into a single representation of the solution for the complex. [4, 6-9]

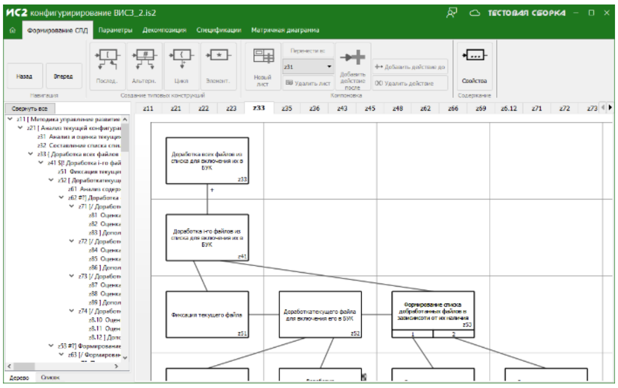

Fig. 4. Graphical interface of "IS-2": module for forming a system of subject actions

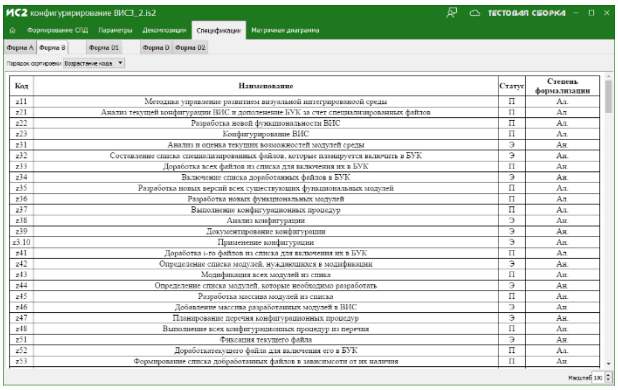

Fig. 5. Graphical interface of "IS-2": module for outputting specifications of the initial model

The integrated environment "IS-2" consists of two subsystems - a specialized one, which includes many modules of the environment, and a universal one, presented in the form of a library of universal components (LUC) used in the development of modules. LUC represents software tools for developers of "IS-2" modules, namely: data structures describing models as graph structures, software libraries containing the main processing algorithms for these structures, which are associated with the implementation of functional modeling procedures, as well as with various forms of information presentation reflected in the models. The library also includes tools for converting invariant structures to models of a certain type for further work with them in the environment modules. Thus, the design and development of modules is greatly simplified due to the repeated use of already implemented software structures.

A feature of the IS-2 is the presence of several modes of its operation. The main operating mode of the complex is the development of information-active and automated systems, their stage-by-stage modeling and subsequent implementation. But, since the environment itself is a complex information-active system, it can be used to develop new versions of itself.

6 Conclusion

At the present time, the "IS-2" versions are developed and used in the educational process and scientific activities of the Department of Information Technologies and Computing Systems of the MSUT "STANKIN" to support the initial and conceptual modeling. Each version contains graphic editors for forming components models and their linkage, means of intermediate documentation of models in the form of diagrams and specifications. Also, prototypes of modules supporting analytical processing and model synthesis were developed and included in the environment. The development of modules for conjugation of various adjacent stages of modeling and modules for supporting infological modeling is in the final stage [6, 10-12].

It should be noted that the "IS-2" includes a SDK (a Software Development Kit), a set of software tools used by third-party developers to create their own modules for the environment.

This article was prepared in the framework of the project No. 17-29-07057 "Development of methods of restructuring and integration for semantic and syntactic representations when creating automation systems for design and management processes" under the RFBR grant. This work was carried out using equipment provided by the Center of Collective Use of MSUT "STANKIN".

References

1. G.D.Volkova, *Methodology of intellectual labor automation. Monograph* (M.: Janus-K, 2013)
2. G.D. Volkova, A.G. Gavrilov, Vestnik MSUT "STANKIN", **4**(39), 85-89 (2016) [in Russian]
3. O.V. Novoselova, G.D. Volkova, A.G. Gavrilov, Proceedings of higher educational institutions. Volga region. Technical science, **1**(29), 81-91 (2014) [in Russian]
4. A.G. Gavrilov, G.D. Volkova, O.V. Novoselova, Scientific and technical journal "Information technologies in design and production", **2**, 30-36 (2018)
5. A.G. Gavrilov, G.D. Volkova, O.V. Novoselova, Digital economy: technology, management, human capital: materials of the III All-Russian scientific and practical conference, 10-13 (Vologda: LLC "Marker", 2020) [in Russian]
6. A.G. Gavrilov, O.V. Novoselova, G.D. Volkova, Vestnik MSUT "STANKIN". Scientific peer-reviewed journal, **2**(41), 83-88 (2017) [in Russian]
7. O.V. Novoselova, A.A. Ruzheynikov, Science today: challenges and prospects: materials of the international scientific and practical conference, 72-74 (Vologda: LLC "Marker", 2020) [in Russian]
8. O. Novoselova, A. Ruzheynikov, A. Gavrilov, EPJ. Web Conf., **224**, 06008 (2019) https://doi.org/10.1051/epjconf/201922406008
9. O. Novoselova, A. Sidorov, IOP Conf. Ser.: Mater. Sci. Eng., **709**, 044014 (2020) https://doi.org/10.1088/1757-899X/709/4/044014
10. A.A. Ruzheynikov, Materials of the student scientific and practical conference "Automation and Information Technologies (AIT-2019)", 105 (Moscow: MSUT "STANKIN", 2019) [in Russian]
11. A.A. Ruzheynikov, Information technologies in modeling and management: approaches, methods, solutions: Collection of scientific articles of the II All-Russian scientific conference with international participation, **1**, 546-552 (Togliatti: Publisher Kachalin Alexander Vasilievich, 2019) [in Russian]
12. I.Yu. Pavlov, Materials of the student scientific and practical conference "Automation and Information Technologies (AIT-2020)", **3**, 148-150 (Moscow: FGBOU VO "MSUT" STANKIN ", 2020) [in Russian]