Life Cycle and Tools for Designing and Implementing a Typical Scenario for the Operation of an Automated System

Valeriy P. Khranilov, Pavel V. Misevich, Elena N. Pankratova and Pavel S. Kulyasov

November 7, 2024

Life Cycle and Tools for Designing and Implementing a Typical Scenario for the Operation of an Automated System

V.P.Khranilov
Computer technologies in designing and production dept.
Nizhny Novgorod State Technical University
Nizhny Novgorod, Russian Federation
hranilov@nntu.ru

P.V.Misevich
Computational systems and technologies dept.
Nizhny Novgorod State Technical University
Nizhny Novgorod, Russian Federation
p_misevich@mail.ru

E.N.Pankratova
Foreign languages dept.
Nizhny Novgorod State Technical University
Nizhny Novgorod, Russian Federation
keibusan@gmail.com

P.S.Kulyasov
Computational systems and technologies dept.
Nizhny Novgorod State Technical University
Nizhny Novgorod, Russian Federation
p.kulyasov@gmail.com

Abstract**—The concept of the life cycle will be topical in the foreseeable future. The main reason for this forecast is the increase of the complexity level of software and hardware complexes of automated systems. As a rule, the life cycle concept is applied to all subsystems which are involved in the information support, software and hardware. This approach covers almost all elements of automated systems. The fact leads to the high level of complexity of CALS systems. In addition to the existing approaches and models the paper proposes to apply the ideas of the life cycle support to the scenario of the automated system operating. It is obvious, that the success of the hardware and software complex mainly depends on the possibility of creating a scenario for solving problems in the subject area which is supported by automated system.**

Keywords—design, scenario, life cycle support, data exchange correctness control.

I. Introduction

The article introduces the concept of the life cycle of a scenario for solving a typical task in the subject area of automation. The life cycle consists of a set of stages. They are the following:

- the preparatory stage;
- decomposition of the method of solving a typical problem into a sequence of steps – events and connections between them;
- conceptual description of events;
- developing of information support and software for events of the scenario;
- script generation;
- implementation of the automated system scenario;
- monitoring of the process of executing a typical scenario by the system and accumulation of statistics;
- modification of the scenario during the life cycle of the automated system.

The first task for the scenario lifecycle of the support system is generating a scenario from a set of events. The constructor toolkit supports the process of designing a scenario from the elements of the scenario-generating set of events. The model of the elements for developing the scenario is described in the article. The examples of the designed scenarios are described in the paper too.

The second task is choosing an event from a set of alternatives. The approach for solving the problem using a dynamic model of the interactive process is described in the article.

The third task is checking the data flows between the events in the typical scenario. If the datum flows are correct, then each datum is “at the right time in the right place in the computing system”. This is the basic idea of the logistics principle in IT systems.

The conceptual model of data for the logistics control, the algorithm of checking of the data flows with an example are described in the paper.

II. Analysis

Engineers use the concept of the technical system life cycle for designing software and hardware complexes. A technical system is usually considered to be an object. The object is supported during the life cycle. A lot of attention is paid to this use of the life cycle concept in the research paper [1].

The advantage of this approach is the full implementation of the ideas of the system method. The object is decomposed into subsystems and relations among them. The support of the subsystems and relations among these subsystems is carried out during the life cycle.

This approach has an obvious drawback. It’s a high level of complexity. Modern software and hardware complexes integrate a large number of subsystems and relations among them. Thus, the CALS system for supporting software and hardware complexes becomes an object of high level complexity. It is the reason why the CALS system distances itself from the systems – objects of support.

The paper proposes using the life cycle concept for the operation scenario in an automated system. The proposed method [2-3] is an alternative to the classical approach [1]. The scenario of an automated system operation is a set of events and cause-and-effect relationships among the elements. The scenario is represented, as a rule, by the models of a graph or adjacency and incidence matrices. The set of events maps in the set of vertices of the graph. The set of relations maps in the set of the graph edges.

Today the operation of most commercial automated systems is based on a scenario model or its modifications (a block diagram, a flow chart, EPC diagrams, etc.). The success of the scenario implementation in an automated system is a very important element of the efficiency of the hardware and software complex as a whole. Thus, it is quiet logical to support the operation scenarios in an automated system during the life cycle.

Note, that the scenario of the automated system is a model, an abstraction. Thus, the support of the operation script during its life cycle is qualitatively different from the support of the hardware elements or software modules in an automated system.

III. Methodology

Definition. The life cycle of the scenario consists of a set of stages. They are the following:

- designing a structure of events for solving a problem in the subject area of operation the automated system;
- supporting the scenario during the operation of the system;
- migration of the automated system to other platforms or integration with other systems.

The life cycle scenario does not have the stages of disposal because a qualitatively designed automated system is in constant development. It adapts to changes in the external/internal environment and has a phase of transformation into a new quality of IT product instead of a recycling phase.

Definition. A typical task in the subject area is a task that is reproduced, formalized by models, and is subject to automation for solving with the use of a software and hardware complex.

Usually the typical task has several solutions that have to be supported by an automated system.

Definition. A typical scenario of the automated system operation is a scenario for supporting the solution of the typical task in the subject area of the hardware and software complex operation.

The typical scenario has a set of phases during its life cycle. They are the following:

1. Preparatory stage. It contains research into a subject area, formalization of typical tasks, setting associative links between them, formation of ways to solve a typical task.
2. Decomposition of the typical task solution into a set of steps/events and setting the relations among them.
3. Conceptual descriptions of events. It consists of functional content, input/output data, constraints and other useful technical information.
4. Developing of software and information support for the events of the scenario.
5. Scenario generation. The stage includes setting cause-and-effect relationships in the event structure.
6. Implementation of the script using the interpreter of the automated system.
7. Monitoring the process of executing a typical scenario in the automated system and accumulating statistics.
8. The modification of a typical scenario for supporting during the life cycle in the automated system.

When a developer designs the information support of an event, it is necessary to create a conceptual model of the object. The conceptual model is a machine-independent model for describing an event, which has to allow using the event in several scenarios of the operation in the automated system and in alternative software and hardware complexes (in the future). In the paper the conceptual model is referred to as an event passport. The event passport keeps the following information.

- the list of tasks that the event solves in the subject area;
- the list of identifiers of the input data of the event;
- the list of identifiers of the output data of the event;
- the description of the procedures for generating the values of the event outcome flag (the identifier of the causal relationship for the initialization of the next scenario event in a scenario);
- the restrictions on its use;
- the list of procedure identifiers for verifying the correctness of the input data.

The event passport stores a collection of data to support the set of operations. They are the following: the detection of incorrectly formed input data for an event of the script; the identification of problematic events in the scenario structure; sorting out problematic (emergency) situations.

The conceptual model of the event passport is presented in Fig.1. The entity is filled in gray in Fig 1. Event attributes are represented by white rectangles. The event passport is the starting point for creating a logistic algorithm for verifying correctness of a scenario.

IV. Constructor Model

Let us consider the constructor model. It is used for describing the process of designing the scenario (stage 5 of the scenario life cycle). The event in the graphical interpretation is represented by the vertex of the graph. The causal relationship of the scenario is represented by the edge of the graph (Fig.2). The constructor model supports the development of various script structures using the scenario-generating set of events *E*. Each causal relationship corresponds to an edge with an identifier. The edge has a terminal. It is indicated by an arrow in the Fig.2. An event (an element of the set E) can be entered into the terminal. Thus, the developer constructs various scenarios for the operation in an automated system from the elements of the scenario-generating set E. Examples of scenario event structures are shown in Fig. 3-4.

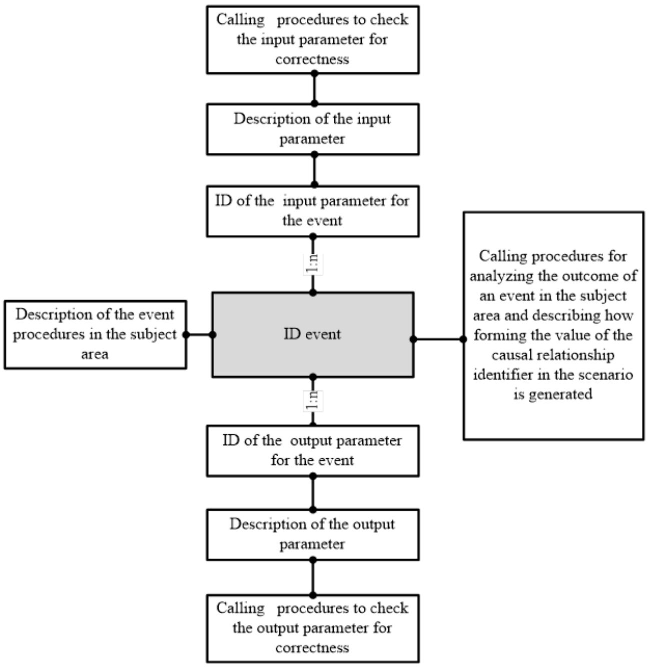

Fig. 1. Conceptual model (E-R diagram) the passport of an event.

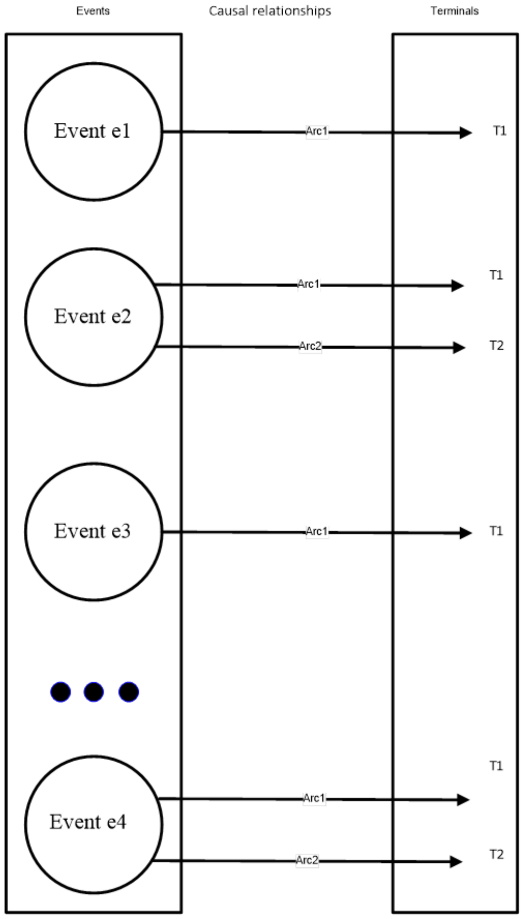

Fig. 2. Elements of the scenario-generating set of events *E* (Events) in the Constructor model.

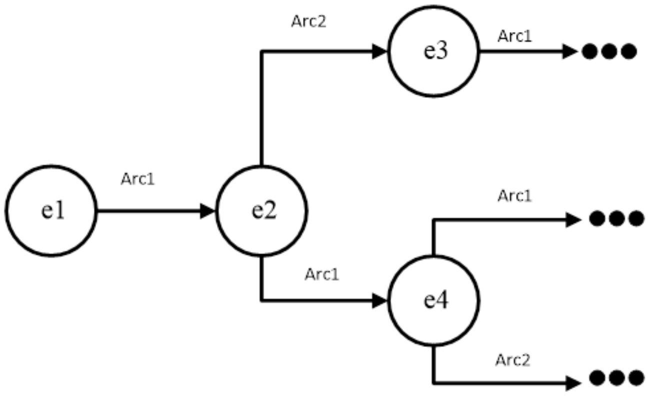

Fig. 3. Scenario 1 is constructed from elements of the set *E*.

Developing event structures from elements of the set *E* involves two technical problems. The first one is the task of choosing an event which has to enter a free terminal.

It is the task of selecting an element from a variety of alternatives. The reason for the appearance of this task is a situation that is reproduced in the set of events E, where there are various events that have to solve the problem of the event in the subject area in different ways. For example, the task of finding a local minimum in the range of acceptable values for a nonlinear function has to be solved by the following methods: full enumeration, gradient descent, the steepest descent method, and others.

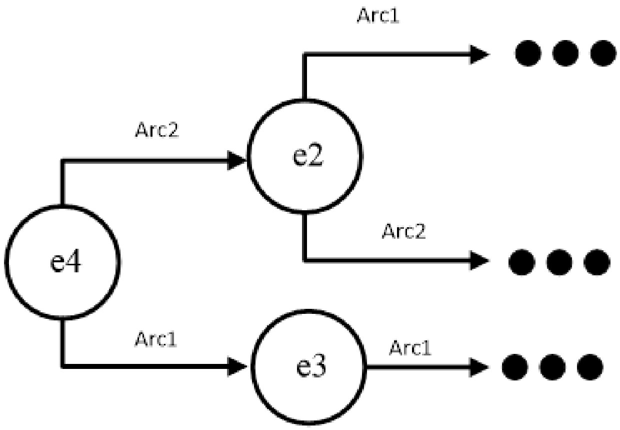

Fig. 4. Scenario 2 is constructed from elements of the set *E*.

Thus, at the stage of script developing (stage 5 of the scenario life cycle) the developers need to have the tools for supporting the procedure of selection the necessary element from the set of alternative events. In this paper it is proposed to use the mathematical model of interactive resource management in the field of technical systems and objects [3].

V. Tools for Controlling Information Flows

The second difficulty in the process of designing scenarios from elements of the set*E* is the appearing of logistical problems. The logistical problems cause emergency situations.

The definition. An emergency situation takes place when incorrectly formatted input data are supplied to the input of the initiated e_i event, or when the data in the hardware and software complex are not formed.

The reason for this situation is usually the following: when a developer designs a scenario using the cconstructor model, she or he has a direct access to all system resources without restrictions. An inexperienced developer runs a risk to develop a scenario with the problems which have been described above.

Obviously, if a designer intendsto create an accident-free scenario, it is necessary to equip the designer with tools for identifying and correcting logistical problems. The principle of logistics in the IT subject area [4] is formulated as follows: "Every required and correctly formed parameter must be at the required time and in the necessary address in computer memory or database". Using the approach we have to consider the scenario of solving a problem in the subject area as a scenario which has to be supported by the logistics tool. Each event of the script uses input data and generates output data. The identifiers of the input and output parameters are stored in the event passport (Fig. 1).

If during the process of solving a typical task in the subject area, all the input and output data of events are kept in the system, the tools in the event passport provide an opportunity to develop a fairly simple and effective algorithm for logistics control. The algorithm checks the validity of the inclusion of an event in the scenario being developed.

The process of designing a scenario using the constructor model is carried out sequentially, step by step. Each step (iteration) consists of selecting an event-candidate from the set of events. Evidently, the scenario of solving typical tasks in the subject area has to have the starting event. The event should be marked.

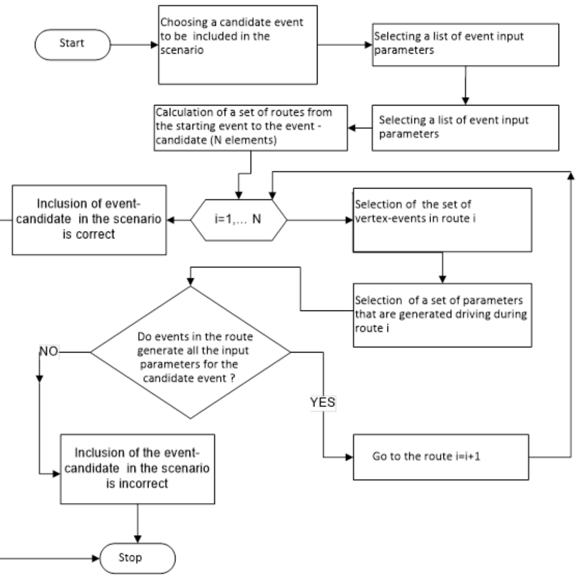

Fig. 5. The algorithm of checking the process of designing a scenario from elements of the set *E* (Events).

In general, there are several routes from the starting event to the stopping event in the scenario of solving a typical task. The algorithm for controlling the logistical correctness of the accession of the event-candidate in the scenario is based on the following steps:

- detection of all routes from the start event to the event-candidate;
- formation of a list of parameters which are generated in the hardware and software complex for every route. For the operation one has to use the list of events in the route and the set of event passports;
- comparison of the list of parameters which are generated in the system during the route (or routes) with the list of input parameters of the event-candidate.

An algorithm for checking the correctness of including the event-candidate in the designed scenario is shown in Fig. 5.

VI. Example

Let us consider a scenario for solving the task of synthesis of an optimal regulator for a discrete linear system with a quadratic quality criterion (1)-(2) (Fig.6). The user of the system has to develop an operation scenario for solving the task using the set *E* of. One of the scenarios for solving the tasks is presented in Fig.6. The mathematical model of the projected object is represented by the system (1).

$$x_{k+1} = A \times x_k + B \times u_k + C \times \varepsilon_k, \qquad (1)$$
$$y_{k+1} = s_{k+1} \times H \times x_{k+1} + \eta_{k+1}$$

$$J = \sum_{k=0}^{N-1} \left(x^T \times \Phi \times x + u^T \times \psi \times u \right) \tag{2}$$

where x_k is the n-dimensional state vector; u_k is the m-dimensional control vector; ε_k is the p-dimensional random noise vector in the object; y_k is the l-dimensional measurement vector; η_k is the l-dimensional noise vector in the measuring system; *A*, *B*, *C*, *H* are matrices of state, control, perturbation, meter with dimensions n∗n, n*m, n*p, n*l, respectively; the matrices *F* and ψ of the weighting coefficients have dimensions n*n and m*m;*N* is the number of steps of the transition process; the noises ε_k and η_k are described by discrete random functions R_k, Q_k, k=0,1, ... It is required to calculate the values of the regulator *G* according to the algorithm [5].

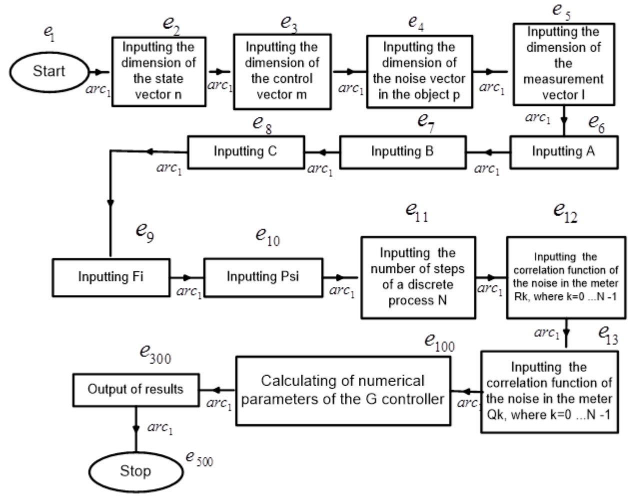

Fig. 6. Scenario with logistical problem in the structure of events.

The designer uses the dialog system constructor for solving the task of synthesis of the optimal regulator for the system (1) with the quality criterion (2). The solution scenario is presented in Fig. 6. A mistake was made when the scenario was being designed.

Let us consider how this error is detected and corrected. The algorithm for checking the correctness of the inclusion of the e_k– event in the scenario createsa list (3) of data which are generated in the CAD system during the process of operating from the start event to the event-candidate (e_{100}). The algorithm finds routes from the starting event (e_1 is the beginning event) to the event-candidate (e_{100}– event of synthesis of the parameters for the *G* controller). In the example there is only one route which connects the events:e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8,e_9,e_{10},e_{11},e_{12},e_{13}. Then, the algorithm generates a set of parameters which are generated during this routine in the system:

$$\{n, m, p, I, A, B, C, F_i, P_{si}, N, R_k, Q_k\} \tag{3}$$

The passport of the event (e_{100}) contains the list (4) of parameters which have to be generated in the system before initializing the event.

$$\{n, m, p, I, A, B, C, H, F_i, P_{si}, N, R_k, Q_k\} \tag{4}$$

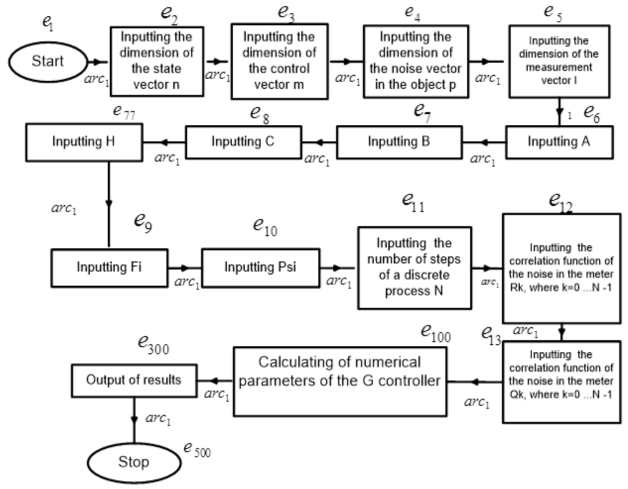

Fig. 7. Scenario after correction of the logistical problem.

Comparison of sets (3)-(4) shows that for the correct initialization of the event-candidate (e_{100}), it is necessary to add the event of forming the values of the missing parameter – the matrix *H* (an event with an identifier e_{77}).

VII. Conclusion

The article discusses the concept of the life cycle in relation to the scenario for solving a typical task in an automated system subject area. It is shown that scenario support is a causal factor of the success of using an automated system in its subject area. This approach differs from the classical one because the traditional concept of the life cycle is applied to software and hardware complexes. The article pays attention to the scenario design tools. The tools are based on the model constructor. Tools are used for supporting the process of developing a scenario from elements of a scenario generating set of events E. An approach to choosing an event from a set of alternatives is described. The application of the scenario support concept during its life cycle allowed to formulate a logistic algorithm for checking the correctness of the structure of events. An example of the application of this algorithm is considered in this paper.

The production model of the operation scenarios in automated systemsis used to solve a set of tasks. They are the following:

- developing dynamic models [6-7] of automated system operations [2];

- designing tools for implementing the logistics approach in automated systems [4];

- developing tools for building automated systems of situational type [8] (a subsystem for implementation ofscripts);

-designing systems of objects for assembling software [9-10].

References

[1] V. A. Anokhin, V. V. Anokhin, “Organization of production at an industrial enterprise: a textbook for universities. Book 2. Comprehensive automation of production processes” Moscow: TsNIIatominform, 2007.

[2] V. P. Khranilov, P. V. Misevich, E. N. Pankratova, and A. E. Ermilov, “Models for supporting the operating scenarios during

a life cycle in automated systems," 14TH International Symposium «Intelligent Systems», INTELS'20, 2020.

[3] V. P. Khranilov, P. V. Misevich, E. N. Pankratova, P. S. Kulyasov, "The Intelligent Systems to Support the Automated Systems During Life Cycle," 4nd International Conference on Control Systems, Mathematical Modeling Automation and Energy Efficiency (SUMMA), 2022.

[4] P. V. Misevich, "Using a logistics approach to issues building procedures for identifying and overcoming emergency situations in automated systems" Management Systems and Information Technologies, No 4.2(26), 2006.

[5] V. V. Kondratiev, A. A. Kocheshkov, "Analysis and synthesis of stochastic automatic control system with measurement control: Application package," NSTU, Moscow, 1984, reg. № P007336.

[6] R E. Kalman, P. L. Falb, M. A. Arbib, "Topics in mathematical system theory," McGraw Hill Book Co., 1969.

[7] R. E. Kalman, "Mathematical description of linear dynamical systems," Journal of the Society for Industrial and Applied Mathematics, No 1(2), 1963, p. 152-192.

[8] A. E. Ermilov, "Monitoring systems development using situational approach and fuzzy logic," IOP Conf. Ser.: Mater. Sci. Eng., 695 012016, 2019.

[9] P. S. Kulyasov, D. V. Zhevnerchuk, D. S. Martynov, and E. V. Tsybul'skaya, "Models and algorithms for assembling multicomponent systems based on algebraic structures," III International Conference on Advanced Technologies in Material Science, Mechanical and Automation Engineering (MIP: Engineering-III-2021). AIP Conference Proceedings 2402, 070032. Melville, New York, United States of America, 2021.

[10] D. V. Zhevnerchuk, L. S. Lomakina, A. S. Surkova, "Automation of a multicomponent assembly of open information systems," III International Conference on Advanced Technologies in Materials Science, Mechanical and Automation Engineering MIP: Engineering-III -2021.