



Software and Methodology for the Design of System Dynamics Models Based on the Situation-Activity Approach

Aleksey Sorokin^(✉) , Elena Brazhnikova, and Liliya Zheleznyak

MIREA – Russian Technological University, Vernadsky Avenue 78, 119454 Moscow, Russia
sorokin_a@mirea.ru

Abstract. An original approach to the system dynamics models' design as a projection of situation-activity analysis, expanding the possibilities of predictive-analytical research, is proposed for consideration. System dynamics models in this aspect are applied to human reasoning with the reference to the specific objects of activity. This makes it possible to establish the specified criteria for assessing the state and trajectory of the development of the system being modeled. This article proves the hypothesis that methods of knowledge representation in situational, expert and system dynamics models are similar. At the same time, the conceptual structures of situation-activity analysis incorporate the tools inherent in the system dynamics language. Conceptual structures are the core of the design of system dynamics models in notations: levels and flows. The result of the situation-activity design is a model represented in a conceptual language of knowledge representation, where the basic element is the act of doing. The conceptual structures of activity acts can be used to build production rules for the expert system. However, it is possible to extract partial representations from the conceptual structures of activity acts: process plans and regularities. On the aggregate of these plans, a graphical image for building system dynamics models is implemented. The knowledge of the fact that constructing conceptual structures is very challenging sparked off different searches for the best way to deal with this problem. Eventually the best solution has become a software toolkit "Designer + Solver + Interpreter which allows one to visualize conceptual structures, implement knowledge bases for expert and system dynamics models, as well as to conduct research on completeness and adequacy of the model.

Keywords: System dynamics model · Conceptual structure · Situation-activity approach

1 Introduction

Each and every one activity that is a part of a dynamically complex environment has a number of properties such as activity and the freedom to choose its state. In addition to the freedom of choice, the elements of activity (acts) have their own requirements for the product, i.e. the choice of state is made purposefully. This creates a situation where the

controlled objects seek to select their states that are the best in terms of their preferences for given values of the control actions. The control actions, in turn, depend on the states of the controlled objects. These conditions create uncertainty in the dynamics of the process due to the complexity of the environment. This inevitably leads to a simplified perception of reality and the manager gets emotional while solving a problem [1, 2].

It is known that problematic situations in decision-making are resolved by the following algorithm: if it is impossible to identify certain patterns in a system object behavior, then a hypothesis of its behavior is put forward. On its basis, simulation models are created. They help to investigate possible solutions.

Simulation models are subdivided into several types: system dynamics model, discrete-event model, and agent-based model. System dynamics models operate with time-continuous processes (states), while the discrete-event model and the agent-based model operate with discrete states. Transitions from one state to another in discrete modeling are considered instantaneous. Meanwhile, the important part of these two types of processes is that they can be presented in real systems, for example, the model of the development of an infectious disease. The appearance of a group of infected people in an agglomeration is considered as a discrete event, and the dynamics of the epidemic's development is described by a set of differential equations, i.e. is continuous [3, 4].

The discrete event model is constructed from standard functional blocks, defined by the average level of abstraction and implemented on the methodology of the structural approach SADT (structured analysis and design technique) or IDEF0 (ICAM Definition). Agent modeling supports all levels of abstraction and is defined by the Unified Modeling Language (UML) state and activity diagrams.

There is a direction in imitation modeling that maintains a high level of abstraction – system dynamics. In dynamic modeling, processes occurring in the real world are represented in terms of levels, the flows between these levels and the information determining the magnitude of these flows. The formal basis for this type of modeling is the flow rate equations that use the concept of dynamic processes in the state space. An important concept in establishing the structure of system dynamics is the idea that all changes are caused by “feedback loops”. Therefore, specialists in the field of simulation modeling have developed a toolkit for cause-effect diagrams taking into account the feedback effect. This graphical representation is essentially a mental map reflecting the relation between the individual elements of the system, as between cause and effect. The connection can be positive, when a change in the cause causes a similar change in the effect, and negative – a change in the cause entails the opposite change in the effect. Therefore, the success of the model directly depends on the correct understanding of the role of cause-and-effect relationships, which are difficult to determine without conceptual research [5].

It must be noted that the methods of knowledge representation in situational and expert systems are similar [6, 7]. At the same time, simulation models are actively used in situational modeling, therefore, a situational design language should include some tools inherent in modeling languages [8].

2 Materials and Methods

The methodology for constructing system dynamics models includes qualitative and quantitative stages. The qualitative stage examines the structure of the problem and the way one element of the system depends on another. At the quantitative stage, in the course of a computer simulation, the researcher determines how correct his model is and tests his hypotheses about the system behavior.

The basic construct of the qualitative stage is the representation of the process under study in the form of a diagram consisting of positive loops (a change in the cause causes a similar change in the effect) and negative feedback (a change in the cause causes the opposite change in the effect). The polarity of the bond is indicated by the “+” or “-” sign next to the corresponding arrow (see Fig. 1).

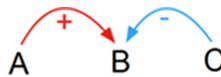


Fig. 1. Communication polarity designation.

At the same time, two serially connected negative links ultimately form a positive link. Causal relationships can form closed unidirectional loops, that is, positive or negative feedback loops. For example, in loop 1 of positive feedback (the polarity of the loop is indicated by a “+” in brackets within the loop), a growth in X results in a reduction in Y which results in an increase in Z, since the $Y \rightarrow Z$ relationship is negative, and an increase in Z causes a further increase X (see Fig. 2). In circuit 2, an increase in A causes a decrease in B. A decrease in B causes a decrease in C, since they are positively linked, in which a decrease in cause B entails a similar change in effect C [5].

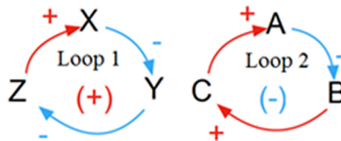


Fig. 2. Positive and negative feedback loops.

The following rules for determining the polarity of the feedback loops can be defined [3]:

- if the contour includes an even number of negative cause-and-effect relationships or there are none at all in it, this is a positive feedback loop;
- if the contour includes an odd number of negative cause-and-effect relationships, this is called a negative feedback loop.

At the same time, the polarity of the cause-and-effect relationship between the two elements of the contour is determined by the reaction of the effect element to a change

in the cause element, regardless of their connections with other elements. When constructing a model of a dynamic environment, a number of difficulties arise in using the feedback method:

- in complex systems, there can be several feedback loops of parameters, both positive and negative, and the causes for changes in the system states can be far from the effects, both in time and in space. Obviously, identifying such effects in a long chain of cause-and-effect relationships is impossible;
- understanding of the intuitive processes during constructing a feedback diagram is an important and non-trivial task. As a rule, only those dependencies between cause and effect are determined that are obvious, understandable and clear;
- correction of feedbacks, if they inadequately reflect reality, is an equally important task. The change in the mental models of a group of decision-makers appears to be even more complicated task.

On the basis of the diagram of cause-effect relationships of the modeled system, a diagram of flows and levels is built. It determines the quantitative stage of building a system dynamics model. Quantitative stage in its basic structure consists of four essential elements: information and material flows, decision function and level (see Fig. 3).

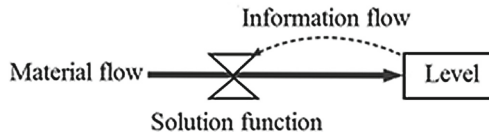


Fig. 3. Basic structure of the quantitative stage.

The levels are represented by quantitative indicators of the object under study and characterize the emerging accumulations within the system. They represent the values of the variables at the moment that they have as a result of accumulation due to the difference between the incoming and outgoing flows. The content of the levels can be of any nature [3].

The material flow, flowing into or out of the level, determines the change in the level. Thus, the value at each moment of time is characterized by a level described by the finite-difference Eq. (1) [5]:

$$X(t + h) = X(t) + (h \times V(t)) \tag{1}$$

Where h is change (increment) of time, t is model time, simulation step; $X(t)$, $X(t + h)$ is a level value at time points; $V(t)$ is a rate (pace) of level change, i.e. the magnitude of its change per unit of time.

$$V(t) = Vs(t) - Vd(t) \tag{2}$$

$Vs(t)$ is an incoming stream (source); $Vd(t)$ is an outflow (drain).

In addition, a distinction is made between information flows, which helps to make a decision on control actions, i.e. the value of the flow rate is determined for the next time interval. Thus, a decision function is defined, which is a formulation of a behavior line that determines how the available information about the levels leads to the choice of solutions related to the values of the current rates. The rate change law is set by functional dependence (3) [5]:

$$V(t) = F(p_1(t), p_2(t), \dots, p_k(t)) \tag{3}$$

F is an arbitrary function of k arguments; $p_i(t)$ is model parameters, the values of which are known at the moment t .

In fact, the graphical structure is expressed by three types of system dynamics model, which form the level equations by graphical representation (see Fig. 4).

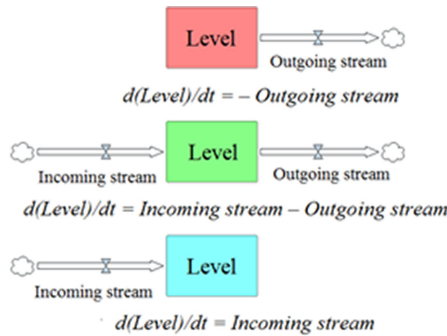


Fig. 4. Basic graphical structures of system dynamics.

Decision functions are pace Eqs. (1), which, unlike level equations (Fig. 4), are not so obvious and simple. In fact, the rate equations reflect an understanding of the factors that determine the actions that will be performed immediately at the next moment in time. Interaction in the system dynamics model occurs with the subsequent impact of paces on levels, which then, in turn, affect paces at later time intervals.

However, you cannot be guided only by the degree of direct influence considered by the parameter on the solution. It is necessary to take into account the ratio of the information flow, as well as the temporal characteristics – the delay, which characterizes the transformation process, as a result of which, based on the given rate of the incoming flow, the rate of the flow at the output is established (4):

$$Vd(t) = X(t + h)/P_{DEL} \tag{4}$$

P_{DEL} is a delay parameter representing the average time it takes to overcome the lag.

Running the system dynamics model provides, at best, obtaining reliable results at one point in the solution search space. However, the study of the environment requires the implementation of a series of experiments, the emergence of new branches of solutions as a result of bifurcation occurring due to the loss of stability of the standard state. Then it becomes necessary to choose an optimal solution from a set of possible situations, which is conditioned by certain criteria and given constraints.

3 Proposed Solution

According to the methodology of the situation-activity approach, it is necessary to identify the activities that exist in a complex dynamic environment. There can be a different number of activities in the environment that exist in the cycles of reproduction. These cycles divide activity into particular images which are acts of activity and spheres. Thus, as a result of decomposition, a hierarchical structure of a dynamically complex environment is formed [9]. This defines the reality boundary of the chosen subject area. For example, let us imagine a dynamically complex environment “Countering the development of infectious diseases” where there are many activities (see Fig. 5).

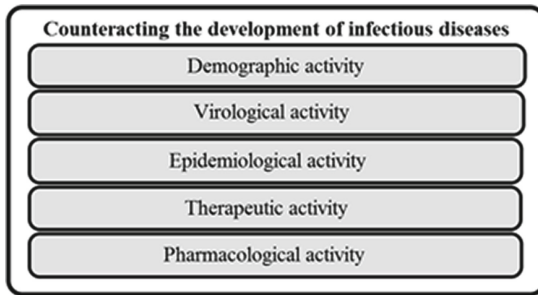


Fig. 5. Complex dynamic environment “Counteracting the development of infectious diseases” and the activities of prisoners in it.

3.1 Conceptual Structure

Thus, possibility to separate an activity from other realities by constructing its structure and to logically move from any element of this structure to another element exists. In this case, any units of activity can be distinguished; however, the elementary unit will be the act of activity [10, 11]. According to the synthesis of activity and situation analysis, let us present a conceptual scheme of the act of activity “Development of an infectious process” in virological activity (Fig. 6).

A conceptual model of a dynamically complex environment must be represented by different acts of activity, consisting of a set of conceptual structures of unit solutions. Then, from the conceptual structure of the act of activity “Development of the infectious process”, the conceptual structures of unit solutions are distinguished: actions “Fall ill” and “Recover”. In these conceptual structures of single solutions, the properties of the action components “Infection Equation” and “Recovery Equation” are not defined. These action components provide means for calculating the number of susceptible/infected/recovered people. We can define them in another indirect act of activity “Determine Infection Function”, which consists of two unit solutions “Calculate Infection Equation” and “Calculate Recovery Equation”.

From the conceptual structures of the acts of activity, four different contents of the act of activity stand out: the functional structure plan, the process plan, the context plan

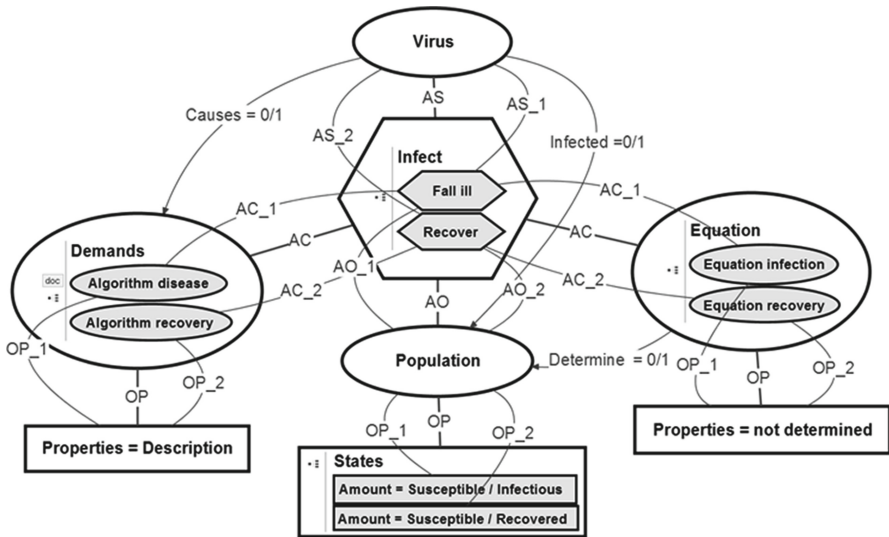


Fig. 6. Conceptual structure of the development of the infectious process.

and the analytical regularities plan. These plans are united into a coherent whole by a single structure in which they are expressed. Consequently, there is no contradiction between the plans - they not only complement each other, but can also be applied in parallel or in sequence to a certain extent. The description of these plans merits separate articles [5, 6]. Therefore, in this paper we will focus in more detail on the description of the process plan and the regularities for the system dynamics model.

The process plan of the act of activity is determined by the process as a set of actions (in a single solution, only an action), the object of the act of activity, its states – the source materials and products, the means of achieving the goal and their properties [6]. Based on the structure of the process plan, it can be assumed that it is similar to the basic structure of the quantitative stage, i.e. a diagram of flows and levels (see Table 1).


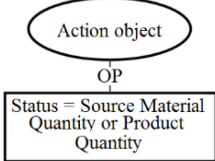

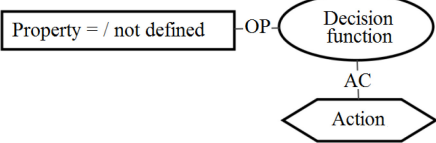

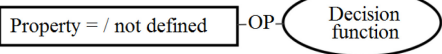
According to Table 1 the level is visualized as an object of action, which is defined by one of the quantitative states: product or the source material. The flow is interpreted by continuous activity, which is performed by an action that moves the content between the levels. If we use a hydrodynamic metaphor, the decision can be imaged as the function is a valve that controls the flow (defined as a means (tool) of the act of activity).

3.2 Regularities Plan

The plan of regularities is defined by gathering manifested properties of conceptual structured objects, which are described by a certain attitude and a ratio. Thus, regularities of transformation of initial material are fixed into a product (see Fig. 7).

During building of a plan for the regularities of a single solution, the following rules must be considered:

Table 1. Comparison of graphic notations.

Graphical notation for flow and level diagram	Process plan graphical notation
Level 	
Flow 	
Valve 	

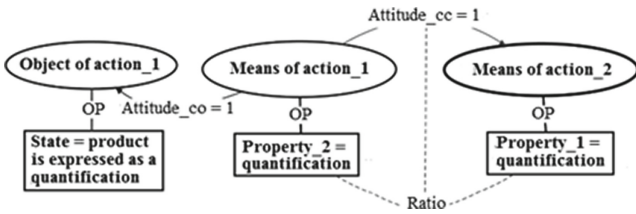


Fig. 7. Plan of regularities.

- the state of the action object (product) is associated with the solution, and the means of action - with the internal parameters of the equation.
- the relations in the regularities plan are unidirectional and singular, i.e., one relationship can come or go from one element.
- the relation between the object and the means of action and the relation between the means are presented by dependency relations and are conditioned by the types “enhances” or “delays”, the relation by the type “determines” is shown as an equal sign.

ratios are associations of dependency relations, which are defined by the following types “more on”, “less by”, “more in” and “less in”. Ratios associations, in turn, can be mapped as arithmetic operations (see Table 2).

Table 2. Association rules.

IF	THEN	
	IF	THEN
Relation type	Ratio type	Type of arithmetic operation
enhances	more on	Addition
	more in	multiplication
delays	less by	subtraction
	less in	division

3.3 Graphical Notations

Based on the regularities plan, various analytic views can be defined, including decision functions (flow rates) for system dynamics models. When modelling the activity “Development of an infectious process”, a synthesis of the process plan and the analytical regularities plan is used according to its structure. As a result, we get a structure identical to the flow and level diagram when constructing models of system dynamics (see Fig. 8 and Fig. 9).

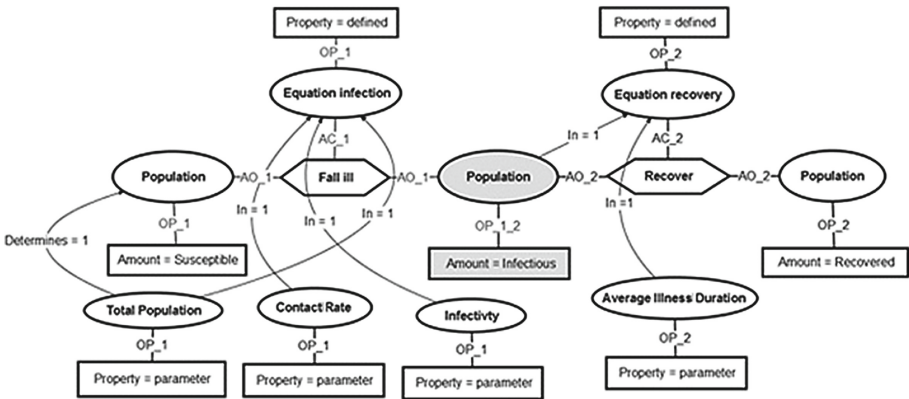


Fig. 8. Graphic notation based on the situational-activity approach.

According to the graphical notation, the population of the agglomeration undergoes several stages of transformation: *Susceptible* → *Infectious* → *Recovered*. This transformation in field of action analysis is finalized as a source material → product. Applying this concept and the corresponding software, the remaining activities are modeled (Fig. 5). The number of pools and flows enlarges correspondingly. Alike the pools already described before, the model should include population - the population of the simulated area; nosusceptible - people not susceptible, for whatever reason, to the disease. It must also include prophylaxis - the number of people covered by preventive

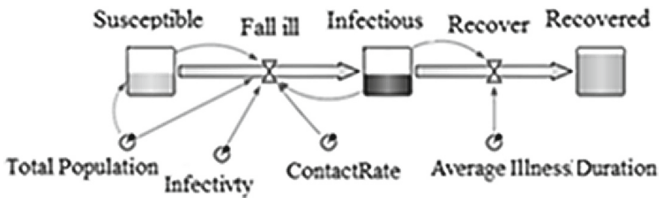


Fig. 9. Graphic notation based on the flow and level diagram.

measures; exposed - people with the disease who are in the incubation period; infectiousT - number of patients covered by therapeutic measures; final - people who died from complications. There are flows among the levels that can be arranged according to the acts of activity included in the environment of the escalation of an infectious disease. An epidemic model with a PS2E2I2RF-type phenomenology is implemented, which takes into account disease control measures [12].

Concept plans are essentially a knowledge base for designing knowledge-based systems. Accordingly, the task of separating such knowledge from the conceptual structure of the act of activity can be solved at the program level.

4 Results and Discussion

Building conceptual models is a non-trivial task requiring an understanding of the methodology of the activity approach, situational analysis, and system features of the subject area. In this case, there are tasks that can be quickly and efficiently solved only at the program level. The first task is to visualize conceptual structures; the second task is to check the conceptual model for completeness and adequacy; the third task is to generate a knowledge base for expert modeling; the fourth task is to generate a knowledge base for system dynamics.

4.1 “Designer” Module

To solve the first task, the software “Designer” is implemented. It is characterized by the following tools [10]:

- node tool – allows one to create different nodes that are based on a variety of geometric primitives (ellipse, rectangle, and irregular hexagon);
- link tool – allows creating various links, relationships, and ratio between the vertices of conceptual structures (geometric primitives) based on Bezier curves;
- text tool – allows one to change the textual content of elements of conceptual structures;
- zoom tool - allows zooming in on the node in question to a full-screen view to examine its contents in detail.

It is possible to customize the image parameters for shapes, lines and text, and to provide storage for conceptual structures. The conceptual structures depicted in the “Designer” software are represented as an XML (eXtensible Markup Language) document

that is represented as a node tree. This allows any element of the conceptual structure to be accessed. Consequently, it is possible to modify and manipulate their content at the software level.

4.2 “Solver” Module

The second and third tasks are handled by the “Solver” software, which is developed in the framework of situational analysis and supports the following set of functions [8]:

- creation, storage, modification, integrity testing, influence of user knowledge bases of production type;
- organization and some optimization of direct logical inference;
- generation of reports with text descriptions of knowledge bases and results of analysis of problem situations.

The performance of these functions results from interaction of the basic parts of the integration module and “Solver” software management:

- block of knowledge formalization (problem condition editor + object editor + object relations editor + rule base editor + situation model editor + knowledge base viewer);
- block of access to a situation model (problem situation model + current situation model + target situation model + knowledge base model);
- block of organization and carrying out logical inference (editor of rules management + module of carrying out direct logical inference + situation analyzer).

Thus, in “Solver” software, “Designer” software has an XML file which is read; the selection of the needed fragments takes turn, and then building of a primary logical structure that describes conceptual schemas as marked oriented graphs is complete. The user is notified, if any syntax errors are detected in the conceptual structures.

The knowledge base is also checked for completeness and adequacy. All sets of facts are divided into the following groups [10]:

- facts describing the initial (or problematic) situation;
- facts describing the target situation;
- facts not included in the initial or target situation are not considered.

The beginning of analysis of a set of conceptual structures of unit solutions for adequacy requires establishing the values of the facts of the problem and target situation. If needed situation is reached by means of logical inference, the fact about the adequacy and completeness of the generated recommendations for the relevant knowledge base is established on the basis of the content of the generated text report. Completeness and adequacy of a conceptual model of a dynamically complex environment are surveyed by exploring the logical conclusions on situations and on activity acts. Completeness is verified against classes of situations. The knowledge base considered incomplete or the rule management strategy is misconfigured if the logical inference is interrupted because the rules cannot be applied to the initial situation. Adequacy checking depends on logical

inference by class of activity acts. The resulting inference report must be consistent with the logic of the activity act process being modelled. If incompleteness or inadequacy is found, the knowledge base should be refined, or other rule management strategies should be attempted.

The knowledge base is implemented in a text file that specifies [8]:

- basic elements of the conceptual model;
- rule name <Subject Action Object>;
- the content of the rule in the form of the following construct: IF <Conditions before the action>, THEN PERFORM THE ACTION <Conditions after the action>.

Rules of the production describe the prerequisites that the objects' states involved in the action must satisfy, and the rules for changing the state of the objects at the end of the corresponding action. An expert system is implemented based on the rules.

4.3 “Interpreter” Module

The development of “Interpreter” software is based on the concept of streaming reading. It consists in representing an XML document as a stream of markers and allows you to create recursive descend parsers. This means that it is possible to split the XML parsing code into different classes. Thus, the structure of the tree is read and its objects are classified by relations (AS, AO, AC, OP (see Fig. 6)). After analyzing the XML document and forming the initial data set, the object search function is called. The execution of these actions leads to the structuring of the data required for the search of intersections and the selection of the necessary planning representations. Based on the found intersections, the necessary conceptual plan reports are output in a text file.

Based on this algorithm, four modules of the “Interpreter” program are developed. They are divided into two blocks: dynamic (functions and processes) and static (context and regularities) representations of the structure of the act of activity. Regarding the plans, a number of assertions are put forward:

- The functional plan allows designing discrete-event models.
- The process plan allows developing flow and level diagrams for system dynamics models.
- The regularities plan allows the design of analytical models in the form of simple mathematical equations.
- A context plan allows developing various models based on cognitive ideas.

Accordingly, within the framework of this report, we will consider the plans of processes and regularities regarding program recommendations for the development of models of system dynamics. In this article, the main elements of the process knowledge base are:

- level – [Object_N Quantity: = <Property of object before action>] and [Object_N Quantity: = <Property of object after action>];
- flow – [Action_N].

The identified intersections in the text file of the process plan are indicated and recommendations on its construction are proposed in the design of the following construct: Level 1 <Object N Number: = ‘Properties of object N before action N’ = Thread <Action N> = Level 1 <Object N Quantity: = ‘Property of object N after action N’.

In the article, the main elements of the knowledge base of the regularities plan are:

- equation – [Object_N];
- parameter – [Object_N Property_N after action_N];
- mathematical sign – [Ratio];
- equal sign - [Relationship <Defines>].

The text file of the regularities plan proposes recommendations for its construction in the model of the following construct: Equation <Object N> Equality sign <Ratio ‘Defines’> Parameter < Object 1Property of means 1 after action 1> Mathematical sign <Ratio> Parameter <Object 2 Property of means 2 after action 2>, etc.

Thus, an intelligent add-on for designing system dynamics models is implemented. It is represented by a knowledge showcase in the software package. The use of the software package is conditioned by the algorithm “Designer + Solver + Interpretator”:

- once the acts of action have been defined, their conceptual unit solution structures are implemented in “Designer” software and defined as a holistic conceptual decision-making model;
- holistic conceptual model of decision-making is checked for completeness and adequacy in “Solver” software. If necessary, a knowledge base report is generated in the form of product rules;
- after confirming the completeness and adequacy of the model in the “Interpretator” software reports on knowledge bases of conceptual plans are generated. Synthesis of knowledge bases corresponds to defined models and systems.

5 Conclusions

The practice of using systems and their synergetic combinations shows that nowadays, in its theoretical basis, there is no methodology and conceptual structure uniting situational, imitation, expert approaches to modeling information systems. There is not an integral knowledge model and a unified graphic description language for modeling in the field of information systems development; knowledge base representation software for information systems design [13].

These circumstances indicate the existence of a problem in the area of decision support, which consists in the absence of a unified conceptual structure of informed solutions concerning the management of complex systems and software extraction of different knowledge from the conceptual model for the design of information systems. The method and software toolkit proposed in this article provides an effective solution to these problems. Thus, the application of situation-action analysis to the study of dynamically complex environments is a type of simulation and is very relevant to the form of expert systems. Therefore, the establishment of the structure of an object is

largely identical with its knowledge as such. All other aspects are very imitative and determined by its structural organization. Like so, the structural characteristic of any object is central to its disclosure. Thus, conceptual structures are considered as a kind of a “substrate” on the basis of which different kinds of intellectual-information systems can be built.

The authors propose several directions for using and developing situation-activity analysis: as a language for understanding processes in a dynamically complex environment; as a language for modelling intelligent information systems; as an instrumental tool implemented in the software package “Designer + Solver + Interpretator”.

References

1. Brehmer, B.: Dynamic decision making: Human control of complex systems. *Acta Physiol. (Oxf)* **81**(3), 211–241 (1992)
2. Andrianova, E.G., Golovin, S.A., Zykov, S.V., Lesko, S.A., Chukalina, E.R.: Review of modern models and methods of analysis of time series of dynamics of processes in social, economic and socio-technical systems (Obzor sovremennyh modelej i metodov analiza vremennyh ryadov dinamiki processov v social'nyh, ekonomicheskikh i sociotekhnicheskikh sistemah ()). *Rossijskij tekhnologicheskij zhurnal* Russian Technol. J. **8**(4), 7–45 (2020)
3. Forrester, J.W.: Information sources for modeling the national economy. *J. Am. Stat. Assoc.* **75**(371), 555–574 (1980)
4. Rodrigues da Silva, A.: Model-driven engineering: a survey supported by the unified conceptual model. *Comput. Lang. Syst. Struct.* **43**, 139–155 (2015)
5. Sterman J.D.: All models are wrong: reflections on becoming a systems scientist. *J.W. Forrester Prize Lecture. System Dynamics Review* **18**(4), 501–531 (2002)
6. Chen, D.-Y., Zhao, H., Zhang, X.: Semantic mapping methods between expert view and ontology view. *J. Softw* **31**(9), 2855–2882 (2020)
7. Sorokin, A.B., Brazhnikova, E.V., Zheleznyak, L.M.: Designing a knowledge base for expert systems based on the conceptual structure of an act of activity (Proektirovanie bazy znanij dlya ekspertnyh sistem na osnove konceptual'noj struktury akta deyatel'nosti). In: *X International Conference on High-Performance Computing Systems and Technologies in Scientific Research, Automation of Control and Production*, vol. 4, iss. 1, pp. 190–196, 15–16 May 2020, Barnaul, Russia (2020)
8. Sorokin, A.B., Brazhnikova, E.V., Zheleznyak, L.M.: Designing a knowledge base for the development of intelligent models based on the conceptual structure of activity act. *Journal of Physics: Conference Series* **1615**, 012023 (2020). Doi: <https://doi.org/10.1088/1742-6596/1615/1/012023>
9. Gonzalez, C.: Learning to make decisions in dynamic environments: effects of time constraints and cognitive abilities. *Hum. Factors* **46**(3), 449–460 (2004)
10. Sorokin, A.B., Smoljaninova, V.A.: Conceptual design of expert systems of support of decision (Konceptual'noe proektirovanie ekspertnyh sistem podderzhki prinyatiya reshenij). *Informacionnye tekhnologii* **23**(9), 634–641 (2017)
11. Ganter, B., Obiedkov S.: *Conceptual Exploration*. Springer (2016)
12. Nakamura, G.M., Cardoso, G.C., Martinez, A.S.: Improved susceptible–infectious–susceptible epidemic equations based on uncertainties and autocorrelation functions. *Royal Soc. Open Sci.* **7**(2), 191504 (2020)
13. Tan, R.P., Zhang, W.D., Chen, S.Q., Yang, L.H.: Emergency decision-making method based on case-based reasoning in heterogeneous information environment. *Control Decis.* **35**(8), 1966–1976 (2020)