BULGARIAN ACADEMY OF SCIENCES
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2020 • Vol. **72**, pp. 17-28
p-ISSN: 0204-9848; e-ISSN: 1314-409X
https://doi.org/10.7546/PECR.72.20.03

Software Modeling of Technical Objects

Vasily Osipov[1], Alexander Vodijo[2], Radoslav Yoshinov[3], Nataly Zhukova[4]

[1] St. Petersburg Institute of Informatics and Automation of the Russian Academy of Sciences, Russia
[2] St. Petersburg State Electrical University "LETI" them V.I. Ulyanov (Lenin), Russia
[3] Bulgarian Academy of Sciences, Sofia, Bulgaria
[4] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia
Emails: osipov_vasiliy@mail.ru, yoshinov@cc.bas.bg, nazhukova@mail.ru

Abstract: In the paper is considered the automatic construction of models of technical objects based on the information flows coming from them. The current state and conditions for solving this problem are analyzed. It is proposed to use multi-level relatively finite operational automats when building the models. Such automats can reflect the behavior of both single and group objects. An example is given that demonstrates the possibility of using multilevel synthesis methods for modeling objects, including in flexible distributed environments.

Keywords: Technical object models, multilevel synthesis, state machines, flexible distributed environments, abstract computing.

1. Introduction

One of the promising areas of artificial intelligence is the creation of abstract computers capable of observing technical objects and generating control actions. For this, machines must be able to build and use spatio-temporal models of objects based on the signals received from these objects. Signals are structured data streams containing information about objects. In this case, there is a need to determine information elements, their processing and linking. Models can describe the structure of an object, its state and behavior at time intervals related to the present, past or future. When modeling, it is necessary to use an artificial intelligence apparatus (AI), in particular, methods of intellectual processing of statistical data, methods of pattern recognition, forecasting.

Various spaces for describing incoming signals are possible. In the general case, these are multilevel spaces of various characteristics inherent in analog and digital signals. At the upper level, these signals can represent messages tied to objects and their addresses. Each message has its own content, internal structure. Messages can contain essential information about both the sender and the recipient, as well as other objects. At lower levels, these can be segments (data blocks), packets, frames. At the lowest level, signals, for example, presented as a sequence of pulses, are described by their physical characteristics. Such characteristics may include the direction to the signal source, the carrier frequency at which they are transmitted, the amplitude, duration and shape of the pulses, their delays, repetition rate, duration of bursts, and others. Between these characteristics, there can be both linear and nonlinear dependencies.

With the automatic processing of simple signals containing time series or individual measurement results, there are no significant problems when building object models. They arise in the analysis of complex spatio-temporal signals. Currently, their solution is reduced to the development of highly specialized methods for processing disparate results of parameter measurements and linking the results to each other according to common properties.

Known methods for constructing models of objects are poorly focused on the use of information transmitted in signals. This significantly impedes the creation of control computers. A search is needed for new, more efficient methods for automatically constructing models of observed objects.

The article in the second section analyzes the current state of the methods of event binding and building object models. In the third and the forth sections, issues of software modeling of technical objects, including in distributed environments, are considered. The fifth section provides an example explaining the construction of these models.

2. Analysis of known models and data binding methods

Signal binding is known in publications, both philosophical and dedicated to specific scientific methods. The general works on this topic include the works [1], [8], [11], [12], which reveal the views on the classification and possible relationships between the various characteristics of the signals. Among the works on specific methods, along with others, can be distinguished papers [1-10].

By analyzing well-known approaches to signal binding, they can be divided into three groups: methods for binding events in different spaces without taking into account time; methods for linking events over time; spatial-temporal event binding methods.

The first group includes methods of static data analysis that are independent of time, for example, pattern recognition methods [12]. They are widely used in the processing of static images, texts, structures. The second group includes methods for analyzing time series, constructing various regression models, and others. The

third group of methods provides for the construction of complex models of objects based on joint spatio-temporal data analysis. At any point in time, a combination of events and static connections between them in a certain space can take place. Relationships between events related to different points in time may also occur. For modeling complex objects, systems of describing differential equations, real-time neural networks and other models built can be used [1], [15-22].

One of the first intensive studies on signal binding and the construction of models corresponding to them were done by A.A. Feldbaum and M.A. Aizerman. Fundamental work was carried out in a number of laboratories of the Institute for Management Problems [13].

Separately, one can highlight the achievements in the field of identification theory N.S. Rybman, who proposed methods for constructing mathematical models of control objects based on experimental statistics.

The next step in the development of signal binding was made as part of the formation of the neural network approach [18], [19]. For its successful application in practice, adequate neural network models and large computing resources are required. In the future, it is possible to implement large neural networks in the form of powerful intelligent neurochips and their assemblies.

So far, the known solutions are not perfect due to the lack of elaboration of models of recurrent neural networks in real time and the technologies for their implementation in the form of neurochips using memristors. Neural networks are required that provide not only fast, but also deep processing of perceived information flows. To some extent, this contradiction can be resolved by creating promising recurrent neural networks with controlled elements. The fundamentals of their construction can be found in [20-22].

The entire sufficiently varied apparatus of artificial intelligence discussed above can be useful in linking events. However, for the operational construction of flexible spatio-temporal models of observed objects, the coordinated use of many existing methods is required.

For a formalized description of object models in the form of connected changing structures, an automated approach to modeling can be used. The feasibility of using this approach is determined by the fact that automatic models reflect the state of objects, conditions and methods of transitions between them. During transitions, data processing about objects, event binding can be performed.

At the abstract level, classes of automats are distinguished in accordance with the properties of the objects that they describe. Classes are determined by the time model used, the dimension and power of the sets of states. The dimension of an automaton is determined by the number of generalized states, which can be finite or countable. Each of the generalized states can take values from a finite, countable, or continuum set, depending on the order of the recurrence relations that describe the behavior of finite automats. Automats of the first kind and second kind are

distinguished. The state of the automats may depend on one or many previous states.

For a more detailed description of the properties of automaton models, one of two systems of automaton models is used. The first is defined for the field of formal languages, the second for the field of logical control. In the first case, abstract automata are studied, in the second, structural automata. Abstract automata are focused on the recognition of software languages. Structural automata are considered as control devices. The number of input influences is not known in advance and not necessarily of course. Structural models can have automata without memory, automata with memory, automata without an output converter, Moore automats, Miles automats, mixed automats. Such automats are used in reactive systems [23]. An analysis of the properties of structural automats shows that they are applicable for a formalized description of object models [24]. There is experience in modeling technical and natural objects using an automated approach. However, previously used automats have a number of limitations. They allow you to build an only single-level model, which does not correspond to the structure of real objects. In addition, the possibilities for rebuilding automaton models of objects when changing modeling conditions are quite limited. Overriding only certain model parameters is allowed. There are problems with the high computational complexity of building object models, the need to improve the accuracy of models.

In [27], new hierarchical relatively finite automaton models were proposed based on single-level models that are devoid of the drawbacks discussed above and can be used to construct models of observable objects. Synthesis methods for such models were proposed in [28].

3. Software modeling of technical objects

In software modeling of objects using information systems, it becomes possible to build models of various sizes, i.e. implement a multi-scale approach to modeling. The scale can be set in relation to the simulation time. At a low time scale, the model reflects the subtle details of the behavior of objects. With an increase in the scale, details are excluded; in a more explicit form, the general trends in the behavior of objects appear in the models. The atomic elements can be models of individual nodes, aggregates, subsystems, or systems of the object under study.

Relations between models of various scales can be established using models and methods of inductive and deductive synthesis. In the first case, the relationships are built in the direction from low-scale models to higher-scale models. In the second case, the relationships are built in the direction from high scale to low scale models. The joint use of the ideas of multi-scale modeling and methods of inductive and deductive synthesis allows the formation of well-structured models. Using such models, it is possible to evaluate the stability of the observed states of objects and the changes occurring with them.

In a generalized form, the structure of a system that implements the construction of models of technical objects can be represented in the form shown in Fig. 2.

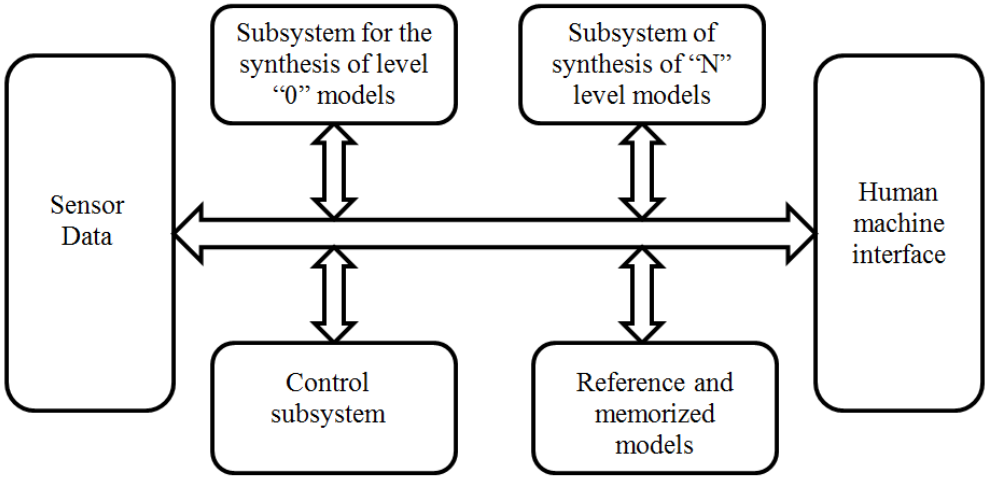


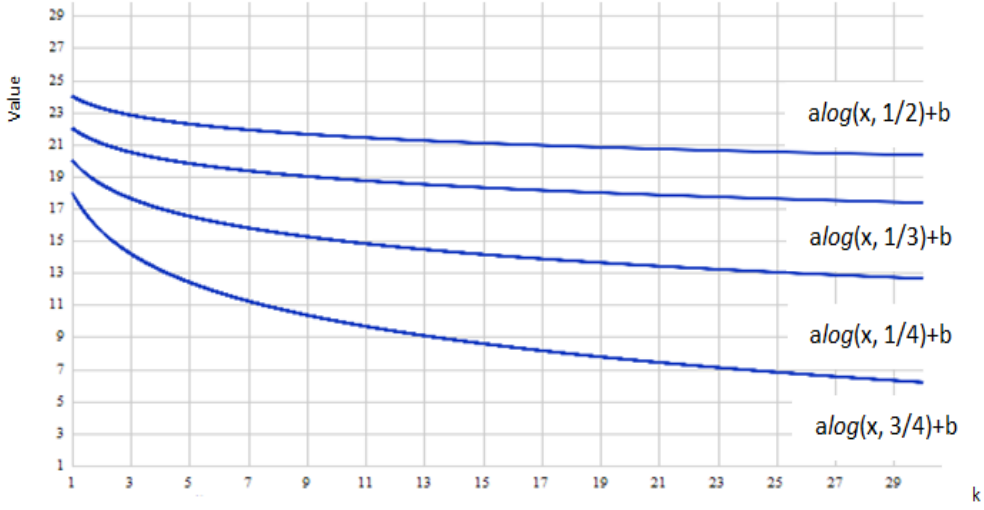
Fig. 2. The generalized structure of the system for automatically building models of objects

The system allows you to build models at different scales using the same algorithms, however the complexity of the simulation is significantly different. Complexity can be rated at $\frac{\alpha_k \alpha_N}{\alpha_k}$, where α_N is the number of model elements that are formed at the smallest scale. α_k is the number of model elements that are formed at scale k. Coefficient α_k is defined as dependency $\alpha_k(\alpha_{1,k}, \alpha_{2,k})$, where $\alpha_1(k)$ is average volume of changes made to the model in one simulation step, $\alpha_2(k)$ describes changes in the state of the object that are not taken into account when rations are estimated for scale k, regarding the smallest scale. Coefficients α_1 and α_2 can be justified by the following features of the modeling processes. The less often a model is built, the more operations are required to rebuild it. Part of the changes occurs between the points in time at which the models are reconstructed. On a large scale, some of them become indistinguishable by the time of the next restructuring of the model. The simulation results for several open data sets make it possible to obtain experimental estimates of the change in complexity with change in scale.

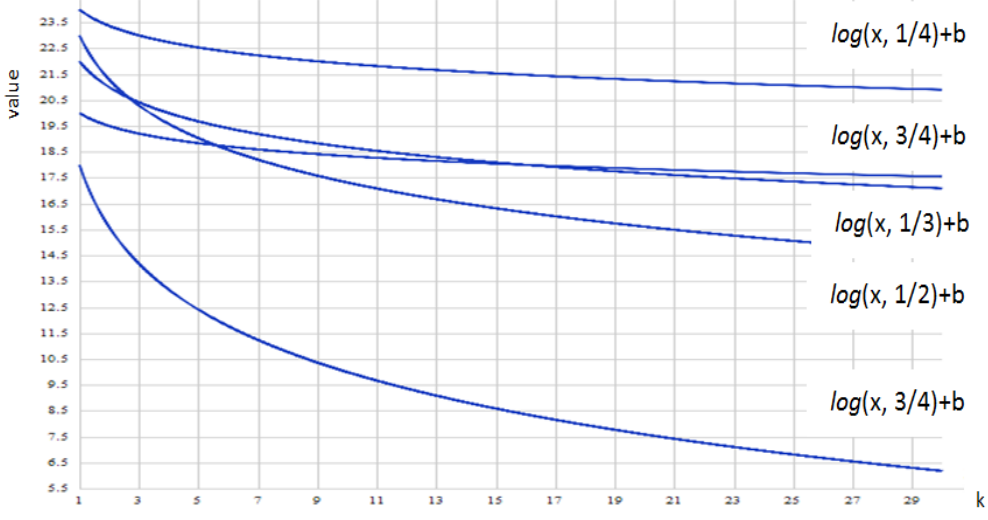
In a significant number of cases, a logarithmic dependence was observed. Examples of dependencies are shown in Fig. 3 a), b).

The graphs show the results obtained when processing data from two sources. The first source transmitted data on four objects, the second – about five. The main difference between the sources was the intensity of the data. The first source can be attributed to low-frequency, the second to high-frequency. The transmission

frequency is justified by the dynamic properties of the simulated objects. At different simulation intervals, the values of the coefficients a and b were different. The complexity assessment is given in arbitrary units, the values are normalized.



a) low frequency data source



b) high frequency data source

Fig. 3 Dependences of the complexity of building models on the time scale of modeling

To assess the changes, a graphical representation of the models was used. Quantitative values are obtained by calculating the distance between graphs.

4. Modeling of technical objects in flexible distributed environments

A flexible distributed environment refers to the technology of building the software and technical modeling infrastructure with the possibility of dynamic reconfiguration. In distributed environments, a consistent construction of object models is provided, implemented on various tools, which are usually combined into a network. Such an organization of resources is able to provide maximum computing power to solve the most complex and resource-intensive tasks [26]. A typical structure of a flexible distributed modeling environment is shown in Fig. 4.

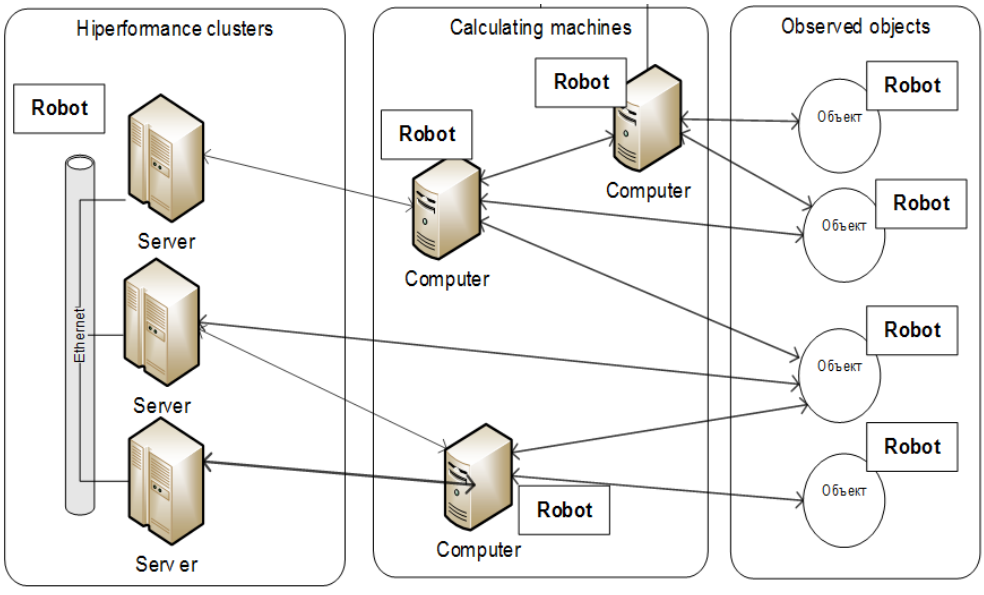


Fig. 4 Typical structure of a flexible distributed modeling environment

According to Fig. 4 models or their fragments can be built directly on the observed objects, as well as through the use of external computing tools. At the same time, external computers can be individual computers or high-performance clusters that combine many servers. The distribution of modeling tasks between elements of the environment can change with changing simulation conditions.

With flexible modeling in distributed environments, a software robot is placed on each element of the environment. The robot is an abstract computer that allows you to interpret the program modules loaded into it [25]. The composition and content of loaded software modules determine the behavior of the robot. Modules can be supplemented and changed in dynamics. The main distinguishing feature of the software robot is the simplicity of its organization and configuration, low resource requirements. There are several basic types of robots: autonomous robots, fully functional robots, simple robots. Autonomous robots are well-placed to

solve all the tasks needed to build models. They are able to independently shape the modeling processes and control the course of modeling, relying on a common algorithm for building models. Full-featured robots can solve individual modeling tasks provided that the composition of the raw data, the requirements for expected results are determined, and modeling processes are known. For example, such a robot can successfully solve the problem of calculating the characteristics of incoming signals or identifying associative dependencies between the values of characteristics. Simple robots can perform a limited number of fairly simple operations. These robots require minimal resources. They usually prepare the data for later processing. Preparatory operations include conversion of formats, identification of model structures and sequences, etc.

By installing and customizing robots on the elements of the simulation environment, it becomes possible to configure the environment flexibly. Here are some of the possible configurations:

The first configuration involves placing autonomous robots on the observed objects. The interaction of these robots with robots installed on external computations is limited. In some cases, the ability of objects to interact with external systems may not exist. Thus, most of the simulation tasks are solved by autonomous robots. With this approach, external systems resources are virtually non-utilized, and the load from data networks is also removed. This configuration can be applied when a surveillance object is a complex remote object.

In accordance with the second configuration, the bulk of the modeling tasks are solved on external computing. At the same time, they house full-featured robots, and on the objects of observation - simple robots. Robots placed on objects control the collection of data, ensure the implementation of incoming external requests and the transmission of the results. In this configuration, the main computational load falls on external means. This configuration can be used when a large amount of data needs to be processed, but the technical capabilities of the observed objects are limited.

A combined option is possible. This option involves a flexible distribution of simulation tasks between robots placed on external computers and objects. Robots have the capabilities of autonomous or fully functional robots. The amount of opportunities being realized depends on the results of dynamic task allocation. Tasks are distributed on the basis of identifying the best ways to organize the collective work of robots.

5. Example of modeling of technical objects

Let's look at modeling technical objects by example. Let radio frequency pulses act as elementary signals. When each such pulse is taken, the swaddling on the source of radiation, carrying frequency, amplitude, pulse duration, relative delay and other characteristics can be determined. As a result, each impulse can be matched by its

set of performance values. All signals received at the same moment or elementary time interval and their characteristics are linked in the spaces of their values. In this bind, the time is not taken into account. As a result of binding, graphs are formed (Fig. 5).

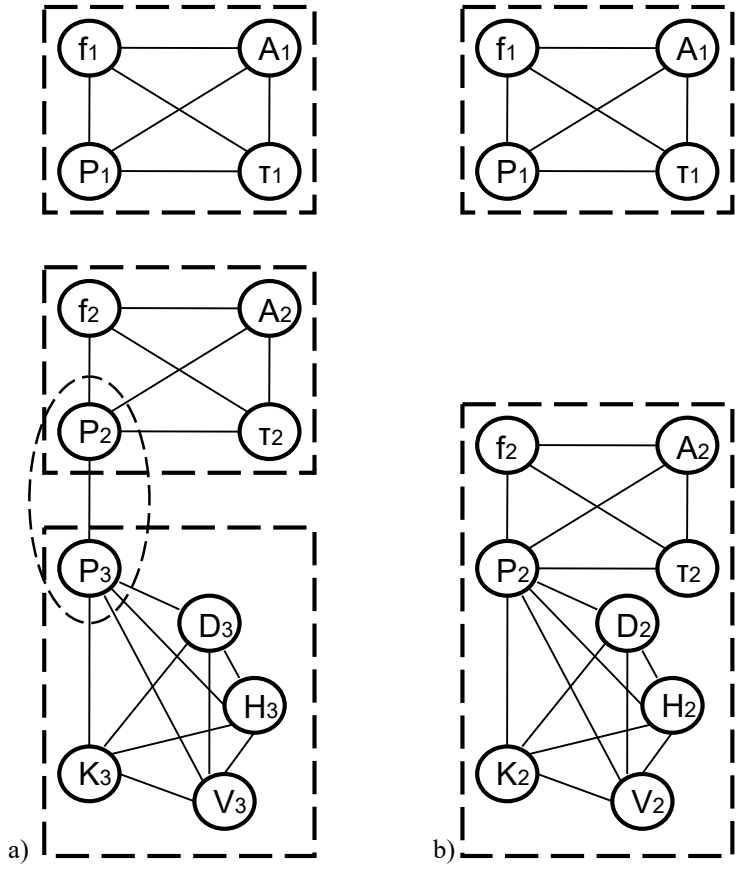


Fig. 5. Graphic representation of the binding characteristics of the simultaneously received signals: a) examples of links of signal characteristics; b) example of combining sets of signal characteristics. f, A, P, D, N, K, V – conditioned frequencies, amplitudes, swaddling, pulse duration, range to the object, its height, course, speed

The tops of these graphs correspond to the characteristics, and the arcs correspond to the relationship between the peaks on Fig. 5 and shows examples of signal links from three objects. Among them, the second and third objects have one common weighty sign in the form of the same value of the swaddling. This allows you to combine the characteristics sets of the second and third objects, believing that it is the same object (Fig. 5b). As a result, the second set of characteristics is supplemented by range values to the object, its height, speed and course of the track. This additional information about objects can be obtained through parallel active channels of extraction. This binding of signals without taking into account

the time from the position of building automatic models of observed processes allows to identify the structure of the input, internal and output data of the synthesized machine, to determine the permissible sets of it. By comparing successive sets of characteristics for pulses, the functions and parameters of transitions from one value to another can be defined. As a result, we get for each pair of consecutive sets of transition selections from one character value to another. In private, these can be functions that connect moments of pulses, frequency values, and others.

Transferring the machine to the next internal state provides that the inputs are those that were received at the previous moment. As internal states of the machine at the moment accepted previously remembered data. In a private case, the release of the machine can be considered as a dependence only on its internal state.

At the next level, the signals analyzed can be formalized in the form of reves of pulses of varying lengths, with specific repetition frequencies, recognized object modes and other characteristics. At this level, the functions and parameters of objects' transitions from one mode of operation to another can be established, taking into account values, such as the range and height of an object, its connections to other objects.

As a result, at each possible level, signals can be presented as threads of sets of their characteristics that are to be processed. Fig. 6 shows an example of such a scheme.

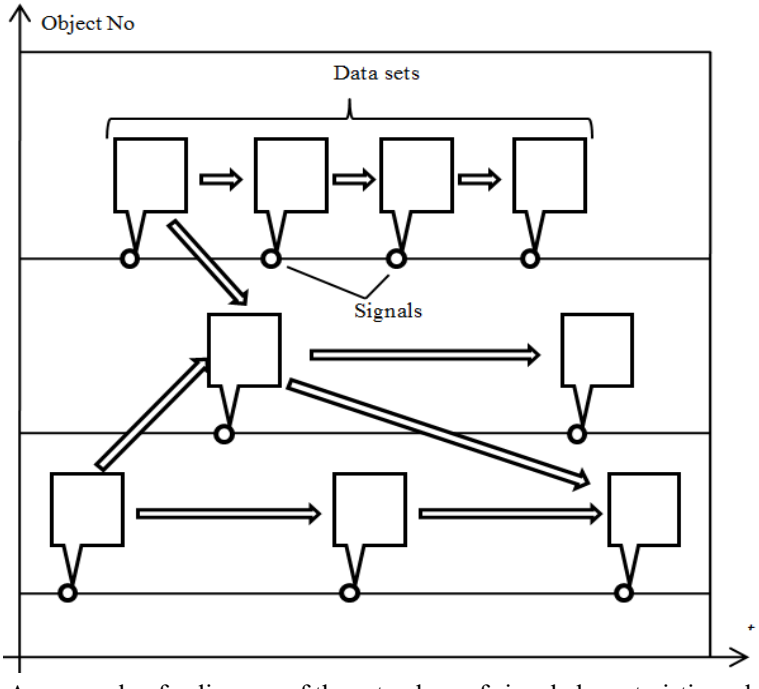


Fig. 6. An example of a diagram of the set values of signal characteristics relating to different objects

According to the proposed method, it is possible to build not only models of individual, but also group objects. In order to do so, there is a need to establish stable links between signals pertaining to different objects.
In all of these cases, the final operating machines are applicable to formalize the observed objects.

6. Conclusion

Analysis of known methods of automatic construction of models of objects has shown that they are largely imperfect. Methods of building full-fledged space-time models of such objects are only now beginning to develop actively.

In order to develop such methods, it is proposed to use multi-level, relatively finite operating machines as automatically formed models of observed objects. The example shows that hierarchical synthesis methods for end operating machines can be put into direct practice in the development of modeling software environments.

Acknowledgment

This work is supported by National Science Program “Information and Communication technologies for unified Digital Market in Science, Education and Security”.

References

1. David, L. (Ed.): Handbook of multisensor data fusion. Hall and James Llinas. CRC Press LLC. (2001).
2. Damerow, F., Knoblauch, A., Korner, U., Eggert, J., Korner, E.: Toward self-referential autonomous learning of object and situation models. Cognitive Computation 8, 703-719 (2016).
3. Korner, E., Knoblauch, A., Korner, U.: Autonomous situation understanding and self-referential learning of situation representations in a brain-inspired architecture. Advances in cognitive neurodynamics IV, 497-501 (2015).
4. Huelsen, M.: Knowledge-based driver assistance systems: traffic situation description and situation feature relevance. Wiesbaden: Springer. (2014).
5. Damerow, F., Eggert, J.: Risk-aversive behavior planning under multiple situations with uncertainty. In: IEEE 18th International Conference on Intelligent Transportation Systems, pp. 656-663 (2015).
6. Keysermann, M., Vargas, P.: Towards autonomous robots via an incremental clustering and associative learning architecture. Cognitive Computation 7(4), 414-33 (2015).
7. Czubenko, M., Kowalczuk, Z., Ordys, A.: Autonomous driver based on an intelligent system of decision-making. Cognitive Computation 7, 569-581 (2015).
8. Городецкий, В.И., Тушканова, О.Н. Ассоциативная классификация: аналитический обзор. Часть 1. Труды СПИИРАН 38. 183-203 (2015).

9. Scargle J., Norris, J., Jackson, B., Chiang, J.: Studies in astronomical time series analysis. VI. Bayesian block representations. Astrophysical Journal, doi:10.1088/0004-637X/764/2/167 (2013).
10. Yeh, C., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H., Silva, D., Mueen, A., Keogh, E.: Matrix, profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. IEEE ICDM. (2016).
11. Искусственный интеллект. В 3-х кн. Кн. 2. Модели и методы: справочник. Под ред. Д.А. Поспелова. М. Наука (1990).
12. Russell, S., Norvig, P.: Artificial intelligence: A modern approach. 3rd Edition. (2010).
13. Институт проблем управления им. В.А. Трапезникова Российской академии наук: краткое справочное издание. М., ИПУ РАН, (2009).
14. Mandel, A.: Expert-statistical data processing using the method of analogs. In: Proc. of 11th IEEE International Conference on Application of Information and Communication Technologies. pp. 147-151, (2017).
15. Гильберт, Д., Аккерман, В.: Основы теоретической логики. М., КомКнига. (2010).
16. Ganter, B., Stumme, G., Wille, R. (Eds.): Formal concept analysis foundations and applications. Springer-Verlag. Berlin Heidelberg. (2005).
17. Osipov, V.Y.: Automatic synthesis of action programs for intelligent robot. Programming and Computer Software 42(3), 155-160 (2016).
18. Haykin, S.: Neural networks and learning machines. 3rd ed. Prentice Hall. New York. (2008).
19. Galushkin, A.I.: Neural networks theory. Springer Science & Buisness Media. Berlin. (2007).
20. Осипов, В.Ю.: Прямое и обратное преобразование сигналов в ассоциативных интеллектуальных машинах. Мехатроника, автоматизация, управление 7, 27-32 (2010).
21. Осипов, В.Ю.: Рекуррентная нейронная сеть с двумя сигнальными системами. Информационно-управляющие системы 4(65), 8-15 (2013).
22. Osipov, V.: Space-time structures of recurrent neural networks with controlled synapses. Advances in Neural Networks, 177-184 (2016).
23. Поликарпова, Н. И., Шалыто, А. А.: Автоматное программирование. Изд. Питер, (2011).
24. Сраговић, В.Г.: Теория адаптивных систем. М.: Наука, (1981).
25. Osipov, V., Vodyaho, A., Zhukova, N.: About one approach to multilevel behavioral program synthesis for television devices. International Journal of Computers and Communications 11, 17-25 (2017).
26. Hwang, K., Fox, G., Jack, J.: Distributed and cloud computing from parallel processing to the internet of things. Elsevier. (2012).
27. Osipov, V., Stankova, E., Vodyaho, A., Lushnov, M., Shichkina, Y., Zhukova, N.: Automatic synthesis of multilevel automata models of biological objects. In: Computational Science and its Applications. LNCS, 11620, 441-456 (2019).
28. Osipov, V.Y., Vodyaho, A.I., Zhukova, N.A., Glebovsky, P.A.: Multilevel automatic synthesis of behavioral programs for smart devices. In: 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), Prague, pp. 335-340 (2017).