Methods and Models for Building Adaptive Automated Systems of Organizational Management

Aleksandr Dodonov[1], Aleksey Nikiforov[1] and Vladimir Putyatin[1]

[1] *Institute of Information Registration Problems of the National Academy of Sciences of Ukraine, Shpaka street, bldg. 2, Kyiv, 03113, Ukraine*

Abstract

The adaptation of automated organizational management systems (ASOM) is closely related to the concept of distributed systems. In this context, adaptation involves modifying structural relationships between information blocks, adjusting processing algorithm parameters, and incorporating reflective elements. Currently, the field of ASOM adaptation within complex distributed systems has emerged, where specialized mathematical models and methods are employed. This paper presents an analysis of the scientific and methodological framework for ASOM adaptation, including examples of successful applications of ontology synthesis and conceptual structure synthesis using degree-based structural typologies. Limitations of current approaches in addressing adaptation challenges, particularly with respect to conceptual changes, are identified. Further development directions are suggested, focusing on enhancing the scientific and methodological framework through tensor transformations in electrical multicoil networks.

Keywords

Automated Management System, Organizational Management, Adaptation, Conceptual Design, System Algebra, Set Degrees, Theoretical-Systemic Construct, Conceptual Scheme, Constituent, Confinement Model, Ontological Universality, Types of Structures.

1. The problem of ASOM adaptation

Automated Systems of Organizational Management (ASOM) consist of the following elements (**Figure 1**):

- **System Model**: Includes a description of functioning and management processes (decision-making) and a logical model of the corresponding database aligned with this description.
- **System of Output Data**: Incorporates decision-making rules or axioms.
- **System of Computational Procedures**: Encompasses information processing and transformation procedures.

Typically, ASOM are tailored to specific conditions, with processes structured in defined control loops that provide partial adjustability within a specified range of external conditions. However, with the emergence of new, previously unconsidered tools, methods, interconnections, and relationships between management elements, ASOM can quickly become outdated. Consequently, the need arises to adapt ASOM to new management concepts. Attempts to revise and modernize ASOM often lead to software conflicts and contradictory situations, making it easier at times to initiate a new project rather than upgrade an existing system.

ITS-2023: Information Technologies and Security, November 30, 2023, Kyiv, Ukraine
dodonovua@gmail.com (A. Dodonov); aleksey.nikiforov.62@gmail.com (A. Nikiforov); putvlgr@gmail.com (V. Putyatin)
0000-0001-7569-9360 (A. Dodonov); 0000-0002-0207-5175 (A. Nikiforov); 0000-0002-5575-9159 (V. Putyatin)

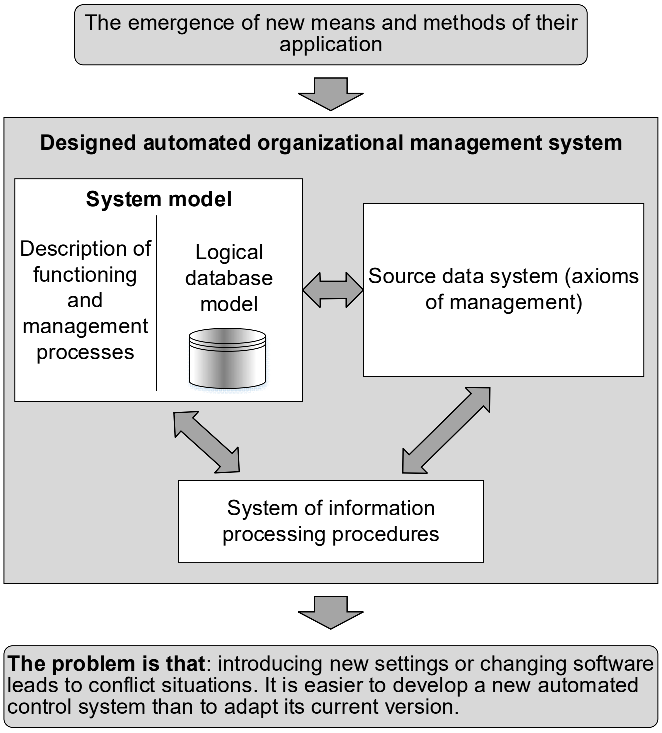

Figure 1: Key Structural Elements of ASOM and the Problem of Their Adaptation.

As technological cycles shorten, the urgency of adapting ASOM structures or concepts intensifies. Such adaptation ideally should occur without a complete redesign or the need for new automation tools, instead relying on transformative procedures that require minimal time and resources. Therefore, when designing new ASOM, it is advisable to integrate adaptability features, preferably in an automated manner.

To address ASOM adaptation, several key research directions should be emphasized. The organization management process operates in three modes (**Figure 2**):

- **Planning or Programming Mode**: Focused on preparing for future operations.
- **Adjustment Mode**: Modifies a pre-existing plan during its execution (targeted management).
- **Operational Management Mode**: Manages ongoing operations in real-time.

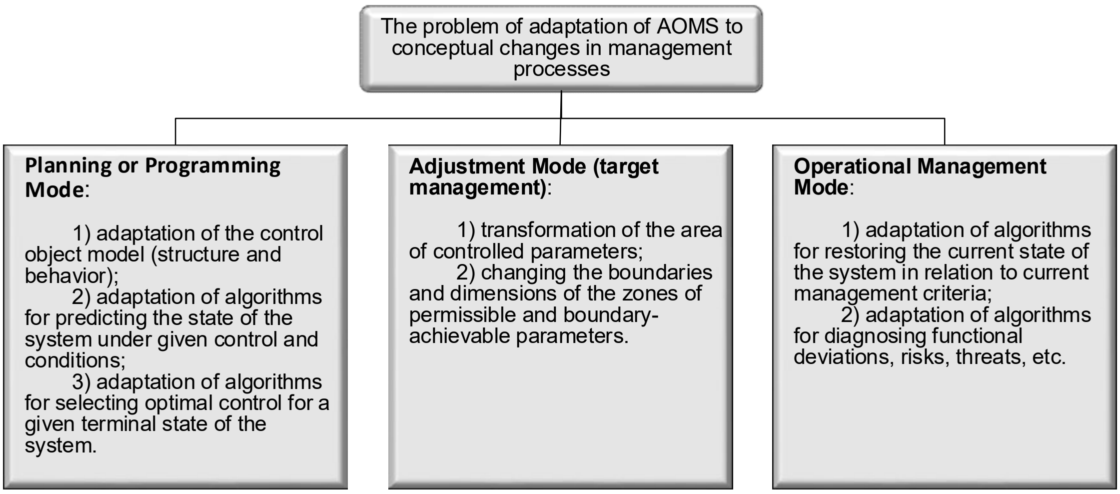

Figure 2: Management Modes and Current Research Directions for Addressing ASOM Adaptation Issues.

Achieving ASOM adaptation to a new management concept during the planning mode requires mechanisms that transform:

1. **Management Object Model**: Restructures the management process to align with a new conceptual framework, including functional relationships and responses to external influences.
2. **Prediction Algorithms**: Adjusts ASOM software to predict the system's state under the new management model and external conditions.
3. **Optimal Management Algorithms**: Updates decision-support algorithms to plan for the organization’s functioning under the modified model and performance criteria.

For the adjustment mode of previously created plans, transformation mechanisms include:

1. **Controlled Parameter Areas**: Expands the parameters, dimensionality, and structure of inter-parameter relationships.
2. **Acceptable and Limit-Reachable States**: Adapts criteria for achieving established goals.

For operational management, transformation mechanisms involve:

1. **Current State Algorithms**: Reflects observed parameters in the cognitive model of the management process.
2. **Diagnostic Algorithms**: Identifies functional deviations, risks, etc.
3. **Decision Support Algorithms**: Provides real-time support for current management decisions.

This detailed breakdown of the adaptation problem highlights the importance of developing mathematical and software solutions for ASOM. Targeted research in these areas will lay the theoretical groundwork for adaptive ASOM development. Some scientific advancements have already been made in synthesizing structures with defined properties and in the conceptual design of complex systems.

2. Mathematical Models and Methods for Constructing Adaptive ASOM

In constructing adaptive Automated Systems of Organizational Management (ASOM), two primary approaches, closely linked through ontology application, can be identified (**Figure 3**):

- **System Algebra Framework**: Utilizes Boolean functions, predicate calculus, and logical inference [1].
- **Set Degree or Structure Type Framework**: Also known as conceptual design [3], this approach relies on degrees or types of structures [2].

The system algebra framework combines mathematical logic with algebraic structures, such as groups and rings, which serve as models for processes, and lattices, which model structural relationships. The set degree framework similarly employs algebra but focuses on set degree formation and logical procedures. Together, these approaches support the development of adaptive mechanisms for systems, with multiple successful implementations already demonstrated.

2.1. Confinement Models

As proposed in [4], confinement models (CM) provide a structured approach to building ontologies, allowing a systemic-cognitive method that remains impartial to individual researchers' modeling techniques. Using CMs, original ontologies are constructed based on ontological universals, which model relational systems across various subject areas.

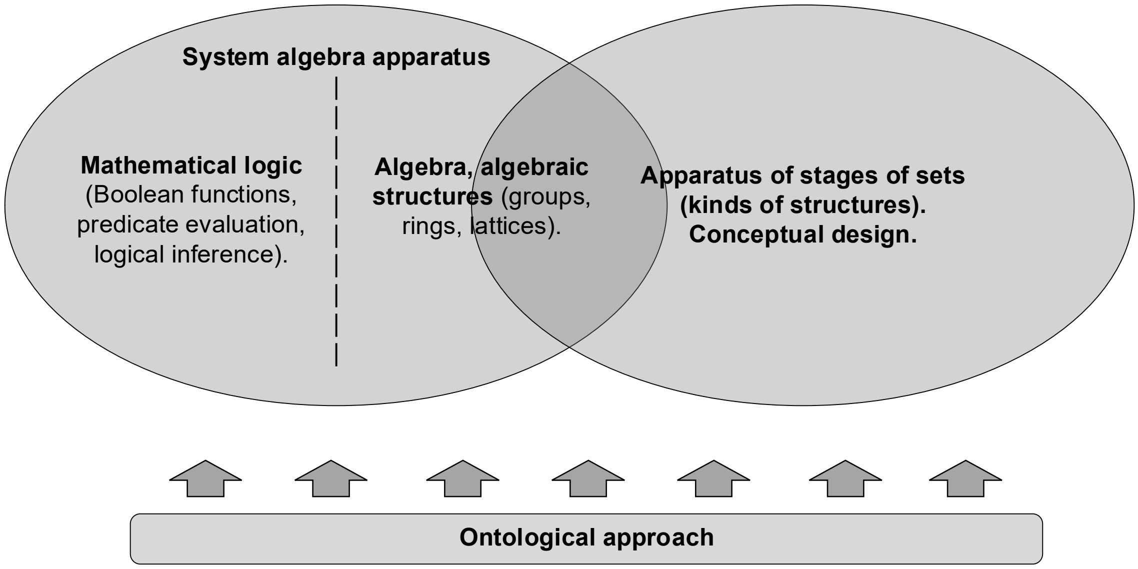

Figure 3: Mathematical Models and Methods Used to Solve the Problem of ASOM Adaptation.

The following types of confinement models are identified (**Figure 4**):

- **Conceptual CM (CCM)**: Connects elements through "causes/depends on" relationships, with a specific case being the triadic CM (TCM) (**Figure 4a**).

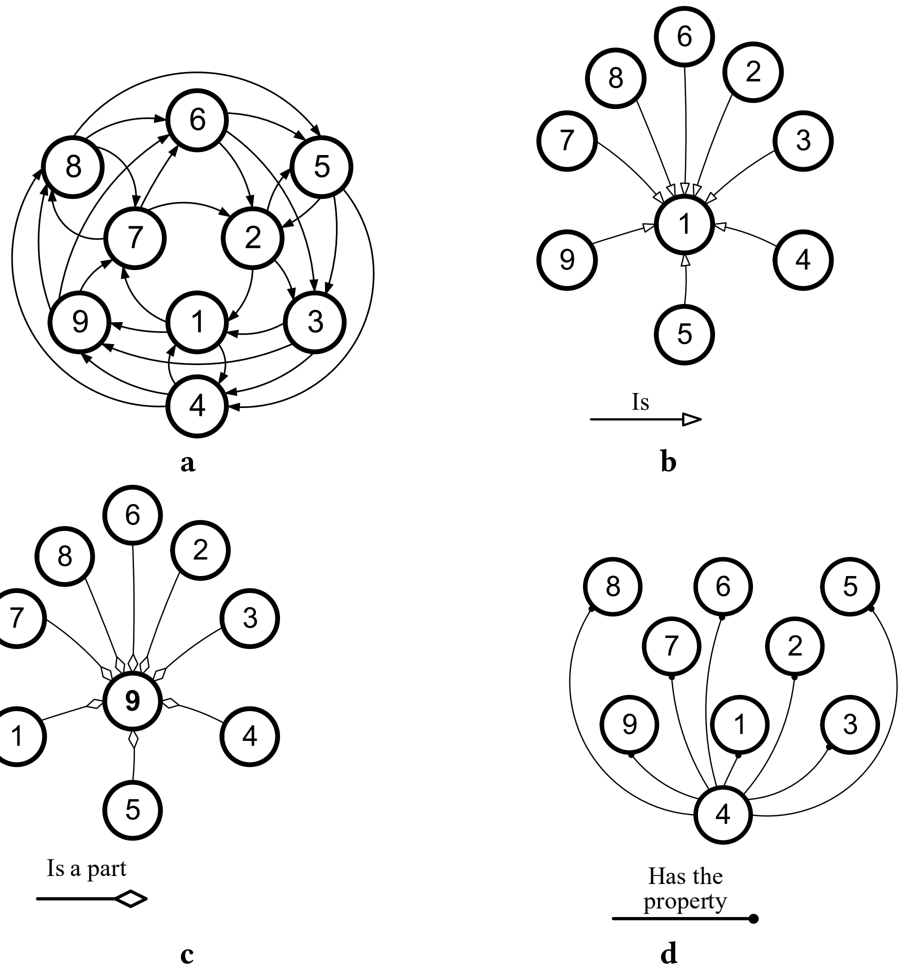

Figure 4: Types of Confinement Models: a) CCM; b) GCM; c) MKM; d) AKM

- **Hierarchical CM (HCM)**: Includes various forms:
  - **Hypernymic (HCM) (Figure 4b)**: Classifies concepts using hierarchical genus-species relations, expressed as "is."
  - **Meronymic (MKM) (Figure 4c)**: Analyzes subsystems systematically with "is part of" relationships.
  - **Attributive (AKM) (Figure 4d)**: Classifies properties using the relation "has a property."
- **Process CM (PKM)**: Structures processes by identifying life cycle stages with "is input/output for" relationships.

Through CMs, knowledge structuring can be achieved using interrelated cognitive models of a specialized kind, setting a standard for ontology detailing by various researchers. This approach facilitates the development of formal methods for operating and transforming ontology structures.

2.2. Automatic Adaptation of Information Processing Algorithms

In [5], ontological task models are proposed for developing software systems capable of adapting to changes in a given domain. Here, a "task" in the task ontology refers to a structured problem situation defined by specific conditions and goals (criteria). The domain ontology represents the management process as a series of elementary tasks. A formal ontological model is constructed using system algebra [1], and practical algorithms for transforming task ontologies are implemented in the CLEPE environment (Conceptual Level Programming Environment), adhering to principles of conceptual modeling [3].

In the modeling system, knowledge is formally represented as follows:

The domain consists of a set of objects

$$A_1, A_2, \dots, A_n ,$$

where each object is classified as an instance of a corresponding concept. These object sets serve as the basis for n many-sorted algebras.

Each set A_i, defines an abstract data type:

$$E_i = (Name, \Sigma, E_x), \quad (1)$$

where $Name$ denotes the type name (e.g., circle); Σ is the signature of the many-sorted algebra (circle parameters like center coordinates, radius); E_x is the defining relations of the type (circle equations).

An algebraic domain set is formed by types E_i

$$E_i = (E_1, \dots, E_i, \dots, E_n). \quad (2)$$

Additional domains are defined as follows:

- **Attribute Domain**, At, defined as key-value pairs:

  $$\{key, value\} ,$$

  where key is the attribute identifier (for example, "x"); $value$ - its value (real number). Operations such as merge, substitute, delete, and interpret (merge, substitute, delete, enterp) are defined for this domain. In addition to these operations, various functions are defined, for example, the value selection function of the domain:

  $$F_{setval}(At, key) \rightarrow value ;$$

- **Boolean Domain**, Cs, contains expressions yielding Boolean values $\{true, false\}$ with variables from other domains. It includes Boolean operations *"AND," "OR," "otherwise"* and simplifies conditions via the interpretation operation (=). Cs elements are constraints (conditions).
- **Entity Domain with Attributes**, T, represented by tuples

  $$\{(E_i, At_j, Cs_j^*)\} ,$$

where for each *i* there is only one *j*, that forms an element of the T:

$$\forall i\ \exists^1 j: (E_i, At_j)\ ,$$

each condition (constraint) belonging to the subset Cs_j^*, is an expression with operands belonging to the domain At_j. Operations on elements of this domain are the union or division (merge, split) of entities. The union operation is interpreted as the synthesis of a new entity with a compatible set of properties and constraints (conditions) of operand entities. The division operation is the reverse. Instances of the domain of entities with attributes are tuples of corresponding facts;

- **Relation Domain**, RI, on structures:

$$\{(T_{1,i} \times \ldots \times T_{k,i}, At_i^r, Cs_i^*)\}\ ,$$

where $T_{1,i} \times \ldots \times T_{k,i}$ is a Cartesian product of algebraic data types from the T domain (attribute space related to the *i*-th structure); At_i^r - types of relation attributes; Cs_i^* - constraints (conditions) defining the relation. Operations such as union, separation, substitution (merge, split, substitute) are performed on relations.

The ontology itself is a tuple of concept domains with attributes, relations, and constraints:

$$On = (T, RI, Cs^*). \qquad (3)$$

The problem's ontological model is described by the tuple:

$$Md = \langle On_{TS}, Ac_{TS}^*, Cs_{TS}^* \rangle\ , \qquad (4)$$

where On_{TS} is the problem ontology according to the construction rules (3); Ac_{TS}^* is the task ontology; Cs_{TS}^* is the set of additional constraints (conditions). Each action is an entity within the T domain, serving as a command to external services or other ontological models. Parameters are attributes from On_{TS} or constants.

The transformation of algorithms depending on changes in conditions and constraints is carried out using the parameter initialization function (*InstPar*), which is given as a set of mappings between the action attribute and the attribute values of entities and relations in the ontology model:

$$InstPar: (Ac_{l,i}, pkey_{l,i}) \to SelVal(At_{l,j}, key_{l,k})\ , \qquad (5)$$
$$Ac_{l,i} \in Md_l, At_{l,j} \in On_{TS}^l \in Md_l\ ,$$

where Md_l is the ontological model of the considered task; On_{TS}^l is the task's ontology.

The software system based on the ontological model of tasks is presented in **Figure 5** [5].

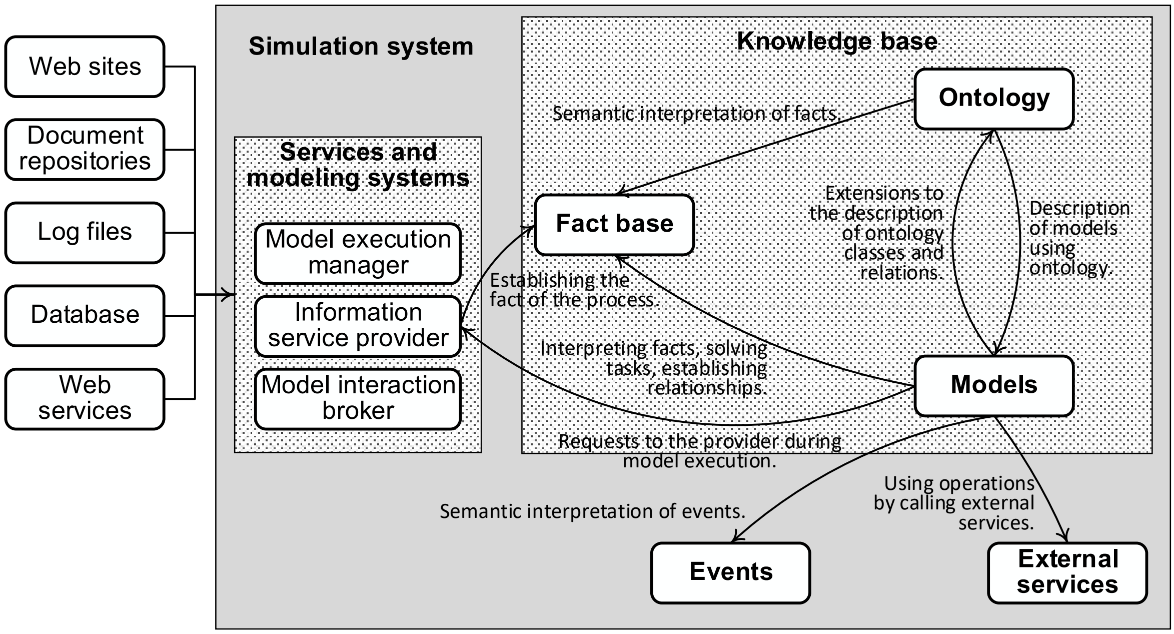

Figure 5: The structure of a software system based on ontological task models.

The management process in the intelligent control system generates events, which are interpreted using the ontology of the corresponding subject area. Based on the interpretation of the event, models and algorithms necessary for implementing the computational process for management in current conditions are activated. Using the ontology of models and algorithms, a new computation schedule is created. The synthesis of the schedule is carried out using the apparatus of mathematical logic (logical inference).

Depending on the context of the occurring events, either one or another ontological model is activated. The initiator of activation is the ontological model being executed (**Figure 6**) [5].

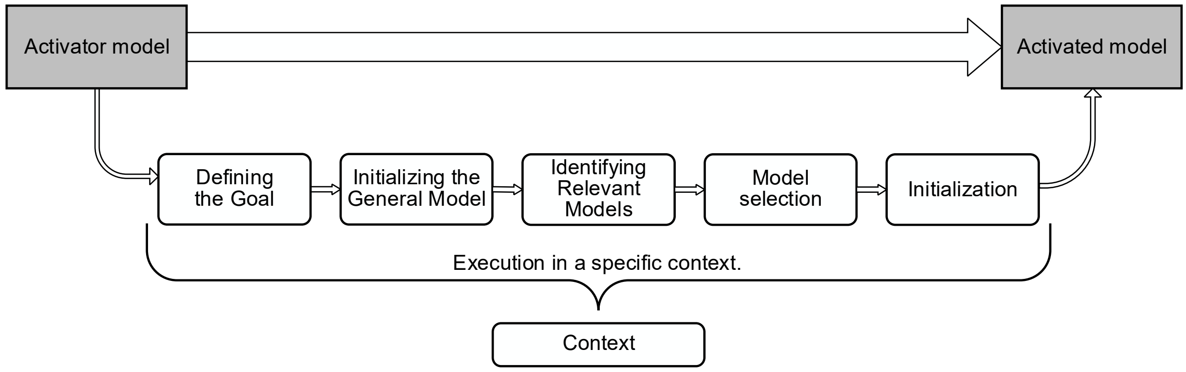

Figure 6: The process of activating a task model.

This framework also supports the creation of new ontologies. This involves the following steps:

1. Analyzing the problem and identifying relevant entities, relations, constraints, and operations.
2. Constructing the ontological model using available components, supplemented with any newly identified elements.
3. Validating the updated ontology and resolving contradictions.

The ontology-based tool for building software systems offers functionalities including:

- **Ontology Operations**: Creation and modification of ontology classes, relationships, and attribute constraints.
- **Fact Operations**: Creation and modification of individual facts, constraint verification, and fact validation.
- **Ontological Model Management**: Creation and editing of models using classes and facts, defining constraints, operations, and service links.
- **Execution of Models**: Testing models on a factual base and verifying outcomes.

The system is implemented in Python with PyQt for its graphical interface, and includes components such as an ontology editor, fact editor, model editor, and simulation programs. Future development will focus on expanding model interactions, implementing logical inference, enabling multi-variant computations, and addressing contextual dependencies.

2.3. Deployment of Security Domain Ontology Based on Conceptual Schemes of Abstract Security Relations

In [6], a methodology is outlined for modeling the "security" domain using conceptual analysis and design, specifically focusing on a mixed inter-subjective and subject-object security relation derived from an abstract security structure. This approach is based on several key postulates for security relations [7]:

- There exists a world of possibilities.
- A possibility can be actualized and then ceases to be a possibility.

- An element of the world of possibilities is the possibility of realizing one or another event.
- Possibilities can be connected by a "genetic" relation (the realization of one or several possibilities is a necessary condition for the realization of another possibility).
- Possibilities can be linked by a blocking relation (if a blocking possibility is realized, a blocked possibility cannot be realized).
- Possibilities are attributes of subjects and objects in the "real world" that realize these possibilities.
- There exists a world of subjects.
- Subjects have interests.
- The interests of different subjects can enter into relations.
- Relations between subjects occur only through their interests.

In the world of possibilities, complex network structures with cycles can emerge.

A subject's interests nominally constitute a subset of its possibilities (the subject is interested in realizing these possibilities).

Any possibility that blocks possibilities from the subject's area of interest represents a threat to the subject's interests.

A possibility that blocks a threat is a security measure concerning that threat.

An action to realize a possibility is a possibility that is formative for the subject related to that possibility.

Since threats and actions regarding the realization of possibilities can also define threats and measures, it can be said that there are first, second, third, etc., order threats and measures.

The ontology construction process involves iteratively breaking down the factors in a structure-tree format within this set of possibilities. This breakdown involves:

- **Concept Elaboration**: Specification and detailing of initial concepts.
- **Inter-branch Relations**: Formation of relations between ontology branches by integrating hierarchical levels and factor-based relationships.

Using conceptual schemes of abstract security relation structures [7], the ontology's factor-structure is built, including types like:

- **Abstract Security Relations**: Basic interconnections and constraints among possibilities.
- **Hierarchical Subject-transforming Security Relations**: Transformational security relationships at different subject levels.
- **Polysubjective Security Management**: Relations involving multiple subjects managing shared security interests.
- **Danger Propagation**: Mechanisms for identifying and propagating security threats across the structure.

A fragment of Security Structure Types is provided in Table 1.

Table 1
The conceptual scheme of types of structures of abstract security relations (fragment)

Denotation of constituents	Formal expression of constituents	Schematic interpretation of constituents
$X1$		Set of possibilities.
$X2$		Set of subjects.
$D1.1$	$\mathcal{B}(X1 \times X1)$	Relationship of genetic connection of possibilities or set of pairs: possibility - possibility, which proves to be a necessary condition for the realization of the considered possibility.

$D1.2$	$\mathcal{B}(X1 \times X1)$	Relationship of blocking connection of possibilities or set of pairs: possibility - possibility that blocks the considered possibility.
$T1.0$	$\{\alpha \subset \mathcal{B}(X1 \times X1) \mid \mathrm{Pr}_1\alpha = \mathrm{Pr}_2\alpha\}$	Set of cycles of possibilities.
$A1.1$	$\{(\alpha \subset T1.0)\&(\alpha \subset \beta) \Rightarrow ((\alpha \subset D1.1) \vee (\alpha \subset D1.2))\} \Rightarrow: (\beta = \emptyset)$	Axiom. Relations of genetic, blocking, as well as mixed genetic and blocking connection of possibilities do not allow cycles and loops.
$A1.2$	$(\alpha \in \mathcal{B}(X1 \times X1))\& ((d_1 \in \alpha) \supset (d_1 \subset D1.1))\& (x_1 \in \mathrm{Pr}_1\alpha)\&(x_2 \in \mathrm{Pr}_2\alpha)\& (\mathrm{Pr}_1\alpha \backslash x_1 = \mathrm{Pr}_2\alpha \backslash x_2) \Rightarrow \neg((\beta \in \mathcal{B}(X1 \times X1)\&(d_2 \in \beta))) \Rightarrow (d_2 \subset D1.2)\&(x_2 \in \mathrm{Pr}_2\beta)\& (x_3 \in \mathrm{Pr}_2\beta)\&(\mathrm{Pr}_1\beta \backslash x_1 = \mathrm{Pr}_2\beta \backslash x_1)$	Axiom. Possibilities connected by a genetic relationship cannot be connected by a blocking relationship, and vice versa.
⋮	⋮	⋮

Constituents presented in Table 1 are types of abstract security relation structures synthesized using formal rules of transforming text structure when applying set degree formation operations [7].

The handbook of theoretical constructs [8] based on this procedure contains over 200 system classes, covering both static systems (fixed relations) and dynamic (degrading) systems, facilitating a deductive approach to security management system design—from abstract principles to concrete applications. A key challenge remains in interpreting high-degree set constituents, which, due to their complexity, require specialized tools for practical use.

3. Development of Adaptation Methods and Models for ASOM

3.1. Current Challenges in Adaptation

The scientific results presented are an example of a fairly successful solution to the problem of adapting ASOM. However, it cannot be claimed that the problem is fully solved. There are limitations and drawbacks to the methods that do not allow concluding the completion of creating a conceptual transformation apparatus for management systems.

Therefore, when adapting the computational procedures' regulations (see section **Помилка! Джерело посилання не знайдено.**), the ontology of models and algorithms is used, which is pre-formed and entered into the knowledge base. The task of expanding and developing the ontology based on the interpretation of facts and events remains unresolved. This leads to the need, once again, to create a new version of the mathematical and software when there are conceptual changes in the management process for which the ASOM is created. The adaptation or "learning" of ASOM under a new management concept has not yet been implemented.

For the method of using conceptual schemes of structure types (see section **Помилка! Джерело посилання не знайдено.**), the bottleneck is the interpretation of the created abstract schemes. For higher set degrees (which are significant in practice), the generated constituents are expressions with a very large dimension. Their size exceeds human capabilities to operate with data blocks. In [8], to overcome this difficulty, it is suggested to create a special language for the formalized description and representation of set degrees. In both cases, the essence of the difficulties hindering the implementation of ASOM adaptation lies in creating a method and corresponding software for synthesizing structures with specified properties.

3.2. Proposed Solutions to Adaptation Challenges

As a method for synthesizing new structures, tensor transformation of networks developed by Gabriel Kron [9] can be used. Kron proposed a methodology for the theoretical-set and algebraic synthesis of topologies in the form of the analysis method and tensor transformation of electrical networks.

An electrical network is convenient for a formalized description of structural connections of various kinds. Unlike other types of non-electric networks, an electrical network is always surrounded by a dynamic electromagnetic field created by itself, extending to infinity in all directions. By using the model of induction and self-induction in the branches of an electrical network, it is convenient to describe various intra-system connections between elements of the analyzed structure (organizational topology).

If we interpret the organizational-functional structure in the form of a multi-coil electrical network, tensor synthesis of Kron [9] can be used to synthesize a structure with specified properties.

The synthesis tensor is defined as follows:

For an existing multi-coil electrical network, there is a connection tensor $\mathbf{C}_1$ or in other words, a primitive network transformation tensor. A primitive network is a set of node pairs (an electric coil with an input and output point) and elementary circuits that form a specific network but are not connected to each other. The transformation tensor $\mathbf{C}_1$ is such that the corresponding electrical network functions in a defined manner:

$$Q(\mathbf{z}_1, \mathbf{i}_1, \mathbf{e}_1) = 0\,, \tag{6}$$

where $\mathbf{z}_1$ is the impedance tensor (electrical resistances of node pairs and network circuits, taking into account the resistance of inductance and self-induction in mutual influence of coils),

$\mathbf{i}_1$ is the current tensor;

$\mathbf{e}_1$ is the voltage tensor;

$Q(\mathbf{z}, \mathbf{i}, \mathbf{e}) = 0$ is the formalized representation of the network's behavior.

The impedance tensor of the output network $\mathbf{z}_1$ is derived from the impedance tensor of the primitive network $\mathbf{z'}_1$ using the transformation tensor $(\mathbf{C}_1)^{\mathrm{T}}$ [9]:

$$\mathbf{z}_1 = (\mathbf{C}_1)^{\mathrm{T}}\mathbf{z'}_1\mathbf{C}_1\,. \tag{7}$$

The synthesis tensor $\mathbf{C}_\sigma$ is derived to transform the impedance tensor of the output network $\mathbf{z}_1$ into the impedance tensor of another network $\mathbf{z}_2$, without changing the network's behavior:

$$\mathbf{z}_2 = (\mathbf{C}_\sigma)^{\mathrm{T}}\mathbf{z}_1\mathbf{C}_\sigma\,, \tag{8}$$

$$Q(\mathbf{z}_1, \mathbf{i}_1, \mathbf{e}_1) = Q(\mathbf{z}_2, \mathbf{i}_2, \mathbf{e}_2) = 0\,, \tag{9}$$

where $(\mathbf{C}_\sigma)^{\mathrm{T}}$ is the transposed synthesis tensor $\mathbf{C}_\sigma$.

The method of deriving the synthesis tensor $\mathbf{C}_\sigma$ depends on the form of the formalized description of the system behavior $Q(\mathbf{z}, \mathbf{i}, \mathbf{e})$. The result of this derivation is a compound tensor (a tensor whose elements are also tensors) [9].

In [10], a method for the formalized representation of conditions for computing regulations based on the transformation of a multi-coil network is provided. The condition is expressed as the equality of zero currents in intermediate node pairs of the network. Intermediate node pairs are interpreted as a group of parameters not determined in the process of computations. Using the synthesis tensor, a variety of electrical networks satisfying the formalized description of the system behavior $Q(\mathbf{z}, \mathbf{i}, \mathbf{e})$

$$\{\mathbf{z}_2, \mathbf{i}_2, \mathbf{e}_2\} \in \Omega^Q\,,$$

where Ω^Q – is the set of electrical networks obtained by transforming from the output network $\{\mathbf{z}_1, \mathbf{i}_1, \mathbf{e}_1\}$ using the synthesis tensor $\mathbf{C}_\sigma$, determined based on the formalized condition (6).

The choice of a specific network $\{\mathbf{z}, \mathbf{i}, \mathbf{e}\}^*$ from the variety Ω^Q is made using an established criterion K:

$$K_{\{\mathbf{z},\mathbf{i},\mathbf{e}\}^*} \succ \cdots \succ K_{\{\mathbf{z}_i,\mathbf{i}_i,\mathbf{e}_i\}} \succ \cdots \succ K_{\{\mathbf{z}_n,\mathbf{i}_n,\mathbf{e}_n\}}, \quad (10)$$

where n is the number of alternative electrical networks forming the set Ω^Q.

Thus, solving the problem of synthesizing structures with specified properties and addressing the adaptation problem of ASOM to conceptual changes in the control process can be implemented by introducing the following intermediate scheme (**Figure 7**).

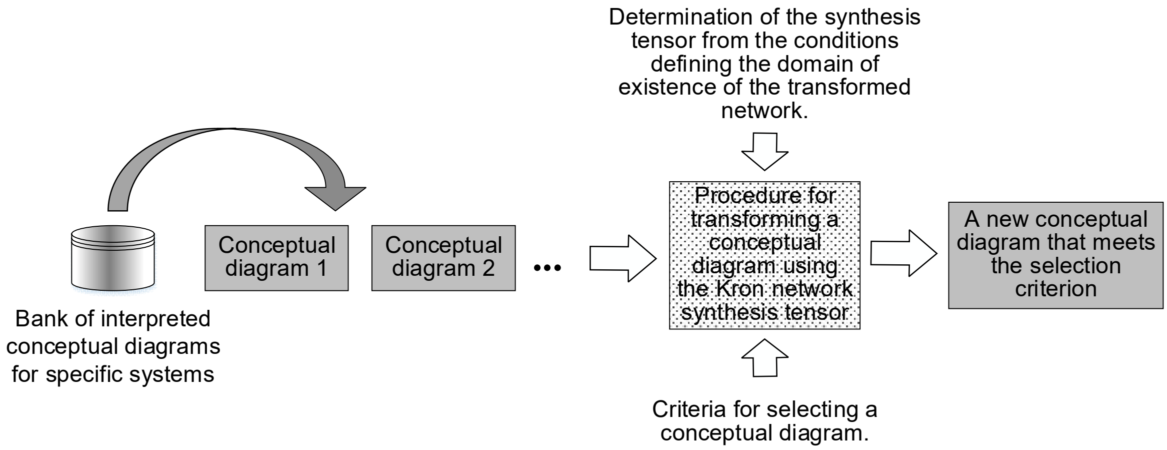

Figure 7: Scheme for Obtaining New Structures through Tensor Transformation of Pre-Interpreted Conceptual Schemes.

This scheme involves using a pre-interpreted bank of conceptual schemes for specific systems. Conceptual schemes are presented as multi-coil electrical networks with a corresponding formalized description of the desired behavior. In other words, defining the boundaries of the region of permissible state parameters or the existence area of the system.

In case of changes in the control loop that cannot be adapted by adjusting parameters within the existing control structure:

- the conceptual schema that is relevant to the initial state of the ASOM is retrieved from the database. Each conceptual schema has a formalized representation of its domain of existence and a pre-formed synthesis tensor to determine alternative schemes within that domain;
- using the synthesis tensor, a transformation of the original conceptual schema results in a set of alternative conceptual schemes;
- a criterion for establishing preference is formulated for comparing these alternative conceptual schemes;
- the criterion previously established is used to determine and choose a new conceptual scheme that best meets the needs of adapting the ASOM to the changes that have occurred in the management process.

Therefore, to address the adaptation problem in this context, progress should be made in the following directions: creating a database of pre-interpreted conceptual schemes for specific systems in relevant subject areas; developing a procedure for transforming conceptual schemes based on the use of the tensor transformation procedure for electrical networks.

Acknowledgements

Methods and models designed to solve adaptation and self-organization problems of Automated Control Systems (ACS) belong to the field of conceptual ontology design and transformation.

There are examples of developing adaptation models based on the application of:

- confinement models for ontological networks;
- algebraic systems apparatus;
- set theory apparatus of N. Bourbaki.

The main practical challenge in implementing these approaches is the issue of dimensionality. To construct adaptive control systems with complexity corresponding to real processes, the problem of creating a specialized language for interpreting (representing) the results of conceptual design needs to be addressed.

As an intermediate step in solving the adaptation problem, it is advisable to address the design task by transforming previously created conceptual schemes of specific systems.

The primary tasks for implementing this approach include:

- creating and accumulating a database (bank) of pre-interpreted conceptual schemes for specific systems;
- developing a procedure for transforming conceptual schemes based on the use of the tensor transformation procedure for electrical networks.

References

[1] B. Koo, W. Simmons, Algebra of systems: a metalanguage for model synthesis and evaluation, IEEE Transactions on systems, man and cybernetics (2009) – Vol. 39, N 3. – P.501–513

[2] N. Bourbaki, Eléments de Mathématique. XX. Première partie. Les structures fondamentales de l'Analyse. Livre I Théorie des Ensembles. Hermann et Cie, Paris, 1956. Première édition. 118 p.

[3] Yevgeniy V. Burov, "Conceptual Modeling of Intelligent Software Systems," Lviv: Lviv Polytechnic Publishing House, 2012. - 432 p.

[4] N.N. Mukhacheva, D.V. Popov, Ontologicheskie modeli i metody dlya upravleniya informatsionnointellektualnymi resursami organizatsii [Ontological models and methods for managing information and intellectual resources of an organization]. Vestnik UGATU – Bulletin of USATU, 14, 1(36), 2011, p. 123-135. URL: https://cyberleninka.ru/article/n/ontologicheskie-modeli-i-metody-dlya-upravleniyainformatsionno-intellektualnymi-resursami-organizatsii.pd

[5] Yevgeniy V. Burov, Volodymyr V. Pasichnyk, "Software Systems Based on Ontological Models of Problems," Bulletin of the National University "Lviv Polytechnic". Series: Information Systems and Networks: collection of scientific works. – 2015. – No. 829. – P. 36–57.

[6] Nikanorov S.P., Vybornov S.V., Ivanov A.Yu., Korshikov S.E., Kostyuk A.V., Kuchkarov Z.A., Mikheev V.V., Shalyapina S.K. Safety research. Ed. S.P. Nikanorova. / Concept, – 2006. – 624 p. https://vestnik.asomio.msu.ru › issue › download

[7] Nikitina N.K., Postnikov V.V. Development of a language for gender-structural explication of subject areas. / Development and conceptual design of intelligent systems: Sat. abstracts of reports and messages. Part 1. 1990. – pp. 70-73

[8] Ivanov A.Yu., Nikanorov S.P., Garayeva Yu.R. Handbook of system-theoretic constructs. Series "Conceptual Analysis and Design" Methodology and technology. – Concept, - 2008. – 314 p.

[9] Kron G. Tensor analysis of networks. John Wiley and sons, inc. London: Capman and Hall, Limited. New York. – 1966. – 720 p.

[10] Oleksandr G. Dodonov, Oleksii V. Nikiforov, Volodymyr G. Putiatin, "Implementation of Decision-Making in Organizational Management Based on the Ontology of Activity," Mathematical Machines and Systems, No. 1, Kyiv, IPMMS NASU (2022). ISSN 1028-9763 – P. 32 – 41.