FOREWORD

September 2025

The **main practical result** of the current stage of works in the field of Artificial intelligence is not only the creation of the next-generation intelligent computer systems ensuring effective interaction in solving comprehensive problems, but also the creation of a technological complex that ensures the rapid and high-quality building of such systems. This collection of scientific papers "Open semantic technology for intelligent systems design" is dedicated to these issues.

The development of this technological complex requires solving the following problems:

- clear identification of the logical-semantic level of intelligent computer systems, which abstracts from all possible options for the technical implementation of these systems (including the use of fundamentally new computers focused on their hardware support);
- development of an ontology for the design of intelligent computer systems and unification of the description of their logical-semantic models;
- ensuring the platform-independent character of the logical design of intelligent computer systems, the result of which is a unified description of the logical-semantic models of the designed systems;
- use of the methodology for component design of intelligent computer systems, which is based on a constantly replenished library of reusable components of these systems (reusable subsystems, knowledge base components, knowledge processing agents, user interface components)
- ensuring semantic compatibility of reusable components of intelligent computer systems and semantic compatibility of these systems themselves, as well as technologies for their design and support of subsequent stages of their life cycle.
- Each sc-agent has a corresponding specification, which is also stored in the same sc-memory as the constructs being processed;
- The ostis-system problem solver is treated as a hierarchical system of sc-agents. Two aspects of organization of such hierarchy are distinguished:
- Classes of functionally equivalent sc-agents, which have a common specification but are realized in general in different ways, are called abstract sc- agents.

participants in the process of decentralized information processing [1]–[3]. In modern works it is common to use the classification of self-organization mechanisms proposed in the works [12], [13]. The work [1] even suggests the idea of expediency of creating libraries of standard algorithms for decentralized computing and group management of networks of autonomous objects, including consensus protocols, protocols for leader selection, protocols for contractual networks, auction pro- tocols, protocols for common intentions, protocols for information exchange in the interests of maintaining situational awareness of participants in group behavior, Classes of functionally equivalent sc-agents, which have a common specification but are realized in general in different ways, are called abstract sc- Classes of functionally equivalent sc-agents, which have a common specification but are realized in general in different ways, are called abstract sc- agents. participants in the process of decentralized information processing [1]–[3]. In modern works it is common to use the classification of self-organization mechanisms proposed in the works [12], [13]. The work [1] even suggests the idea of expediency of creating libraries of standard algorithms for decentralized computing and group management of networks of autonomous objects,

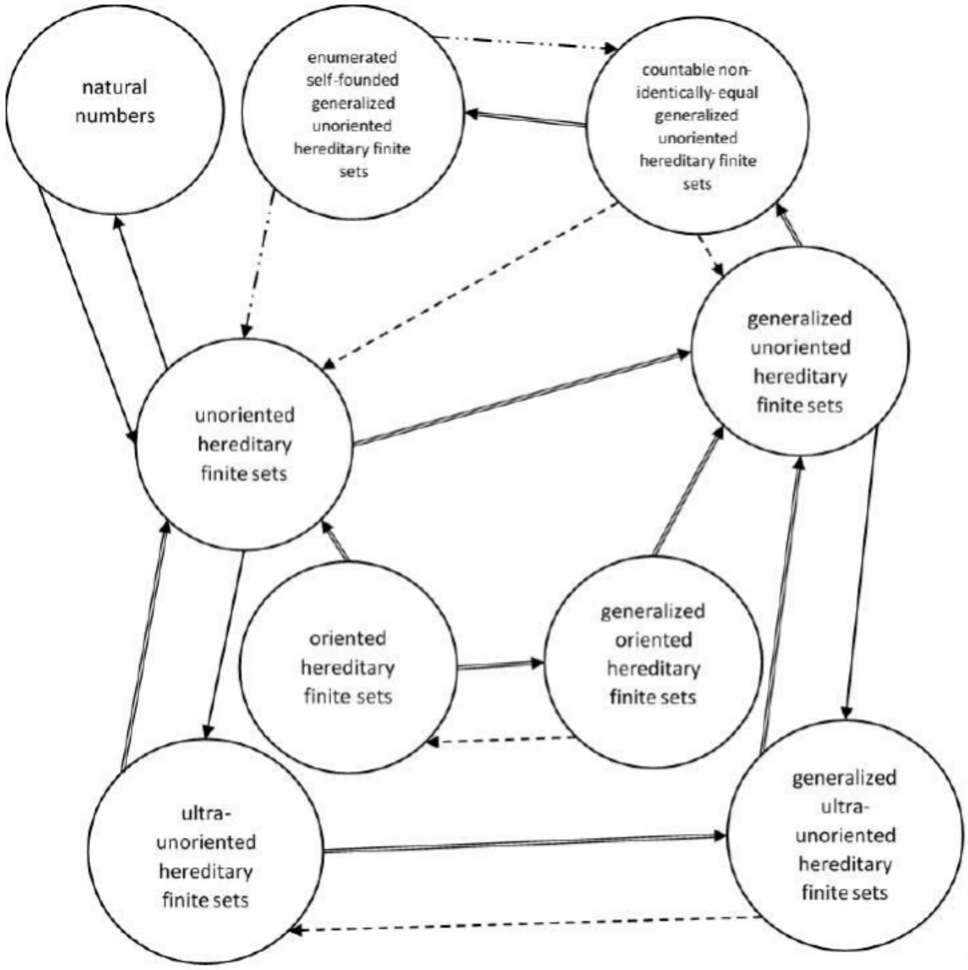

Figure 1: General finite structures classes

- Each class of logically atomic actions corresponds to a solver component (sc-agent) capable of perform- ing actions of the specified class;
- Sc-agents react to various events in the ostis-system memory (sc-memory) and communicate with each other only by specifying the actions they perform in this memory. Direct message exchange between sc-agents is excluded;
- Each sc-agent has a corresponding specification, which is also stored in the same sc-memory as the constructs being processed;
- The ostis-system problem solver is treated as a hierarchical system of sc-agents. Two aspects of organization of such hierarchy are distinguished:
- Classes of functionally equivalent sc-agents, which have a common specification but are realized in general in different ways, are called abstract sc- agents.

The above principles allow to ensure hybridity, modifiability of ostis-systems problem solvers, as well as convenience of their design and evolution [4], [11]. The key difference between the distributed ostis-system and the internal system of sc-agents within an individual ostis-system is the absence of a common memory storing a common knowledge base for all sc-agents and acting as a medium for communication between sc-agents. In general, as a means of communication between the participants of a distributed collective of ostis-systems can be used [7]:

- Shared unallocated (monolithic) memory, as in the case of sc-agents over sc-memory;
- Shared distributed memory. In this case, from a logical point of view, agents may think that they are still working on a shared memory, where the entire available knowledge base is stored, but in reality the knowledge base will be distributed among several ostis-systems and the performed transforma- tions will have to be synchronized among these ostis- systems;
- Specialized communication channels. Obviously, when solving a problem in a distributed collective of ostis-systems, there should be language and tech- nical means allowing to transfer messages from one ostis-system to another.

All of the above means of communication can be combined depending on the class of the problem solved, the knowledge and skills required for its solution, and the currently available set of ostis-systems. When solving a particular problem by a distributed collective of ostis-systems, in general the following "orga- nizational" subproblems related to the organization of the communication process of the ostis-systems themselves must be solved before proceeding directly to problem solving within the subject domain: The very idea of OSTIS Ecosystem and complex automation of human activity implies the need for self-organization of agents performing problem solving within the ecosystem. Currently, there is a large number of works devoted to the issues of self-organization of participants in the process of decentralized information processing [1]–[3]. In modern works it is common to use the classification of self-organization mechanisms proposed in the works [12], [13]. The work [1] even suggests the idea of expediency of creating libraries of standard algorithms for decentralized computing and group management of networks of autonomous objects, including consensus protocols, protocols for leader se- lection, protocols for contractual networks, auction pro- tocols, protocols for common intentions, protocols for information exchange in the interests of maintaining situational awareness of participants in group behavior, and many others.

$$E = mc^2$$

$$f(x) = \int_{-\infty}^{\infty} e^{-x^2}\,dx$$

At the same time, the proposed architecture of the *OSTIS Ecosystem* has a number of important features in comparison with traditional multi-agent systems, within the framework of which the existing self-organization mechanisms are implemented:

- Agents in traditional self-organizing systems usually have rather limited functional capabilities, a small amount of knowledge about the environment and rel- atively low reliability. This is especially pronounced in the works devoted to the so-called "swarm intelli- gence", where each agent in the system is maximally simplified, and the number of agents in the system grows accordingly. In turn, each ostis-system within the OSTIS Ecosys- tem is a complex computer system with an extensive knowledge base and functionality that allows such a system to solve a variety of problems from the relevant subject domain. The implications of this distinction are as follows:
- Traditional self-organizing systems are usually not considered as hierarchical structures, all agents are considered as autonomous units within the system, interacting with similar agents at the same level. The exception is the approaches to self-organization, which imply the allocation of special coordinat- ing agents or agents-arbitrators, whose task is to control other agents. Within the OSTIS Ecosystem framework, it is assumed to explicitly distinguish a hierarchy of agents corresponding to the hierar- chy of ostis-communities, besides, OSTIS Ecosystem agents are classified according to the role they play in the process of collective problem solving, in particular, corporate ostis-systems are distinguished. The hierarchical nature of the agent system makes it easy to develop and modify such systems by analogy with the hierarchical structure of problem solvers in individual ostis-systems [4], [11].
- In traditional systems, often all agents of the system, or at least a significant part of them, may be involved in problem solving. Taking into account the complexity of ostis-systems included in the OSTIS Ecosystem, such a situation is unlikely in the OSTIS Ecosystem and most often in the near future several ostis-systems, most often belonging to one ostis- community, will be involved in problem solving.
- Traditional self-organizing systems are usually con- sidered in isolation from the means of represen- tation of information processed in such systems, i.e. neither the form of representation of processed information, nor the semantics of processed infor- mation are explicitly fixed. An important advantage of OSTIS Ecosystem and OSTIS Technology as a whole is the orientation on unified and universal models of information representation, realized in the form of OSTIS Technology and a family of top-level ontologies built on its basis. This approach allows us to say