The Use of Intelligent Multi-Agent Systems Based on Ontologies in the Management of Transport Enterprises

Olha Tkachenko, Kostiantyn Tkachenko and Oleksandr Tkachenko

March 28, 2022

The Use of Intelligent Multi-Agent Systems Based on Ontologies in the Management of Transport Enterprises

Olha Tkachenko[1], Kostiantyn Tkachenko[1], Oleksandr Tkachenko[2]

[1] *State University of Infrastructure and Technology, I. Ogienko str., 19, Kyiv, 02000, Ukraine*
[2] *National Aviation University, Liubomyra Huzara ave. 1, Kyiv, 03058, Ukraine*

Abstract

The article discusses the principles of creating intelligent systems for situational management of transport enterprises in real time. A multi-agent approach to the construction of such systems associated with the use of ontological models is proposed. Situational management includes reaction to events, distribution and planning of resources, optimization of the solution, coordination with users, monitoring and control of the implementation of the constructed plan. An increase in management efficiency is associated with the achievement of consensus in the processes of joint decision-making by persons in a problem situation and participating in its settlement. In this context, in order to speed up and improve the decision-making procedure based on consensus, it is proposed to build multi-agent models of situations that provide support for consensus processes. The stages of work that need to be carried out on the way to creating consensus support tools are described. The article presents the experience of semantic modeling of the subject area (transport sphere) based on ontology. As a prerequisite for choosing the basic elements of ontological specifications, the cognitive ability of modeling subjects to distinguish between objects and detect relationships between objects is postulated. The general scheme for the use of ontological models is analyzed, its organic orientation towards the integration of heterogeneous knowledge is stated, and the mechanisms for managing models necessary for this are outlined.

Keywords [1]

Ontology, multi-agent technologies, agent, actor, intelligent system, decision making

1. Introduction

Classical scientific methodologies turned out to be poorly adapted to work with complex, ambiguous, inaccurate and contradictory information, which is typical for modern applied problems in the transport sector. Formal mathematical modeling is of limited use due to the fact that semantic models of reality are either *content-descriptive* in nature, or include both content-descriptive and formal-mathematical components. New opportunities for modeling are opening up thanks to the development of intelligent technologies.

These technologies advance formal ontologies to the role of basic semantic models [1, 2]. The use of intelligent systems is effective only with a coordinated representation of *the subject* of the acting subject – *the actor*.

Systematization, development and use of such representations constitute the content of the ontological approach in modeling. The article describes the essence and possibilities of such modeling in the subject area (SA) – the transport sector (transport companies, logistics centers, vehicle manufacturers, etc.). An SA in the transport sector can be represented by a main SA and auxiliary SA.

The set of objects considered in the context of any situation or task forms its SA. Links between objects define relations in pro: unary are interpreted as *properties of objects* (functions of objects);

CMIS-2022: the Fifth International Workshop on Computer Modeling and Intelligent Systems, May 12, 2022, Zaporizhzhia Ukraine
EMAIL: oitkachen@gmail.com (A. 1); tkachenko.kostyantyn@gmail.com (A. 2); aatokg@gmail.com (A. 3)
ORCID: 0000-0003-1800-618X (A. 1); 0000-0003-0549-3396 (A. 2); 0000-0001-6911-2770 (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

arbitrary arity, or relations proper, describe various *associations of objects*. This makes it possible to present the SA as a *semantic network*.

This network has two subnets: "*classes*" and "*instances*". The first subnet has vertices (class vertices and property vertices) and arcs (the "is a view" arcs connect the class vertices, and the "is a part" arcs connect each property vertex to one and only one class vertex). This subnet defines the ontology of an SA: a set of concepts about the properties of objects and varieties of objects in the context of available properties. The "instances" subnet is a denotative object model of the SA, in which the ontology of the SA determines the types of vertices (images of SA objects) and arcs (images of links of SA objects).

When solving problems, the ontology represents the theory, and the denotative model concretizes this theory in relation to the actual situation in the modeled SA [3, 4].

Many of these situations lead to:

- availability of separate specifications for the ontological and denotative components of the SA object model (for reusing the ontology when modeling situations in the SA);
- using the modeling context as an actual pair {ontology of SA, denotative model of SA}.

The semantic models of the SA (ontologies and denotative models) are homogeneous. For a denotative model, this is an SA, which is described by the ontology of this SA. For ontology, this is the SA, which owns the concepts of "object class" and "property". Metaontology (where these concepts are described) should be semantically closed, describing itself.

2. Use of ontological models

The use of ontologies provides the process of solving applied problems with systematicity and interdisciplinarity (due to the multi-modeling and integration of models inherent in the ontological approach).

The central figure in the scheme of construction and use of ontological models is the actor. The most important information flows converge to it, from empirical data on SA to interpretation of simulation results. He participates in the iterative formation of all the necessary modeling specifications. The following models are used in the scheme:

- $O^K{}_D$ – ontology – the result of the conceptualization of the SA "$^K D$".
- $OM^{EX}{}_D$ – exogenous object model of SA "*D*" (based on ontology of SA), description of situations in SA.
- $DM^T{}_D$ is a denotative model of the "technological" SA "$^T D$". This model serves as a specification of the task solution scenario, a description of the impacts on the $DM^{EX}{}_D$ model, as a result of which it must acquire the properties necessary for a particular user.
- $DM^{EX}{}_D$ – exogenous denotative model of SA – interpretation of knowledge recorded in $DM^T{}_D$ in relation to $DM^{EX}{}_D$. The result valuable for the actor is either $DM^T{}_D$ or a sequence of actions on $OM^{EX}{}_D$, transforming it into $DM^{EX}{}_D$, or both.

The expansion of variants of multi-model structures should be carried out taking into account the following features of the semantic modeling of the considered SA:

- conceptualization is characterized by the possibility of formation and existence of several ontologies ($^K D$-models) of the main SA;
- with the help of ontology one can build many different denotative models of the main SA;
- for any basic SA, it is possible to specify a set of actual tasks (a set of options for solving a certain task), which leads to a set of "technological" ($^T D$) models;
- to solve real problems, along with the modeling of the main SA, it is required to involve knowledge from several auxiliary SA.

Knowledge integration in ontology-based semantic modeling is closely related to the composition of formal ontologies (*ontology merging*). In [5 - 7], methods for combining ontologies are described.

In a multi-model environment, you need to support model management, which can provide an implementation of:

- simultaneous manipulation in solving the problem by several denotative object models of missile defense, which is described by one ontology;

- combination of several views on the main SA, when in solving a problem it is necessary to operate with object models built according to different ontologies of this SA;
- sharing of object models from different SAs;
- solving the problems of building or changing the ontology (^{K}D-models), as well as ^{T}D models based on the corresponding meta-models;
- organization of interrelated "computational" experiments for the implementation of alternative and evolutionary studies of SA.

To realize these possibilities, it is sufficient to use two "circuits" (task and design) for managing models at the "computation" stage. In the task circuit, by switching the modeling contexts – the corresponding definition of a pair (^{K}D -model, *D*-model) - it is possible to work with several *D*-models from different SAs. At the same time, the composition of admissible SAs must be controlled (the ^{T}D-model must be competent in each of them) and the correctness of the modeling contexts.

The design loop is optional and is linked to the "calculation" macro control. Management consists in conducting interconnected series of experiments with simultaneous structuring of information about the performed "calculations" in the form of an acyclic graph.

3. Use of multi-agent technologies in decision making

Achieving consensus in decision-making is carried out by joint discussion of the current situation (problem). A decision that determines a productive way out of any situation should be developed in the course of a dialogue between people who are direct participants in the situation.

Consensus can be viewed not only as agreement or unanimity (*understanding*), but also as a "unity of opposites" (reciprocal understanding): if one person has the ability to satisfy the need of another, then an agreement on the provision of an appropriate service can be interpreted as reaching a consensus between them. It is in the second sense that these concepts are used in [8].

Multi-agent technologies make it possible to automate the processes of resolving conflicts and finding balances of interests. Deciding to resolve a situation by consensus is a "poorly convergent" procedure, which can sometimes "diverge" if people no longer understand each other. In order to solve (at least partially) this problem, negotiators must agree on the principles on the basis of which they will seek compromises, and put them into the created multi-agent model of interaction [9, 10].

As such a universal principle, the concept of a network of needs and opportunities (NO-network) is proposed, in which agents of needs and opportunities of participants in the decision-making process to resolve problem situations are identified, who have access to a formalized multi-agent model of situations.

NO-network agents may have conflicting interests, preferences and goals, the progress towards which (if we talk about the economics of the enterprise) is supported by bonuses and is punished by fines in the virtual market. In the course of competitive and cooperative actions of agents of the NO-network, a solution to the problem is formed in the form of an agreed action plan for the participants.

The plan is considered to be built and consensus found when no agent can improve the situation, even if he is not satisfied with the decision. Let us explain what has been said, based on examples of real-time resource management (by transport enterprises, trucks, etc.), in which, in order to create tools to support consensus, you need to go through:

Stage 1. Creation of an interactive "smart environment" of adaptive planning:

- in the multi-agent system of adaptive planning, a general action plan is built, which is formed and agreed upon by the agents of the NO network acting on behalf of all participants (for example, for a transport company - dispatchers, managers, drivers, etc.);
- the created plan is not a static data structure, but an invitation to a dialogue to coordinate positions, since at the same time important indicators are calculated for the enterprise as a whole and for each participant individually, "bottlenecks" become visible (indicators of downtime, profits, delays, etc.);
- each participant can ask questions and make a move – propose to redistribute orders for resources by changing the sequence of operations, or come up with a proposal that improves the performance of the system;
- the proposal made will change the situation for other participants, and they will react within

the framework of the general system of restrictions.

Stage 2. Creation of an intelligent system for the development, coordination, adoption and execution of decisions:

- extending the model and enabling agents to identify bottlenecks in the plan and come up with proposals for their elimination;
- providing the parties with the opportunity not only to accept or reject the plan, but also to enter into a dialogue to improve it, when the performer of the lowest level (driver, worker, etc.) can accept or reject the task, break the task into subtasks, indicate the actual execution time operations, etc.;
- with this approach, the lower-level performer becomes a full-fledged and full participant in decision-making processes, since it often turns out that he knows more (but more narrowly) than fellow managers;
- users should be able to enter into the system data on the fact of execution of operations (delays, transfers, cancellations of operations, etc.).

Stage 3. Creation of a self-learning intelligent system for the development and coordination of decisions:

- the use of SA ontologies, separated from the program code of the system, gives users the opportunity to supplement and develop the functions of the system without reprogramming it;
- the system gets the opportunity to self-learn in the course of work;
- the need to expand the knowledge base of the system (for example, a worker may report that the operation cannot be performed at the specified time, or the dimensions of the product do not allow parts to be connected and new operations are needed, which then the technologists will add to the technical process by changing the knowledge base).

4. Situational management and multi-agent technologies: search for consistent solutions

The challenges of the global economy associated with competition, increasing complexity of tasks to be solved, uncertainty and high dynamics of changes in supply and demand, force enterprises to look for new approaches to increase productivity and efficiency in the use of their resources: human, financial, material, and others.

“Program-targeted management” is increasingly being replaced by “situational management” [13], which denies formal adherence to previously created rules without analyzing the essence of the situation and involves a detailed analysis of all the features of the situation, the collective development of new rules for making a decision on the context of the situation.

For example, when planning the schedule of trains of the Hyundai class, in the event of unforeseen situations, the dispatcher must rely on the rule "always let the Hyundai go ahead." But the complexity of the task of compiling and correcting the schedule is so great that sometimes the dispatcher, acting according to this rule, may not calculate the consequences of his decision, and the delayed freight train of the lowest priority will quickly create a huge traffic jam from other trains and slow down the movement of several other Hyundais at once, whose interests were not taken into account.

The modern solution to the problem of increasing the productivity and efficiency of enterprises is to:

- transition from centralized hierarchical structures to open, distributed, network organizations;
- organization of team work, rather than decision-making by one "boss";
- using knowledge, competing opinions and beliefs as the main tool for making decisions;
- respect for other people's opinions, not ignoring them;
- focus on unlimited payment based on the final result.

To realize this shift towards better enterprise management, businesses need intelligent systems that can:

- work in real time and respond to events;
- change work plans of employees;

- optimize plans and interact with all stakeholders to develop and agree on decisions;
- monitor and control the implementation of the plan;
- in case of discrepancies between plans and reality, actively initiate the creation of a new plan.

At the heart of such an enterprise management process is communication aimed at developing agreed decisions. The control system itself should be built on the basis of communications, which can be implemented on the basis of multi-agent technologies, in which the solution to any problem is sought in the course of dialogue (communication) between agents who defend their interests, but are able to make mutual concessions. A new generation of intelligent systems gets the ability to communicate with users and other systems.

If agents are built on the basis of rigid logic and clear algorithms that are close to formal models, then the operation of the system is built on the basis of interactions and negotiations between them. The main result is the construction of a coherently operating enterprise working towards the ultimate goal in the consensus of opinions of all participants.

4.1. The essence of situational management

The crisis of modern management most clearly manifests itself in conditions when the world around is becoming more and more uncertain, unstable and rapidly changing. Situational management involves stimulating and encouraging new ideas, removing restrictions on the areas of activity of the unit and their employees, recognizing the role of knowledge in project management, loyalty to trial and error, even failure, teamwork, result orientation, development of systems of remuneration for work.

The implementation of this approach breaks the existing stereotypes in the company's management and leads to the emergence of fundamentally new network forms of enterprise organization. A streamer can be considered an enterprise, the basis of which is a multi-level network of business units, self-organizing to solve problems on the basis of resource centers.

The general scheme of situational management assumes the following:

- any member of the team discovers a problem and fixes the situation in which the problem manifests itself (a new order has arrived, a machine has broken down, etc.);
- the duty of a team member is to describe the situation, formulate the problem and “throw” it on the “common table” for discussion;
- any team members who understand the problem or know approaches to solving it, and who are ready to join the solution, respond to the event associated with the appearance of a problem;
- looking for knowledge, documents and materials, examples of solving such problems;
- the leader-actor, who is ready to take on the solution of the problem, acts as a moderator of the teams;
- any member of the team can give his proposal on how to solve the problem, add or clarify the conditions for solving the problem, restrictions or preferences, invite new members;
- other team members give their proposals in response, which may compete with or complement others;
- when contradictions appear, a decision-making point is searched for that caused this conflict, a return is made, mutual concessions are made and a new decision is made in the interests of the organization, after which the search for a solution continues;
- on the “common table” the essence of the problem situation is clarified and the solution of the problem begins to form (for example, a sketch of the object design, an outline of an action plan, etc.);
- several processes can develop simultaneously to find alternative solutions to a problem;
- when the problem is solved and there are no more proposals for improving the solution, then the process of finding and agreeing on solutions stops and the process of its implementation begins.

Consider the design and operation of an intelligent system of the Smart Solutions class for situational management of enterprise resources.

4.2. Smart Solutions system: structure and functions

To solve the problems of any enterprise, the architecture of the enterprise resource management system is proposed, which contains the following subsystems:

- Pattern recognition.
- Adaptive scheduler.
- Scene constructor.
- Ontology editor.
- Modeling subsystem.
- Evolutionary design.

Pattern Recognition – recognizes situations that arise when events occur (for example, new applications) and develops recommendations for the allocation of enterprise resources and planning. This subsystem supports automatic learning to improve planning efficiency.

For example, this subsystem can recognize the fact that a certain request enters the system at a certain frequency. This will allow you to pre-book certain resources for its implementation. If the request is not received at the expected time, then the subsystem can generate a message to the manager with a request to clarify the situation with the client and release the resources previously reserved for the request.

Adaptive scheduler – processes the flow of incoming events (for example, receipt of applications, introduction of new resources, failure of resources, etc.) As a result of event processing, a plan for the distribution and use of resources is formed and the plan is dynamically changed. The scheduler of one division of the enterprise can interact with the schedulers of other divisions, transmitting events to them and coordinating the decisions made.

Scene Builder – allows you to edit the initial network configuration and define all the parameters of the enterprise resources. In this case, it is necessary to import data from various sources. This subsystem uses a common knowledge base (ontologies) that describes the activities of the enterprise and expands with the development of the business using the ontology editor.

Ontology Editor – allows you to maintain and modify the general enterprise ontology that describes the SA knowledge model. This model is used in the scene designer to describe enterprise business configurations. Ontology contains basic concepts and relationships between them, represented in the form of a semantic network. Ontologies initially emerged as a convenient means of knowledge representation for creating a new generation of Internet-systems (*Semantic Web*), but recently they are increasingly used in various modeling and decision support systems [11, 12].

The ontology of the enterprise in solving problems of resource allocation contains a description of:

- classes of objects and relations of the enterprise, including types of orders, classes of resources, etc.;
- classes of relationships (a resource is reserved for a request, a request is fulfilled by a resource, etc.);
- classes of operations of the business process of the enterprise, describing the life cycle of the application;
- classes of attributes of objects and relations.

Ontology makes it possible to separate knowledge of the enterprise's SA from the program code of the system, which creates the basis for further development of the system and increasing its function without reprogramming.

Modeling subsystem – carries out modeling of situations on the principle of "what if?". At any time, the current state of the enterprise and the work plan for the near future can be loaded into this system in order to simulate what will happen in one case or another (for example, signing a contract with a new client, increasing the volume and range of resources, selling resources, etc.).

Evolutionary design – develops proposals for improving network configurations in terms of increasing or decreasing resources, changing the geography of resources, etc.

Let us consider in more detail the key subsystem of the adaptive planning system and its main components:

- *Executing system* (*Run Time Multi-Agent Execution System*) – a subsystem that ensures the execution of agent programs during the transition from one state to another (agent manager) and the transmission of messages between agents, in which the agent receives a "quantum" of time for event processing and returns control to the dispatcher to promote other agents. The Agent Inspector and the Agent Log are also part of this system.

- *Event Queue* - a subsystem that provides the accumulation of events and their processing. Since the system is event-driven, each event has a pointer to the time of its arrival. The system provides regulation of the order in which events enter the system for processing, when the priority of the event is taken into account.
- *World of NO-network agents* / virtual market (virtual world of RDN – virtual market) – place of work of NO-network agents. Under the control of the system, agents can be created, destroyed, exist, receive and transmit messages, access the scene to read information, write information to the scene, etc.
- *Scene of the World* – a data structure that:
    - contains a formalized model of situations in the outside world;
    - can be refined through ontology.

The scene of the world is corrected by events in order to ensure the adequacy of the system in the perception of situations in the surrounding world. The scene contains a description of the situation, which gradually transforms into an event-based solution to the problem, and contains, for example, a new action plan for the user (truck driver, foreman and worker, etc.).

- *Designer of ontologies, models and scenes* (*Ontology Editor*) – allows you to correct the initial scene or dynamically make changes to it.

• *Ontologies* – data structures representing the SA knowledge models used to build models of initial situations and correct them. There are basic ontologies that can be supplemented with concepts, relations and special extensions for each enterprise that are specialized for SA.

- *Scheduling libraries* (*Basic Virtual Market & Domain-Specific Extensions*) – contain components that ensure the operation of classes of NO-network agents and their negotiations in the virtual market (for example, conflict detection, order overlap zones determination, time shift calculation, etc.), access to a scene containing a formalized model of situations, processing of criteria, preferences and restrictions of agents, and other functions.
- *Data Base* – allows you to save the original and intermediate scenes, as well as scenes with the result of solving the problem.
- Specialized components and integration with third systems (*3^rd^ Party & Integration Components*) – components that allow you to perform additional functions for the Software (eg, calculation of distances on the map for trucks, etc.).

4.3. Modeling the NO-network

The proposed system is based on the use of NO-network agents operating in the virtual market of the system.

This approach allows you to design a multi-agent system as a set of agents of "planner-optimizers" with their own goals and objectives, criteria, preferences and restrictions for decision-making, but which, at the same time, can cooperate and compete in making decisions in the interests of the common whole uniting them (section, workshop) enterprises).

In a NO-network, agents can receive and apply the roles of needs (orders) and opportunities (resources). For example, each “truck” agent knows what his route is, where he is now, what kind of cargo he is loaded with, what is his plan of action in a given situation, etc. By receiving offers from different "trucks" (opportunities), "order" (need), the agent can decide which one suits him best. But the “truck” itself can generate a new “need” by specifying the “orders” it needs at the current time.

A NO-network may consist of agents of needs and opportunities seeking to find each other, break existing ones and establish new ones.

For example, for a freight transportation company, the N0-network model may include agents:

- "customer" and "order";
- "truck" and "cargo";
- "travel itinerary";

- "shop" and "warehouse";
- "driver", etc.

At the same time, the “order” is constantly looking for the best “truck”, and the “truck” is constantly looking for the “order”, “route”, “driver”, etc.

Decision-making by several agents and the establishment of links between them to solve problems that continuously arise when each new event arrives, causes a change in the conditions for the functioning of other agents and determines the process of system self-organization, leading to a restructuring of the schedule in response to the event.

The knowledge on the basis of which agents make decisions is separated from the program code and stored in the system ontology, which is provided using a special ontology constructor, models and scenes. A specific resource situation is described as a scene linking specific instances of objects (client's company name, truck driver's name, vehicle number, etc.). Typical configurations can be saved as models (for example, workshops - placement of equipment and workers).

The activity of all agents of the PV network causes multilateral negotiations in the virtual market. At the same time, a feature of the approach is that each agent is considered as a state machine that returns control to the dispatcher after each cycle of negotiations.

Each agent tries to achieve his goal and for this he enters into relationships (connections) with other agents (an order is booked for a truck, a truck for a driver, etc.), which can be reviewed by agents as a result of identifying and resolving conflicts under the influence of coming from outside or generated within the event system.

Conflicts generated by events (for example, failure of a truck) can be resolved by agents of orders and resources through negotiations and mutual concessions aimed at reaching compromises acceptable to all. Conflict resolution can cause a whole chain of rescheduling operations, the depth of which may be limited by the allowable response time or other factors.

When a new order arrives in the system, its agent is created, which comes into contact with resource agents to find the best place for it. If the most suitable resources are already occupied, they can begin to offer orders placed on them earlier to look for new placements. If suddenly the selected truck becomes unavailable later (breakdown, accident, etc.), then its agent must find all the orders that are currently planned for placement in this truck and inform them about the unavailability of the resource. These orders are activated and start looking for other trucks, which allows you to quickly, flexibly and reliably reschedule travel routes. The result is considered achieved and the system terminates when no agent has more opportunities to improve its state.

The developed approach integrates many modern ideas of optimal planning, creating an environment of algorithms (agents). Agents can remember and avoid bad decisions by using their memory, inform each other about intermediate options, make decisions randomly when options are close, stop searching if there are time limits for making decisions, etc.

At the same time, by presenting the task in a form close to natural, the decision-making logic of the system becomes more transparent for programmers and operators, which reduces the system development time, and also makes the system results understandable to the user.

5. Ontologies and multi-agent approach to system development

Operating principles of the multi-agent adaptive planning subsystem [13]:

- for each task and executor, a software agent is created that receives requirements, preferences, and planning constraints;
- the agent starts planning by searching for the resources he needs in the scene, which describes the current situation in the department, namely, which employee performs which tasks;
- if suitable resources (performers) are occupied, the conflict is fixed and negotiations are started to resolve it;
- in the course of negotiations, options are possible: a new task will go to a less suitable resource (executor), the previous task will go away or move;
- after solving his problem and building a plan, each agent does not stop and continues to try to improve his situation. An intelligent transport enterprise management system is implemented as a multi-agent system.

Agents work on behalf of orders, projects, tasks, divisions, products, transportation services, employees, software components, documents, and so on.

Table 1
Main classes of agents in management system

Agent Name	Agent Description	Attributes
Order	The order is looking for the best opportunities for implementation within existing or new business centers and knowledge centers, interests and competencies of performers	Content, cost, term and other characteristics
Transport service	Attempts to provide a transport service, taking into account the specified criteria, technological and business processes, performers	Essence, cost, term, performer and other characteristics
Project	Attempts to organize and execute the project taking into account the specified criteria, preferences and constraints, technological and business processes, the presence of performers	Ontological descriptor of the project content, staff, budget, deadlines, results
The organization	Tries to achieve and improve the result according to the specified criteria, monitors the situation, changes the strategy for the selected agents, identifies "bottlenecks", fixes the result	Type of organization, composition of the organization, criteria and action strategies, expected results and current indicators
The employee	Wants to be as busy as possible according to the profile and receive bonuses for quality, productivity, etc. He is trying to improve his competencies to achieve a higher level of qualification and remuneration.	Organizations to which it belongs, competency profile, work plan, current task, skill level, salary, program authorship, etc.
The software component	Wants to be used as much as possible, if necessary – to be finalized. Takes into account the connection with other components, documents, tests, etc.	Purpose, application, ontological descriptor, author, use, connection with other components, cost.
Technological or business process	Wants to be performed in the best possible way as a chain of individual operations (tasks) necessary to complete project orders.	The composition of the product (order, service), the list of operations and the graph of communication between them, performance criteria, terms and cost.
Activity	Searches for the best workers and components, taking into account the preferences and constraints of the project and the relationship with other activities.	Competence and qualifications of the performer, duration, connection with other operations
Result	Attempts to be created as a result of a project (order, transport service) from ready-made or new components	Result characteristics
Other PC Network Concepts	Customers and partners, events, leaders, etc	It is expected to be supplemented as the system develops

5.1. Architecture and system components

The central component of the system is the application server, which performs adaptive event-based scheduling and interaction of all subsystems, processes data and provides differentiation of user access rights in the system. In the systems being created, several main specialized automated workstations can be provided: the head of the company and project executors, dispatchers and drivers, foremen and workers.

The workstations of managers of all levels are designed to manage orders, projects and tasks, including the distribution of tasks, work planning, monitoring and control of results, adjusting plans, etc.

The adaptive scheduling subsystem runs continuously on the server side and can use any relational database.

The main components of planning are: dispatcher agent, messaging service, agent lifecycle support service, agent creation and removal services, support for the communication protocol between agents, and plan scene support. When data is processed in the system, events occur that are processed in real time by the planning subsystem, including, in particular:

- changing project parameters;
- emergence of a new task;
- changing the parameters of tasks (planned deadlines, etc.);
- change in the state of resources (appearance of a new resource, change in the availability of a resource, change in the assessment of an employee's skills);
- the fact of task execution (marking the fact of completion, indication of user preferences).

The systems under consideration continue to work even when a solution is found and there is no way to improve the results. The meaning of this is not only the expectation of a new event in order to constantly keep the plan up to date, but also the possibility of improving the resulting solution by using different agents.

As a result of the work, the system generates a plan, which is built in the course of negotiations between agents that identify and resolve conflicts.

The resulting plan may not please the user, and then the user can change and reassign resource orders, which can immediately trigger the process of reallocating other orders to resources; but the user can freeze parts of the schedule that seem to him to be built in the right way.

If the built plan suits the user, he can accept it for execution, and then the system will switch to the mode of monitoring the execution of the planned plan. When the time of the next operation approaches, the system will ask the user if he is ready to perform the required action, or if the task should be rescheduled. As a result, the coordinated work of the employees of the enterprise is ensured, actually moving to work in the consensus mode.

Each *actor* reads the initial information about the task and context, translates it into a system of its own representations, solves the task based on its knowledge base, but with given preferences and restrictions, and returns the result to the scene context.

The resulting decision is made by other actors, the process continues and control is transferred to the next actor, or there is a conflict between the goals, preferences and constraints of the actors. In this case, the process stops and direct negotiations between the actors begin, aimed at resolving the conflict through mutual concessions.

Ultimately, a solution to the problem is formed, agreed with all the actors. Let's consider an example of using the proposed conceptual model.

Example 1: designing a new complex product. Let it be required to solve the problem of designing a new car of the middle class. In this case, the virtual "round table" should gather, in particular, such actors: the chief engine designer, the chief technologist, the chief designer, the chief financier and a number of other car development specialists.

Let the basic requirements for the parameters of a new car (overall dimensions, technical and driving characteristics), the maximum price, the maximum amount of possible investments, etc. be known and set. In this situation, the chief designer of engines can start solving the problem, who will offer his own ready-made engine design.

This design activates the chief technologist, who will see that the production of the engine will require new technological equipment and will offer the best equipment of one of the firms. At the same time, the cost of the project will increase, which will be immediately reported by the agent of the financier, who will show how much the level of possible investments is exceeded.

This signal will stop the process and cause the need to reconsider decisions - as a result, either a cheaper production line will be found, or another engine will have to be selected that can be assembled on existing equipment. The design process will continue until all participants have worked out their wishes, taking into account the current restrictions, and the new car has been worked out in all details.

At the same time, each actor can be included in several active teams ("virtual tables"), showing the main tasks to be solved and the contours of the enterprise's management decisions at any given time.

5.2. Intelligent enterprise resource management systems

Consider the use of a distributed intelligent system for transport enterprises. For them, the transition to real-time decision-making provides an increase in the efficiency of the use of shop floor resources by reducing downtime or shortages (expensive equipment or highly skilled workers).

The proposed approach is implemented in an intelligent system, which is a network of intelligent subsystems for managing individual departments (workshops of a transport enterprise). The system is focused on the use of adaptive planning, which takes into account information coming from users, existing enterprise management systems, enterprise equipment (sensors, meters, controllers, etc.), tablets of masters or touch screens of workers. All this can cause a real-time real-time redistribution and re-negotiation of resources.

Consistency of decisions is ensured by a multi-level adaptive network of systems using the “peer-to-peer” principle.

Proposed a distributed intelligent system for real-time resource management is created as a multi-level adaptive network of intelligent shop floor control systems, including:

- the system of strategic planning for the enterprise as a whole for the long planning horizon (1-2 years);
- operational management systems that provide coordinated decision-making at all levels of enterprise management and a quick, adaptive response to unforeseen events for the shorter horizon (2-3 months).

The strategic planner (one of the components of a distributed intelligent system), having built the first version of the plan, will pass it "down" for approval to the operational planners of the shops, who, having planned their work autonomously, will begin horizontal negotiations to agree on their operational plans. Successfully created "downstairs" plans will be brought to the strategic planner.

There is a typical situation from the real life of an enterprise, when the work plan becomes obsolete before its implementation begins (for example, a new order has arrived, payment for an already accepted order has been delayed, a new task has appeared, a worker has gone on vacation, etc.).

The proposed intelligent system will solve the problem of constantly "outdated plans" by adapting the connected network of operations in the general field of enterprise resources through their vertical and horizontal interactions, the regulations of which will be developed during the project.

The creation of such intelligent systems in the provision of transport services and the functioning of transport enterprises, which allow working in conditions of uncertainty and high dynamics of change, when neither the number of orders nor the number of resources is known in advance, allows replacing the components of such traditional ERP-systems (*Enterprise Resource Planning*).

The developed intelligent systems can be used both completely autonomously and integrated with existing enterprise management automation systems, which significantly expands the sales market for the final product.

The multilevel structure of a distributed network of interacting intelligent systems for managing organizations will make it possible to implement the proposed approach to managing transport enterprises by creating an architecture that is fully displayed in the structure of the enterprise itself.

The proposed approach provides such advantages of the developed intellectual system as improving the quality and efficiency of resource planning solutions, openness to the phased connection of plans for new units, high efficiency, flexibility and performance, reliability and survivability, scalability and integration of the overall resource management system, reduces costs and risks with such a system.

6. Conclusion

The paper describes an approach to the creation of intelligent systems for situational management of transport enterprises using ontological models of SA.

The proposed multi-agent approach to the creation of the considered intelligent systems is associated with the transition to an autonomous resource management cycle, including:

- reaction to events;
- allocation and planning of resources;
- solution optimization;
- coordination of the solution with users;
- monitoring and control of the implementation of the developed plan of operations (actions);
- replanning in case of discrepancy between the plan and the fact.

The ontological approach gives a fairly clear view of the composition, purpose and structure of the situation model, as well as the mechanisms for managing models in solving practical problems of the transport sector (transport enterprise).

This covers most of the problems of developing semantic models, in particular:

- organization of the knowledge system of the considered SA and ways of solving problems in this SA;
- planning of problem solving.

The solutions to these problems turn out to be highly unified, and the models used are homogeneous.

This approach distinguishes the proposed intelligent system from existing resource management systems, in which tasks and resources are considered known in advance and do not change during planning or execution.

The complexity of the considered intelligent systems is offset by the additional opportunities that they open up for transport enterprises in terms of ensuring the efficiency and consistency of decisions, thereby improving production efficiency.

7. References

[1] C. List, Levels: descriptive, explanatory, and ontological, 2018. URL: http://eprints.lse.ac.uk/87591/1/List_Levels%20descriptive_2018.pdf.

[2] A. Gelfert, The Ontology of Models, In: Magnani L., Bertolotti T. (eds) Springer Handbook of Model-Based Science, Springer Handbooks, Springer, Cham, 2017.

[3] Kamran Munira, M. Sheraz Anjumb, The use of ontologies for effective knowledge modelling and information retrieval, 2017. URL: https://www.sciencedirect.com/science/article/pii/S2210832717300649. doi: 10.1016/j.aci.2017.07.003.

[4] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, E. Zavitsanos, Ontology Population and Enrichment: State of the Art. In Knowledge-driven multimedia information extraction and ontology evolution. LNAI 6050. Springer-Verlag Berlin. (2011). pp. 134-166.

[5] V. Lytvyn, V. Vysotska, D. Dosyn, O. Lozynska, O. Oborska, Methods of Building Intelligent Decision Support Systems Based on Adaptive Ontology. Second International Conference on Data Stream Mining & Processing (DSMP). 21-25 Aug. (2018): 145-150.

[6] E.M. Sanfilippo, Feature-based product modelling: an ontological approach, International Journal of Computer Integrated Manufacturing, volume 31(11), 2018, 1097-1110.

[7] S.E. Greger, S.V. Porshnev, Building an ontology of information system architecture, Fundamental Research, № 10, 2013, pp. 2405-2409.

[8] J. Sowa, Building, Sharing and Merging Ontologies. URL: http://www.jfsowa.com/ontology/ontoshar.htm .
[9] L. Zhou, Ontology Learning: State of the Art and Open, Information Technology and Management. No 8(3), 2007. pp. 241-252.
[10] K. Highnam. The Future of Multi-Agent Systems Research, 2020. URL: https://medium.com/@kwhighnam/the-future-of-multi-agent-systems-research-5aef163f1d69.
[11] A. Dorri, S.S. Kanhere, R. Jurdak. Multi-Agent Systems: A survey, 2018. URL: https://www.researchgate.net/publication/324847369_Multi-Agent_Systems_A_survey.
[12] Matthew U. Scherer, Regulating artificial intelligence systems: risks, challenges, competencies, and strategies, Harvard Journal of Law & Technology, volume 29, № 2, 2016.
[13] O. Tkachenko, A. Tkachenko, K.Tkachenko, Ontological Modeling of Situational Management, Digital platform: information technology in the sociocultural area. Vol. 3. № 1, 2020. pp. 22-32.