Development of the algorithmic core of the control system framework

R. Osokin

***Abstract* — The concept of a software platform as a universal tool for developing control systems for robotic devices is proposed. The design of the system core is proposed based on the constructed framework model. A template of the framework algorithmic core is developed, which is aimed at decomposing the strategic tasks of the control agent into tactical control commands executed by robotic agents. The nature and properties of the proposed algorithmic core were considered from the points of view of the processes during its operation, the flow of global control cycles during device control, and in the perspective of the internal structure of functional components. Based on the work done, a descriptive model of the algorithmic core was detailed for creating a robotic device control framework.**

***Keywords* — algorithmic core concept, descriptive modeling, framework, robotic device.**

I. Introduction

The active development of technologies in the field of robotics tends to automate processes using various robotic devices. The range of industries in which robotic agents can be used is quite large: industry, oil and gas, energy, exploration, logistics and delivery, construction, etc. [1]. However, as a result of this, a tendency has emerged in which, when creating a new type of technical device, workers appeared to control their activities. A solution to this problem may be a control system for such technical means. At the same time, when creating such systems, a situation arises in which different developers, creating similar systems for different clients, solve the same problems, spending their own efforts and the clients' resources on their solution [2].

To manage robotic agents, companies can use software specifically developed for them, use control systems provided by the supplier of robotic devices, or use ready-made solutions [2], [3].

The problem with such ready-made solutions is that they are not always capable of either extensive expansion by increasing the number of robotic devices used or intensive expansion based on the use of a larger number of different types of devices. Also, the functionality of ready-made solutions may not always

meet the needs necessary for organizing the management of robotic agents. Such problems do not arise when a management system is specially created

for a specific company-user of robotic agents, such as: AWS IoT, Microsoft Azure IoT, Particle, Google Cloud IoT and others representing sets of services for industrial, consumer and commercial solutions capable of solving device management problems [4], [5]. Their main emphasis on cloud architecture allows making systems more flexible, easily integrating with other solutions or new services, scaling for the required tasks.

However, all of them actively predispose to the use of other parts of the entire web services package, which is expressed in the fact that even for a simple solution several services of different directions can be used at once. Because of this, the developer's task of creating a system is significantly complicated so much that in order to use these services it is necessary to obtain special knowledge on the operation of not only the required service, but also other parts of the service set [5].

II. Problem Description

The relevance of developing a universal software environment aimed at creating automated control systems for robotic agents is justified by the active growth in the number of areas of activity in which robotic agents are used, the automation of work processes of large companies by introducing robotic agents into their activities using automated control systems for robotic agents [6].

The idea of designing a universal tool for developing control systems for robotic agents is to create a software platform (framework), i.e. a special-purpose software environment, a framework used to facilitate the process of combining certain components when creating programs [7], [8]. The object of automation is the process of controlling robotic means using software specially developed for them, or using control systems provided by the supplier of robotic devices.

In the work [9], one of the authors of this article analyzed automated control systems for robotic agents and, based on the analysis, proposed a descriptive

Article received 24 January 2024.
R. Osokin, postgraduate student is with the Omsk State Technical University, Omsk, Russia (e-mail: origin1410199@mail.ru).

model of the average control process and a data flow model for device control. Conceptual requirements for control systems for robotic devices were formulated. The preliminary modeling results obtained by the author-developer allow us to form a framework model, the development of which will be described later in this article.

The main goal of this work is to create a more detailed model of one of the three layers of the framework described above.

III. Description of the core concept

The main part of the framework is its programmable algorithmic core template, which performs a series of sequential actions to execute the task of the control agent by timely generating appropriate control commands for the involved robotic agents.

Initially, the task set by the control agent is analyzed, as well as data on: the results of the agents' work, the environment surrounding the agents, the state of current processes, and the execution of tasks set by the control agent. This allows the context to be formed for subsequent decision-making on the allocation of resources for execution, if their description is not defined within the task by the control agent itself. Next, the task set and data are prepared for transfer to other processes for further use.

Then, in accordance with the information received, the process of allocating resources to begin solving the assigned task begins, focusing on the current situation of work processes.

Among such resources, the following can be highlighted:

1) robotic agents, divided into groups by: type, purpose, characteristics and features;

2) objects of interaction, which are the focus of the activity of robotic agents, which can be physical objects, space, people, etc.;

3) action space, which refers to the physical area of activity of a robotic device, such as the trajectory of a self-driving taxi or delivery drone [10];

4) the time allocated for performing some action, etc.

Once the tasks are defined and all the necessary data and resources are prepared, the process of solving the assigned control tasks begins. It is in this section of the algorithmic core that the abstract tasks of the control agents are transformed into sequential sets of control commands.

In accordance with the specified algorithms displaying the required work processes, the core begins to link the work procedures, their executors, objects over which the procedures proceed and the conditions of the procedures into single subprocesses of control of robotic agents, expressed, for the most part, by two types of actions: checking the conditions of the procedure and issuing the next control command, provided that the previous one was executed, and the secondary conditions correspond to the requirements of the work processes. The principle of decomposition is shown in Figure 1.

For such decomposition, the rules for transforming tasks into processes, processes into procedures, procedures into actions must be defined in advance. It should be noted here that such decomposition of business logic is limited by the scope of atomization of actions, directly determined by the set of functions of devices involved in solving the tasks. At the same time, in order to be available for different areas of application, the set of rules must be located outside the kernel itself, i.e. in the form of a separate file, which the kernel will access during its work.

It is also impossible not to note the fact that the depth of such a composition tree is directly proportional to the ease of integration of the framework with external decision-making systems.

In the case where the robotic agent control system developed using the proposed framework is already in the process of solving a task, and the assigned task is an addition to those already assigned, instead of this, the state of the resources allocated for the task is checked and, if necessary, new resources are used or the use of already used resources is stopped [8], [9]. This whole situation justifies the need for some “Control manager” component that would monitor, plan and administer the currently running work procedures. This solution allows us to present the model of the kernel processes as follows (Figure 2):

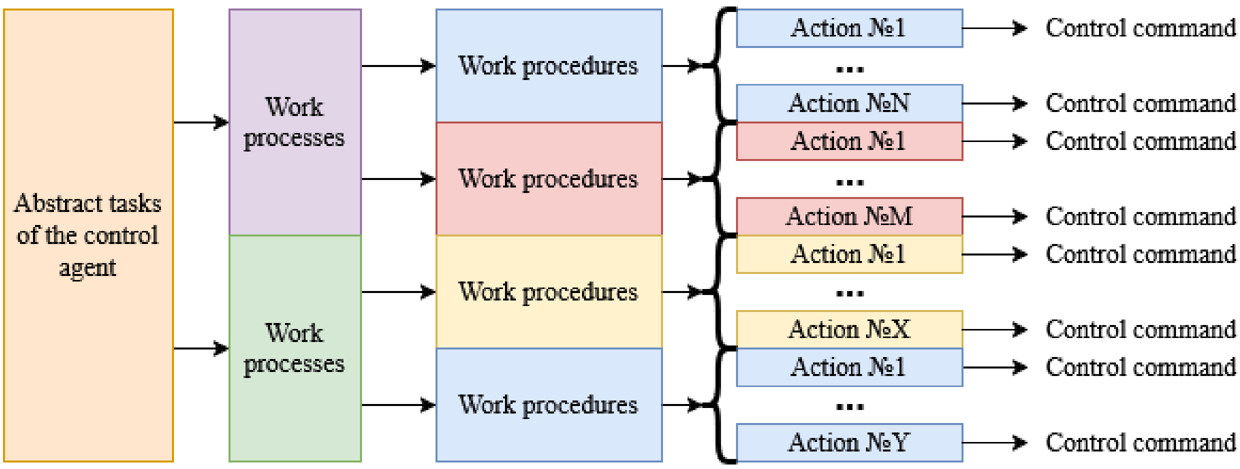

Figure 1: Decomposition of tasks into teams

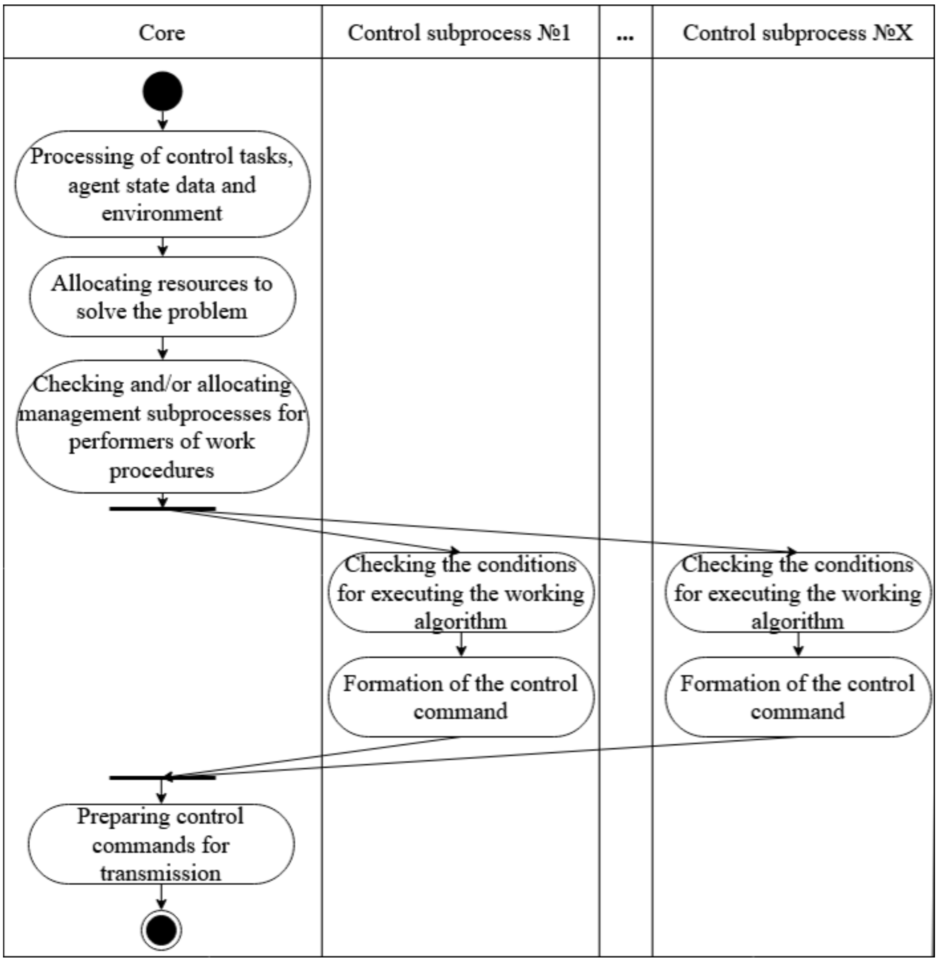

Figure 2: Core process model

It is also especially important to emphasize that in situations where procedures include direct interaction between controlled devices, a special role is played by the solution to the problem of synchronization of executable processes. The solution to this problem can be a system of waiting for interprocess communication or a composite representation of the process, where several devices are involved at once, etc.

IV. The position of the core in the framework structure

The core is a repository of all the business logic of the management system and within the framework occupies an intermediate position between the system management level and the level of communication with robotic devices (Figure 3), thereby allowing one to quite easily separate from aspects of interaction with both external, more abstract decision-making systems and from communication with more specific robotic devices.

Analyzing the device management process itself

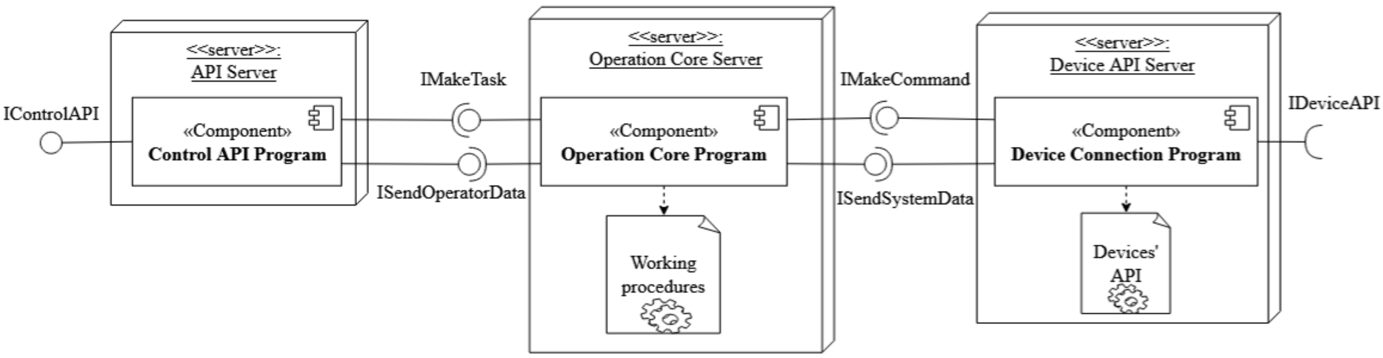

Figure 3: Framework structure model

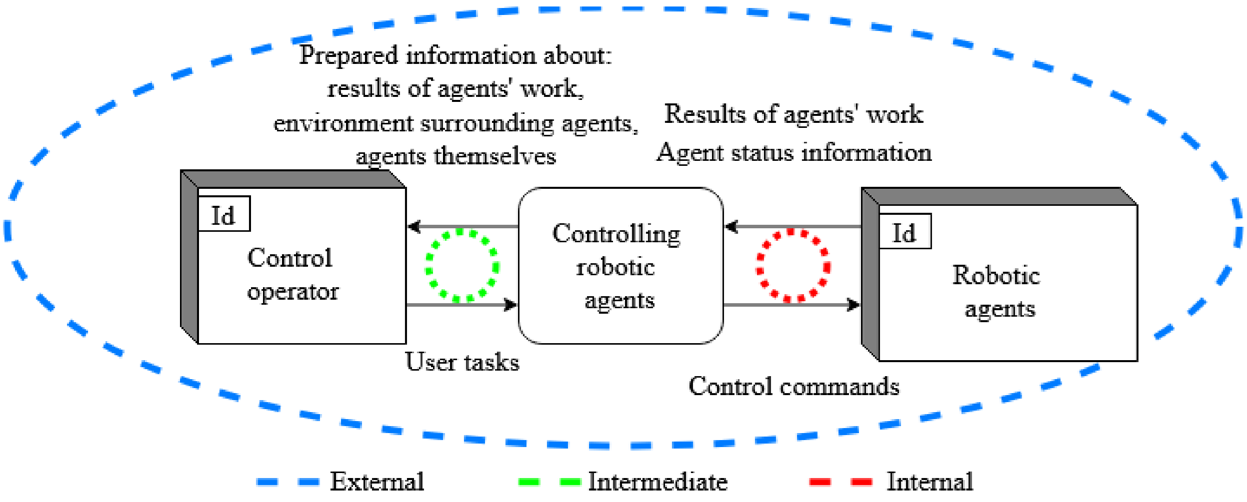

Figure 4: Control Cycle Diagram

(Figure 4), from the perspective of the framework being developed as a whole, the following management cycles can be identified.

External – a strategic management cycle that is global in nature and characterized by a relatively small volume of data passing through and/or at least a rare frequency of transmission. Information flows slowly in it and the changes it causes occur just as slowly. From a systems analysis perspective, this cycle can be characterized as decentralized.

Internal - a tactical management cycle, local in nature. The volumes of data passing through it can vary dynamically, while the frequency of their sending is very high, while both characteristics are dynamically determined directly in the process of managing devices and can also change depending on external factors. Management iterations are very short, especially compared to the external cycle, and therefore replace each other very quickly, with the changes caused by them occurring at a corresponding speed. This cycle is already characterized as centralized.

Intermediate – a cycle of transformative management, which is rather formal in nature, since it is aimed not at managing devices, but at supporting the process itself. Its essence is to ensure the exchange of necessary information between the two cycles described above. The cycle can include primary processing of returned data, accumulation of metadata of the management process, etc. This cycle determines the balance between centralization and decentralization of management in the system and the transition between them.

It is precisely the implementation of this cycle that the algorithmic core described in this work is aimed at.

Based on the above, we can determine the components that should be present within the algorithmic core, the resulting list of components allows us to create a primary representation of the algorithmic core from the point of view of the class model (Figure 5):

1) StateDataHandler – a component aimed at determining the current state of robotic agents, environment, etc. by systematizing the received data.

2) ResourceManager – the part that deals with

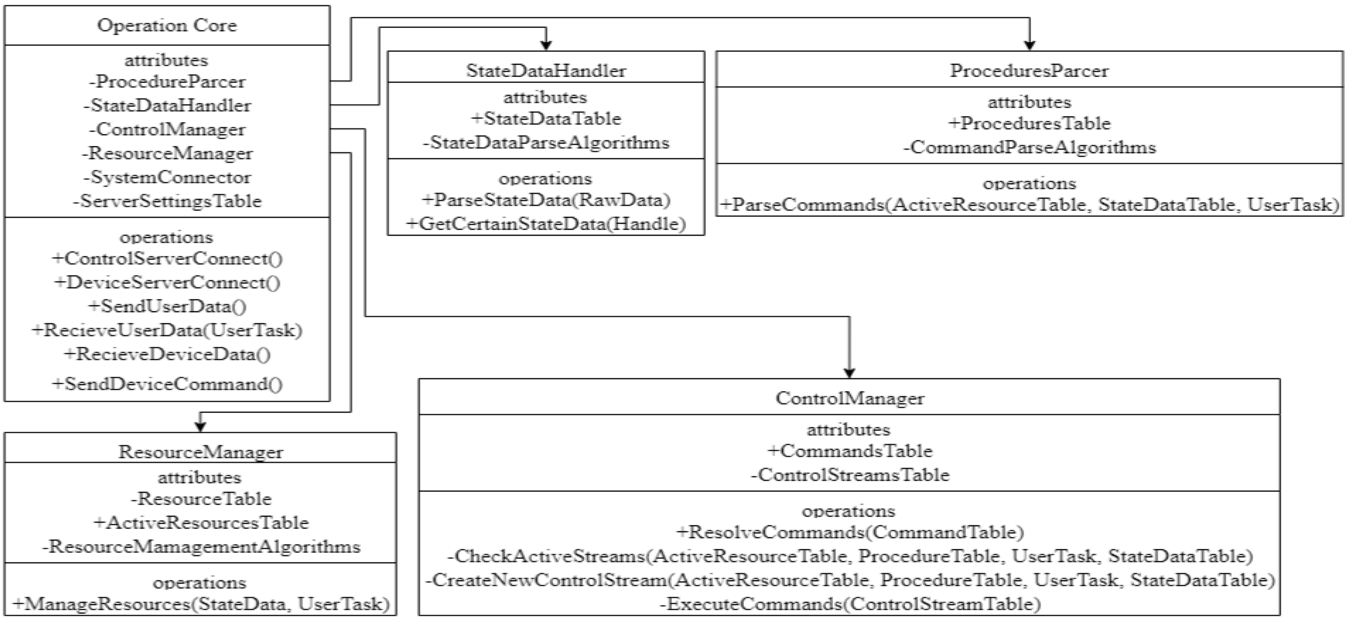

Figure 5: Algorithmic core class model

resource management based on the information received, operator tasks, information about available resources.

3) ProceduresParcer – a component that defines a list of procedures to be executed in accordance with operator tasks, allocated resources, data on the state of agents, the environment, etc.

4) ControlManager – a segment that deals with the management of device control subprocesses in which a control command is selected for each device based on the stage of the operating procedure, the state of the device, the state of the environment, etc.

5) SystemConnector – an interface designed to organize unified communication between the main levels of the framework: Control API, Operation Core, Device Connection.

V. Conclusion

Within the framework of the task set, an analysis of automation systems for managing robotic agents was performed, and the problems of creating such systems were defined. Based on the results of the analysis, a decision was made on the design of the system core, based on the previously developed framework model. A concept of an algorithmic core that performs the decomposition of tasks into commands was developed. For such decomposition, a concept of transforming tasks into processes, processes into procedures, procedures into actions was defined. A description of the essence of the core's work and a definition of its global goal were made from the perspective of the framework component. The conducted research allowed us to present models of the processes of the considered algorithmic core and its functional components.

References

[1] Wurll, C; Henrich, D; Woern, H; Schoel, J; Damm, M; Meier, W. Distributed Planning and Control System for Industrial Robots. International Workshop on Advanced Motion Control, Jun 29-Jul 1, 1998, pp 487-492.

[2] B. Eberhardinger, A. Habermaier, A. Hoffmann, A. Poeppel and W. Reif, "Toward Integrated Analysis & Testing of Component-Based, Adaptive Robot Systems," *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Vienna, 2016, pp. 301-302. doi: 10.1109/QRS-C.2016.45.

[3] Industrial Applications for Collaborative Robots [Online]. Available: https://www.rethinkrobotics.com/sawyer/industries

[4] AWS Certification [Online]. Available:: https://aws.amazon.com/ru/certification/

[5] Wing Demonstrates Decentralized Drone Traffic Management as Part of FAA Pilot Program – Wing. [Online]. Available:: https://wing.com/resource-hub/articles/wing-demonstrates-decentralized-drone-traffic-management/

[6] Supervising and Controlling Unmanned Systems: A Multi-Phase Study with Subject Matter Experts. [Online]. Available:: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4878290/

[7] JetBrains Project Rider. [Online]. Available:: https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:JetBrains_Project_Rider

[8] Freeman Adam. ASP.NET MVC 5 with examples in C # 5.0 for professionals. M.: Williams. 2015. 736 p.

[9] R. M. Osokin. Creation of a software platform for controlling robotic tools // Information technologies and control automation: Proc. of the XIV All-Russian scientific and practical. conf. students, postgraduates, education and industry workers (Omsk, May 26-27, 2023) / Omsk state tech. univ. - Omsk: OmskSTU Publishing House, 2023. - Pp. 166-170.

[10] Air Traffic Control System Command Center (ATCSCC) [Online]. Available: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/systemops/nas_ops/atcscc/ (дата обращения 25.05.2021).

[11] Olifer V. G., Olifer N. A. // Computer networks. Principles, technologies, protocols. - 4th ed. - St. Petersburg, 2010. - P. 438.

[12] Hunt, Craig. TCP/IP Network Administration. — 3rd Edition. — O'Reilly Media, Inc. — ISBN 0596002971.

Scientific advisor Nikonova Galina Vladimirovna, Omsk State Technical University.