# Smart Contract Protocols of Cognitive Agents Interaction in Distribution Control Networks

Arsenjev, D.G.; Baskakov, D.E.; Shkodyrev, V.P.

## Abstract

The concept of consensus dynamics in the control of distributed industrial plants and interconnected technological processesis developing based on contract network protocols of interaction in the environment of cognitive agents. The inefficiency of the classical control theory of complex organizational structures is demonstrated due to their uncertainty, multitasking and multidimensionality. As an advanced control model, an intelligent multi-agent control system is considered.

Keywords: Intelligent Multi-Agent Control System; Cognitive Agents; Consensus Dynamics; Network Cognitive Interaction Protocols.

## 1 INTRODUCTION

The modern infrastructure of industrial enterprises and production processes is characterized by extremely high complexity of interacting systems. Hierarchical information flows in such control systems are distributed, which often leads to abnormal situations. Traditional control paradigms are characterized by low efficiency in complex network structures with a large number of interacting distributed nodes and parallel information flows. This makes it necessary to create new principles for the control of similar objects based on using models of consensus dynamics with the use of intellectual cognitive agents. The proposed report is devoted to modern paradigms of intellectual control based on contracted network protocols of interaction of cognitive agents in distributed intellectual systems.

## 2 SMART CONTRACT PROTOCOLS OF COGNITIVE AGENTS INTERACTION IN DISTRIBUTION CONTROL NETWORKS.

The most promising approach to managing a modern industrial enterprise with complex production structure the use of an intelligent multi-agent paradigm. It should be noted that the concept of an intellectual control system is central to modern control system theory. Under the intellectual control system, we mean, first, a control system that is able to allocate, accumulate and order knowledge about the object or control objects, as well as self-learning in the broadest sense based on acquired knowledge, to change its objective functions and control algorithms.

Multi-agent control systems are a kind of agent approach to the control of distributed systems. An agent is a mathematical or software entity with its own control function [1]. The agent is part of a complex multi-agent system. The intellectual agent is primarily a software implementation with the functions of artificial intelligence.

An intelligent multi-agent control system consists of a set of intelligent agents in the structurally complex hierarchy of the system. Each agent has the following properties:

1. Autonomy - the ability to act in accordance with its objective function under the influence of external influence and the ability to control their actions.
2. Sociality - the ability to first exchange information, extract experience and knowledge, influence the control model, and find a consensus in the case of interaction with other agents.
3. Reactivity-the ability to respond to any external influences from other agents and control loops, as well as to environmental influences (fire, power outages, for example).
4. Proactivity - the ability to learn from previous knowledge, predict or anticipate possible changes, self-learning.

Complexity of control objects leads to the fact that the initial information about them often is incomplete, inaccurate. Quite often, such systems should be viewed as open actively interacting with the external environment. In the classical theory of multi-agent control systems, the agent is not endowed with the function of intelligence, that is, the ability to be proactive and self-learning, for example.

It should be noted, that the market has several solutions for creating universal multi-agent control systems. Let us briefly consider the main ones.

The JADE platform (*Java Agent Development Framework*)-the development framework supports the FIPA standard (*The Foundation for Intelligent, Physical Agents*). This environment involves cross-platform development of an entry level with a flexible graphical interface using the Java language at least 5.0. JADA agents exchange messages with themselves. Messages are generated in different queues, which allows you to change the configuration of the system. The key disadvantage of this product is its simplicity, which does not allow the development of industrial-level systems, the more intelligent multi-agent systems. The latest distribution kit of the system is dated June 8, 2017, and the FIPA standard of the latest specification is dated 2007, which is not a testament to the high relevance of the proposed solutions with this platform.

Agent Builder is an integrated environment for rapid and flexible development of intelligent control systems using agents, as well as agent-based applications. The developer claims full support for all versions of Java, as well as the KQML (*Knowledge Query and Manipulation Language*) language with the CORBA specification (*Common Request Broker Architecture*). Developer support has stopped on the version of Windows XP, which indicates that there is no prospect for this project in its current state.

Within the framework of this concept of intelligent multi-agent approach to control, it is proposed to consider the following paradigm, which differs significantly from the existing systems on the market.

Traditional control agents are essentially event-based information participants who are able to participate in control only based on receiving and transmitting one or another information, as well as actions in accordance with a given objective function. Intellectuality of such systems is essentially limited to a given system model, in which it is in principle incorrect to use this term in its entirety. At the same time, most authors completely overlook the process of learning and accumulating knowledge, as well as directly the algorithms of this training.

It is within the framework of creating promising models and intelligent multi-agent control systems using cognitive agents that the authors offer the following important conceptual additions that qualitatively change the existing paradigms of intellectual control systems, namely:

1. Use of theory and mathematical apparatus of consensus dynamics.
2. The use of graph theory, as well as the concept of a direct acyclic graph (DAG, Directed Acyclic Graph).
3. Use as a knowledge base of graph databases (for example, GraphDB Lite, Neo4j Community Edition), as well as machine learning systems such as H2O, TensorFlow and Scikit-Learn [2].
4. Consensus algorithms of interaction between intellectual agents.

As previously reported, the agent in our understanding is a certain mathematical entity, which usually has a software implementation. The intellectual agent acts rationally and optimally based on the acquired knowledge in the process of its development and training. In a complex distributed control system, we have hierarchies of intelligent agents, each of which participates in the control of individual subsystems. The most important feature of the intelligent agent in the multi-agent control system is the presence in addition to the basic target functions and models of interaction with other agents using the functions of artificial intelligence, such as:

1. Learning using machine learning algorithms with the teacher (controlled learning): Decision tree, naive Bayesian classification, support vector method, ensemble method.
2. Learning using machine learning algorithms without a teacher (uncontrolled learning): clustering algorithms, main components method, independent components analysis.
3. Neural networks.
4. Consensus algorithms based on artificial intelligence.

It is important to understand that the intelligent control agent, being a part of a complex multi-agent system, can have different states of its targets and control functions, as well as the whole system as a whole. In addition, the control system itself should be able to change on the fly as control hierarchies depending on external and internal influences, as well as contingencies, and the control algorithms themselves. All this requires the use in real time of ready-made control models, as well as other control models obtained in the training process, especially in the learning process without a teacher.

To store data control models, as well as their possible modernization and it is intended to use graph databases. The latter proved to be an extremely effective tool for working with incomplete data, complex structured data, as well as data that have a hierarchical structure.

Under the contract network protocol of interaction, we mean primarily a set of algorithms and rules for the interaction of intelligent agents among themselves in the hierarchy of a distributed multi-agent control system. Since these interaction algorithms also include elements of artificial intelligence, we consider it possible in this case to speak of cognitive intellectual agents of control.

The software implementation of such control systems is a non-trivial task, which is the reason for the complete virtual absence of certain open solutions on the market, which at least partially satisfied the task in this formulation.

Cognitive multi-agent control systems have the following important characteristics that significantly affect the software implementation of these complexes, namely:

1. High competitiveness of the interaction of cognitive agents among themselves, which is the reason for using the methods of consensus dynamics.
2. Extremely high degree of parallelism of ongoing control and information processes.
3. A large number of loops with feedbacks.
4. Information processing in real time.

It seems extremely promising to use the paradigm of reactive programming to implement similar tasks. Since such systems are additionally saturated with a large number of internal and external events and calls, the control system data relate to high-data data applications (DIA, Data-Intensive Applications) [3]. We understand by application, high-loaded data, an application in which the quality of data, the degree of their complexity, and the speed of their changes are key, in contrast to the application-intensive applications (CIA, Compute-Intensive Applications).

Thus, the concept of reactive programming is becoming key in the development of such multi-agent control systems [4].

It is important to note that reactive programming is part of the functional programming paradigm, which, according to the authors, should be the basis for the development of such control systems.

The most interesting and useful in the process of writing code is the use of the library Rx Java (Reactive X) in conjunction with Java Stream [5]. This Java library has the following key benefits that are inherent in functional programming:

1. Pushing and pulling events.
2. Synchronous and asynchronous modes of operation of the control system.
3. Competitiveness and concurrency.
4. Lazy and energetic type.
5. Duality.
6. Use of reactive extensions (subscription to notifications, wiretapping notifications, hot and cold objects and events).
7. Multithreading.
8. Periodic sampling and discarding of events.
9. Back pressure and resistance to events.

One of the most promising areas for the use of cognitive agents in distributed intelligent control systems is the area of supply chain management (SHM, Supply Chain Management) and logistics in particular with the additional use of the IoT concept

(Internet of Things) or II (Industrial Internet). The use of the latter will require with a high degree of probability the introduction of distributed registry technologies or blockchain technologies, which allows us to speak about the creation based on cognitive agents of control systems of the latest generation with advanced functions of artificial intelligence.

Using intelligent multi-agent control system based on the concept of contracted network protocols shows its high efficiency in terms of reducing, for example, the cost of human capital. In the course of test trials, supply chains managed to achieve a reduction in such fixed costs by more than 2 times. At the same time, the overall efficiency of the control system proved to be much higher than the use of traditional logistic models based on linear programming.

## 3 CONCLUSION

The report examines the concept of contracted network protocols for the interaction of cognitive agents in distributed intelligent control systems. Prospective directions of realization of these models with use of modern development tools are offered. These methods and paradigm the team of authors use in practical work and creation of software and hardware prototypes for industry customers from the sphere of oil refining and oil production.

## 4 REFERENCES

[1] Д.Г. Арсеньев, Иванов В.М., Шкодырев В.П. Адаптивное управление вычислительным процессами. СПб.: Издательско-полиграфическая ассоциация университетов России, 2016, 315 с.
[2] Robinson I, Webber J, Eifrem E. Graph Databases. 2015. O'Reilly Media.
[3] Kleppmann M. Designing data-Intensive Applications. 2017. O'Reilly Media.
[4] Burns B. Designing Distributed Systems. 2018. O'Reilly Media.
[5] Nurkiewicz T, Christensen B. Reactive Programming with RxJava. 2017. O'Reilly Media.

---

Author: Arsenjev, Dmitry

University: Peter the Great St. Petersburg Polytechnic University

Department: Higher School of Cyber Physics and Control

Author: Baskakov, Dmitry

University: Peter the Great St. Petersburg Polytechnic University

Department: Higher School of Cyber Physics and Control

E-mail: baskakovdmitrii@gmail.com

Author: Shkodyrev, Vyacheslav

University: Peter the Great St. Petersburg Polytechnic University

Department: Higher School of Cyber Physics and Control

---