ISSN 1392 – 1215 ELEKTRONIKA IR ELEKTROTECHNIKA. 2004. Nr. 3(52)

T 125 AUTOMATIZAVIMAS, ROBOTOTECHNIKA

Investigation of Multi-Agent Control System

V. Mačerauskas, V. Teresius
Dept. of Control Technology, Kaunas University of Technology
Studentų str. 48, LT-51367 Kaunas, Lithuania; phone: +370 37 300290. e-mail: v.macerauskas@ktu.lt
e-mail:vittere@merkys.ktu.lt

Introduction

The new trends in the market and business, new requirements of customers, short life cicle of products, e-business technologies require re-usable, distributed automation systems, that improve competitiveness of the business [1]. The new requirements for the new control systems are the distribution and decentralisation of system units, knowledge and skills, definite communication mechanisms for the integration of distributed system units, the fast adaptation in the control and the reaction to disturbances (failures and changes of organization) and interaction between physical devices [2].

The methods were developed to suit the new requirements and are agent-based technologies, that are autonomous, organized and reactive. Using the results of a multi-agents systems application and using traditional control methods, it is possible to compare both methods and highlight the benefits of the use of multi-agents technology in manufacturing and automation systems. Reduction of batch size, increased production mix and the fast response to changing customer demands have increased the flow of information and the variety of data within the factory. Monolithic production systems cannot effectively suit these new requirements.

These changing requirements lead to the next generation - Intelligent Manufacturing Systems (IMS), which consist of an integrated network, of distributed resources, of control knowledge's and material processing.

Intelligent control

This control system is particularly suited for Intelligent Manufacturing Systems and is being developed using intelligent agent technology [3]. The physical implementation of the intelligent control system has the major components [4]: microcomputers (network hosts), primary (factory control) networks, distributed intelligent controllers and secondary (field bus) networks.

The microcomputers are using for the purposes of the intelligent programming interface, system configuration, status monitoring, data acquisition and supervisory planning and control. These costs are maintaining running information flow and present logical view of the entire system. The primary communication network is necessary for control and coordination all the distributed intelligent controllers in the factory control system.

The distributed intelligent controller dynamically controls manufacturing resources in real-time and consists of multiple intelligent controllers. The physical architecture of a distributed intelligent controller is a free coupled system of intelligent controllers [1]. The secondary communication network is necessary for the interface between one distributed intelligent controller and its distributed intelligent sensors and actuators. Such architecture provides a effective control and fault-tolerance [5,6].

Each intelligent controller consists of a number of logically (and even physically) separate autonomous and cooperating operational modules. These operational modules include the processor module, communication module, I/O module, system software module and application software module. The general functional characteristics of the operational modules of an intelligent controller can be categorized as follows: autonomy, fault tolerance, interoperability, real-time functionality, reconfigurability, intelligent system interface.

Both the physical and logical structure of a distributed intelligent controller is changeable according to application requirements.

Intelligent design

An internet-based distributed database and knowledge base modeling system has been developed for modeling concurrent design activities conducted at different locations. In this system, the databases and knowledge bases for different product life-cycle aspects, including design, manufacturing, etc., are modeled at different internet locations.

The database at an internet site is modeled by particular features [1]. A feature is a collection of product descriptions for a particular purpose. For example, a gear is a design feature for modeling a design object and a hole is a manufacturing feature for planning production process. Each feature is defined by its element features, attributes, relations among element features, and quantitative relations among attributes. Features are modeled at two different levels, class level and instance level, corresponding to standard product libraries and special product data, respectively. Instance features are generated using class features as their templates. The relations among product data are maintained using a data relation network. Any change of the product description can be propagated to other parts of the product descriptions through the relation network.

The knowledge base at an internet site is modeled by IF-THEN rules. These rules are organized in many groups, called rule-bases.

Intelligent Scheduling

Production scheduling is a process to allocate appropriate resources for the required manufacturing tasks and to identify the sequence and timing parameters to accomplish these tasks [1]. Limited resources include facilities, personnel, materials and so on. A good schedule can reduce the efforts in manufacturing, thus improving the competitiveness of products.

An intelligent scheduling system has been developed based upon artificial intelligence techniques. This system provides two scheduling mechanisms: a predictive scheduling mechanism and a reactive scheduling mechanism.

Predictive scheduling is an approach to identify the optimal schedule based upon the provided constraints, prior to the actual production process. Reactive scheduling, on the other hand, is an approach to modify the generated schedule during the production process, when the original schedule cannot be executed due to the changes of manufacturing conditions, such as breakdowns of machines and sudden absence of persons.

Manufacturing requirements are defined as tasks. Each task specifies its required types of resources, including machines and persons, and required time period. The sequences for these tasks are defined by a graph that consists of these tasks. A product is modelled by product primitives called features. Each feature is associated with a graph of tasks.

Traditional system approach

The traditional manufacturing system uses a hierarchical architecture approach (fig. 1). The cell controller architecture consists of several modules, whose "brain" is the manager module, which is responsible for the control and the supervision of the production process of the manufacturing system and also for the management of system resources. Each physical device has an module, designated by device controller.

The interface between the manufacturing controller and each of the industrial machines is implemented using the communication and define a standardized message system for exchanging real-time data and supervisory control information between networked devices and computer applications.

The traditional approach presents the following problems:

1. **Reconfiguration -** applications have a rigid organizational structure. It falls down when it is necessary to change the organizational structure (for example, new shop floor layout, new strategies for the hierarchy, etc).
2. **Learning and disturbance management -** it is hard and complex to introduce intelligence in the application, in order to optimize its execution and to manage the disturbances and warnings.
3. **Distribution and decentralization -** doesn't support efficiently the distribution and decentralization of functions and entities.
4. **Code re-usability -** the development of this type of applications based in this traditional approach has the advantage of its simplicity, when compared with other advanced approaches, but the code developed cannot be re-used.

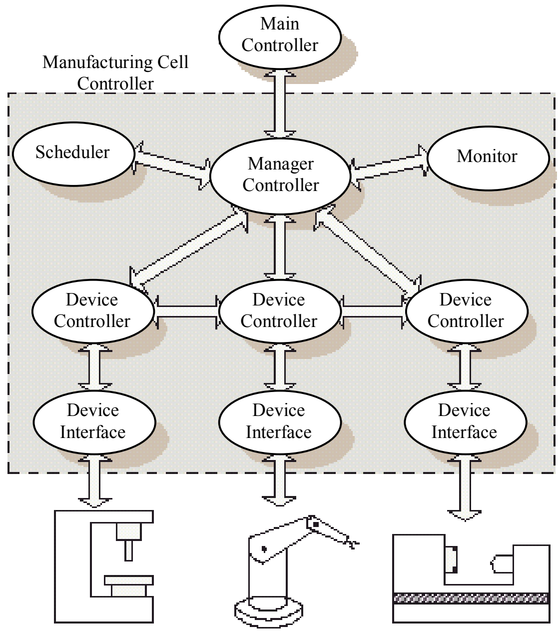

Fig. 1. Manufacturing system using a traditional approach.

The multi-agents based approach

The main and unsolved problems of traditional control approaches are the distribution of functions, cooperation between distributed entities, reaction to disturbances, adaptation to new environments and code re-usability.

The multi-agent systems are defined as sets of agents, which represent the objects of the systems and through co-operation mechanisms perform complex tasks. In the automation and manufacturing range, an agent is a software object that represents automation and manufacturing system objects, such as tasks, CNC machines, robots, PLC devices and sensors.

The multi-agent technology is suitable to the distributed manufacturing environment, since the automation and manufacturing applications characteristics like modular, decentralized and changeable. The benefits of multi-agent technology are [7]:

1. **Autonomy -** an agent can operate without the direct intervention of external influence and has some kind of control over their behavior.
2. **Cooperation -** the agents interact with other agents, in order to achieve a common goal.
3. **Reactivity -** the agents perceive their environment and response quickly to changes that occur on it.
4. **Proactivity -** the agents do not simply act in response to their environment.

5. **Adaptation and decentralization -** the agents can be organized in a decentralized structure, and easily can reorganized into different organizational structures.

The improvements of the use of multi-agents technology in automation and manufacturing systems are the fast adaptation to system reconfiguration (for example addition or removal of resources, different organizational structures, etc.), re-use of code for other control applications, increase of flexibility and adaptation of the control application and more optimized and modular software development.

The architecture defines agent classes (fig. 2): operational agents, supervisor agents, product agents, task agents, system management agents.

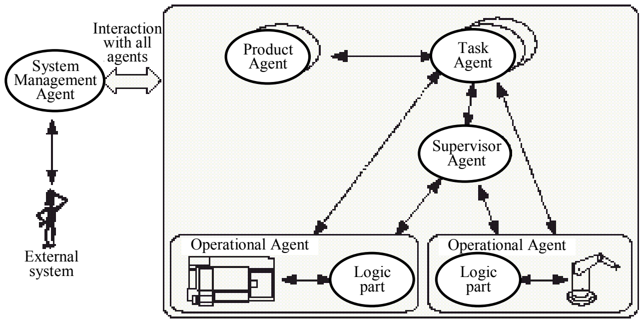

Fig. 2. Agent classes

The operational agent controls the interactions with the physical resources, such as CNC machines, robots, humans, PLC devices and sensors. The operational agent has two components: a logic part that controls and interacts with the physical device.

The supervisor agent coordinates and supervises others agents, in accordance with the organizational structure and can represent cell controller.

The product agent represents the available resources in the system and takes care of the product knowledge and the associated information, such as the process plan.

The task agent controls the execution of a task in order to produce a product and it contains the dynamic information about the manufacturing orders.

The system management agent administrate the system, supervise and registrate the agents belonging to the system.

The architecture for a generic agent is based in three modules and a local database, which contains all relevant information about the behavior of the agent.

The decision-making module controls all activities of the agent and includes a decision engine, cooperation mechanisms, dynamic organizational techniques and learning mechanisms, in order to take decisions and solve problems. The information stored in the local database involves several types of knowledge, such as objectives, constraints, experience, decision rules, procedures, dynamic information and organizational related information.

This structure supports the agile adaptation to different organizational structures and to self-organization, which allows the re-organization of the agents into different organizational structures. To support this feature, are used methods to represent the relationships between the agents for each organization structures and techniques to handle this re-organization. For example, if one agent leaves the system, it is necessary that all other associated agents update automatically and dynamically their relationships.

There are disturbances (alarms, layout changes, etc.) in the manufacturing environments that deviates the process from the original plans. In case of disturbances the control system should respond dynamically and quickly, using mechanisms that comprise a disturbance engine, which finds out the best plan to control the disturbance in accordance with pre-defined rules and knowledge's.

Control of the system

In order to interact with other agents in a agent system, it is necessary that all agents make the registration into the system management agent, that is responsible for the list of available agents in the system.

The existence of an agent that manages the registration in the system can lead to a possible centralization in the system, because if the system management agent fails, all the agents lose the reference to the other agents. In order to avoid this possible centralization, it is used a decentralized procedure to have the references for the system agents.

Whenever an agent enters in the system, it sends a message to the system management agent to register in the system and receives a list of actual available agents in the system. This agent sends a notification message to all active agents in the system indicating its availability. The agents that receive this notification add the new agent to its agents list. When an agent leaves the system, it sends deregistration messages to all active agents and to the system management agent. The agents that receive this message should remove the agent from its agents list.

Multi-agent modeling

Multi-agent systems are defined as sets of agents which interacting together perform complex tasks. In the automation and manufacturing area an agent is a software object with an indepedent programming code. The mentioned agent controls such objects as CNC machines, robots, PLC devices and sensors [9].

The purpose of this work is to model the multi-agents with the help of Concept software and visually to show the interaction of agents. By this experiment it is shown such features of multi-agents as reciprocity, easy reconfiguratio of the system, which consists of some agents, decentralized control [5,6].

In the experiment there are shown five agents, wich seeking the object move in different directions in the area of measurements' set. Agents move in different speed. The algorithm of agents' control is simple (fig. 3)

The agent start moving from the start-up place till it reach the area limits or bumps into another agent or it findes the object. When the agent bumps into another agent or reach the area limit it starts moving in backwards. Realistically these agents could be as moving robots, wich are shown as circles. It is used the simulation of agents in the experiment, so it is created one robot‘s moving control programme, which is the same as in other robots‘ one.

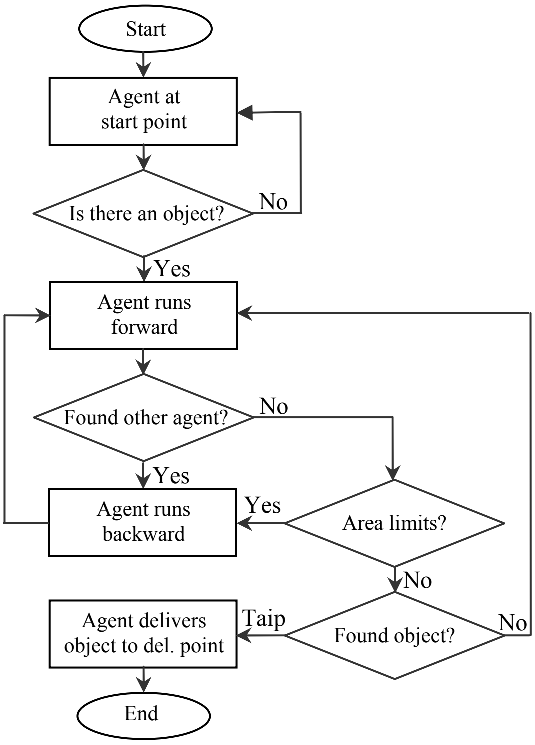

Fig. 3.- Agent control algorithm.

Agent control programme is based on Concept software function block, in which existing control‘s alghorithm process values of inputs and gives some values of outputs. Multi-agents is characterized that their control programmes are the same, but they process different values and give different outputs (fig. 4).

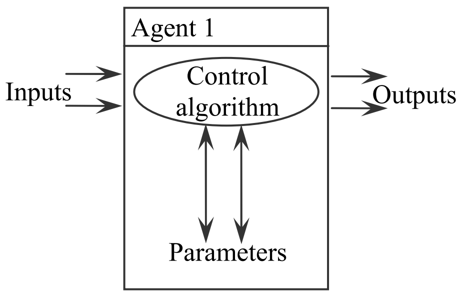

Fig. 4. Agent structure

In the case of traditional hierarchic control all the agents are controled by common control device, which decides where the agents should move [5,6].

Some changes appear in the programme when a new agent is beeing sent to such system. The mentioned system is not realible. For instance if the programme of control device or controller for some reasons has failed so all the system becomes unusable. In the case of multi-agents one or some agents can be broken, but the remaining capable agents are performing the common task.

So the process is not stopped even some troubles have appeared. The chance that agents‘ system‘s process will be broken is in proportion to the number of agents and is less than a traditional control system with one control device.

By experiment the control programmes of all agents run in one controller (in one imitator). In this case the process is like a centralist control. Despite this the funcional blocks are programme separate. In practise control programme‘s functional blocks would be robot‘s contoller‘s programme, which would cooperate with other robot‘s programme by external sensors. According to such principle is programmed agents‘ controllers.
Every agent has digital and analog inputs and outputs, for which are assigned addresses of simulated Concept controller. By such addresses simulated controller exchanges data with visualization software In Touch.

In the programme agents exchange data by variables. (fig. 5):

1. Position coordinates – it can be explained as meanings, which can be calculated by agent‘s programme or those meanings can be taken from other agents (for example, a1_x and a1_y are agents No. 1 variables‘ meanings, x and y are coordinates, which are sent to other agents.).

2. Information – the signals of object detection „found“, „found 1“ and so on.

3. Agents seeking for the object move in the territory of set area (inputs „xmin“, „xmax“, „ymin“ and „ymax“). Moving means the change of agent‘s coordinates in special period of time.

Agent calculates its position‘s coordinates according to the coordinates of other agents and then sends them to other agents.

For example, the agent No. 1 (Agent_1) according to inputs x_a, y_a (the coordinates of agent No. 2), …, x_d, y_d (the coordinates of agents No. 5) and x_obj_1, y_obj_1 (the coordinates of seeking object) calculates its own coordinates x, y. Inputs x_skaic, y_skaic are coordinates‘ variation speed, which is determined in the visualization window (0...100).

Agents compare each others position‘s coordinates. If the distance between coordinates is less or equal to defined ranged, then the meanings of coordinates is being calculated backwards (for example, if it was calculated in cumulative order, than after equalization it will be calculated in descending order). Also agents begin moving backwards when they reach the limits of area.

Moving in the area agents compare thier coordinates with the coordinates of the object. Object is an autonomic agent too. If the distance between coordinates is less or equal to defined one, its mean that the agent has found the object. The agent which has found the object sends a signal “radau” to the object (object‘s inputs “rado1”, ..., “rado5” according to agent‘s number). From this moment the coordinates of agent and object becomes equal and then they both move to the object‘s delivery area.

A lot of agents with thier own programme can be incorporated in such a multi-agent system. For such a purpose control programmes are being changed a bit. For example if we want to reduce the number of agents, we have just delete the block of agent and in the rest blocks are deleted not used agent‘s variables, by which agents have exchanged data (fig. 6).

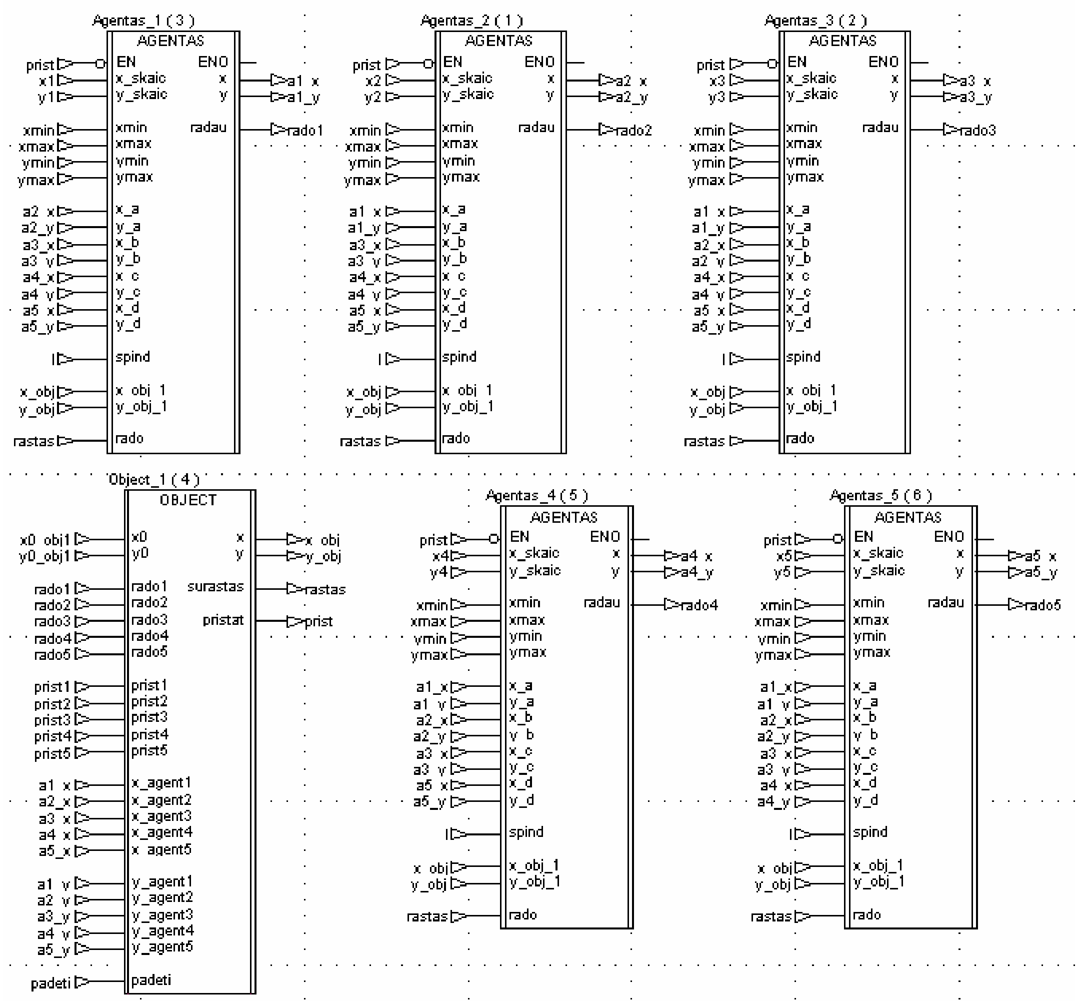

Fig. 5. Multi-agent function blocks.

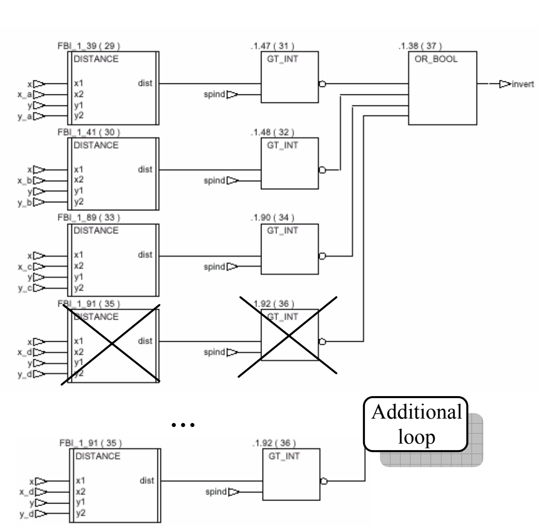

Fig. 6. Deleted loop in agent control programme

By the experiment it was used Schneider automation GmbH software of Concept for controllers‘ programming, version 2.1 XL, which fit IEC 1131 standarts. By that software is written the control programme, which was load into the simulated controller.

For the system portrait was used Wonderware® FactorySuite™ InTouch software of visualization.

Simulated controller‘s (Concept imitator) and visualization software data‘s exchange is actualized by Wonderware® Modicon Ethernet I/O Server software.

Conclusion

After the implementation of the agent-based cell controller prototype and with the experience of previous implementation using a traditional approach for the same application, it is possible to compare both approaches and find out the benefits of each approach.

The use several language and distributed communications platforms, to develop control applications, allows the use of the same application in different operating systems environments. The multi-agent technology allows an easy and modular development of control applications.

Some components of the developed control application can be re-used for other applications. Each agent is autonomous, has control about its behavior and has local and community knowledge. It is possible to build separate and independent agents that can be placed in a distributed environment.

The addition of intelligence to a agent, for example to take decisions, manage disturbances or learning, can be viewed as a plug-in of an intelligence module, which takes easier the development of control applications. For small or specific uses, the traditional approach can present advantages of small complexity in the code development. In this case it isn't important the re-use of the code and the multiplexing of the developed code.

Multi-agents systems are more reliable than traditional control. The main difference between multi-agents and traditional control system is that multi-agent have not common control device, which could dictate the agents' actions. Agents make decisions by thier own.

An agent-based architecture approach to the development of distributed manufacturing applications and its implementation into the agent-based manufacturing cell controller was investigated.

In comparison with the traditional approaches, the agent-based approach presents important improvements such as expansibility, robustness, reactivity, support to distributed environments and re-use of the application code.

References

1. **Klostermeyer A.** Revolutionizing Plant Automation: The PABADIS Approach. – Germany: University of Magdeburg, 2002. – 22 p.
2. **Goldberg D., Matarić M.J.** Design and Evaluation of Robust Behaviour-Based Controllers for Distributed Multi-Robot Collection Tasks. – USA: Los Angeles, 2001. – 24 p.
3. **Zhou B., Whang L., Norrie D.H.** Design of Distributed Real-time Control Agents for intelligent Manufacturing Systems // Proceedings of the second International Workshop on Intelligent manufacturing Systems. - Leuven, Belgium, September, 1999. - P. 237-243.
4. **Leitao P., Restivo F., Putnik G.** A Multi-Agent Based Cell Controller // 8th IEEE International Conference on Emerging Technologies and Factory Automation. - Nice, France, October 15-18, 2001. - P. 1-8.
5. **Diedrich C., Russo F., Winkel L., Blevins T.** Function Block Applications in Control Systems Based on IEC 61804. – Germany: Ifak, 2001. – 10 p.
6. **Schrott G. A**. Multi-agent distributed real-time system for a microprocessor field-bus network. - Germany: Technische Universität München, 2002. – 15 p.
7. **Norrie D. H.** Schema-based conversation modeling for intelligent agent-oriented manufacturing systems. – Canada: The University of Calgary, 2000. – 6 p.
8. **Mačerauskas V., Teresius V.** Multi-agentų valdymo sistemų apžvalga // Automatika ir valdymo technologijos – 2003. – Kaunas: KTU, 2003. - P. 132-136.
9. **Teresius V.** Multi-agentų modeliavimas Concept paketu // Automatika ir valdymo technologijos – 2003. – Kaunas: KTU, 2003. – P. 137-140.

Pateikta spaudai 2004 03 01

V. Mačerauskas, V. Teresius. Multi-agentinės valdymo sistemos tyrimas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2004. – Nr. 3(52). – P. 21-26.

Naujos tendencijos gamyboje, produktų trumpas gyvavimo ciklas, elektroninio verslo technologijos reikalauja greitų, daugkartinio panaudojimo valdymo sistemų. Keliami reikalavimai naujoms valdymo sistemoms yra greitas valdymo prisitaikymas reaguojant į trikdžius (gedimai, organizaciniai pasikeitimai) ir sąveika tarp fizinių įtaisų. Lyginant su tradiciniais metodais, agentais paremtas metodas atspindi svarbius pagerinimus, tokius kaip išplėtimas, aukšta kokybė, reakcingumas, pagalba paskirstytoms aplinkoms ir pakartotinas pritaikomųjų kodų panaudojimas. Multi-agentinė architektūra leidžia greitą prisitaikymą prie naujų aplinkų ir savaiminį persiorganizavimą, siekiant individualių ir bendrų tikslų, yra atspari klaidoms ir greitai reaguojanti į sistemos pažeidimus. Tyrimais įrodyta, kad palyginus su tradicinėmis paskirstytomis automatizavimo ir gamybos valdymo sistemomis, multi-agentų technologija sudaro geras galimybes judrių valdymo sistemų įgyvendinimui. Il. 6, bibl. 9 (anglų kalba; santraukos lietuvių, anglų ir rusų k.).

V. Mačerauskas, V. Teresius. Investigation of Multi-Agent Control System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2004. – No. 3(52). – P. 21-26.

The new trends in the market and business, new requirements of customers, short life cycle of products, e-business technologies require re-usable, distributed automation systems, that improve competitiveness of the business. The new requirements for the new control systems are the distribution and decentralisation of system units, knowledge and skills, definite communication mechanisms for the integration of distributed system units, the fast adaptation in the control and the reaction to disturbances (failures and changes of organization) and interaction between physical devices. After comparing both traditional approach and agent-based approach, it is possible to find out the benefits of each approach: platform independency, code re-usability, distribution and autonomy, plugging intelligence. Ill. 6, bibl. 9 (in English; summaries in Lithuanian, English, Russian).

В. Мачераускас, В. Тересюс. Исследование мулти-агентной системы управления // Электроника и электротехника. – Каунас: Технология, 2004. – № 3(52). – С. 21-26.

Новые направления на рынке и промишлености, новые требования от клиентов, короткий цикл жизни продуктов требуют повторно используемых, распределённых систем автоматизации, которые улучают конкурентоспособность дела. Новые требования – это распределение и децентрализация системных устройств, знания и способности, механизми интеграции распределённых систем, быстрая адаптация в управлении и реакции на сбои (неполадки или изменения структуры), взаимодействие между устройствами. После сравнения традиционных и агентных систем, можно почеркнуть преимущества перед традиционными методоми: многократное пользование кодов, распределеность и децентрализация, интелект. Ил. 6, библ. 9 (на английском языке; резюме на литовском, английском и русском яз.).