Event-based Cooperation of Functional Networking Components in Distributed Technological Systems

Valery A. Kokovin
Department of Automation of Technological Processes
State University "Dubna", branch "Protvino"
Protvino, Moscow reg., Russia
kokovin@uni-protvino.ru

Alexander A. Evsikov
Department of Automation of Technological Processes
State University "Dubna", branch "Protvino"
Protvino, Moscow reg., Russia
eaa@uni-protvino.ru

Saygid U. Uvaysov
Department of Design and Production of Radio-Electronic Means
MIREA – Russian Technological University
Moscow, Russia
uvaysov@yandex.ru

Svetlana S. Uvaysova
School of Computer Engineering
Higher School of Economics
Moscow, Russia
uvay@yandex.ru

***Abstract*—The paper defines a new class of network devices Functional Networking Components (FNC). FNC can receive and process information (the presence of a computer with network ports), the FNC has a physical nature (mechanics), and it is able to affect the physical environment. The FNC mechanisms that allow processing input information from sensors and generate events and messages are examined, as well as the requirements for them are determined. The paper analyzes the advantages of using FNC computers based on a traditional microcontroller (ARM as an example) with the addition of a hardware accelerator on FPGA for processing fast processes. A simplified block diagram of the event's generator, messages and synchronization is given.**

Keywords—distributed technological systems, functional networking components, FPGA

I. INTRODUCTION

The development of real-time control systems for distributed technological systems is associated with certain difficulties. These difficulties are caused, firstly, by the territorial remoteness of individual subsystems, which complicates their real-time interaction. Secondly, the process control algorithms of individual subsystems can be interdependent.

An example of such distributed systems is the subsystems of a continuous rolling mill. The main feature of the continuous rolling process is the interconnection of individual stands (systems of rolling rolls) of a continuous mill through a rolled pipe in compliance with the equality of the metal volume passing through each stand per unit time.

Another example is the technological subsystems of a cyclic charged particle accelerator. Each accelerator is equipped with certain timing tools that are necessary to provide real-time control systems and synchronization of technological processes on the charged particle accelerator. To obtain a stable beam of charged particles with given parameters [1] the technological processes of the individual subsystems are interconnected.

To ensure cooperation in solving a distributed technological problem, telecommunication systems are usually used. The term "cooperation of subsystems", in contrast to the terms "synchronization" and "coordination", will be understood as the interaction of technological subsystems, the algorithms of which are interconnected. The work [2] analyzes the algorithmic dependence of such subsystems according to data and control. Of course, the term "cooperation" of devices of distributed technological systems does not exclude synchronization, but only indicates a certain additional feature of these devices.

Successfully introduced into production, the concept of Internet of Things (IoT) [3], called Industrial IoT, added intellectual capabilities to manufacturing equipment, processes and control. The participation of leading Hardware / Software companies, such as Intel, Bosch Si, ABB and others, ensures the successful integration of IIoT components in industrial automation [4]. The smart manufacturing solutions of these companies use connected sensors and devices to increase machine and human performance in real time and transfer data to the cloud for deeper analysis. The development of IoT and IIoT gave impetus to the emergence of a new direction, which co-operates the work of robots or robotic devices with the help of network technologies. This Internet of Robotic Things (IoRT), aims to implement robotic technology by expanding the functionality of IoT and IIoT devices. In [5], the IoRT concept was presented, which emphasizes great flexibility in the development and implementation of new applications for network robotics, while achieving the goal of providing distributed computing resources as the main utility. In these network associations (IIoT and IoRT), the term "internet" can be interpreted as a combination of devices through computer networks that use the protocols of global and local networks for interaction. In addition, participants in network associations use a huge variety of interfaces and protocols to receive information from sensors and to transmit control signals to actuators. The main difference between IoRT devices and IoT is that IoRT can affect the physical environment, and sometimes even change it.

As the network associations of functional devices described above, we can talk about the Internet of Mechatronic Components, a subset of which is IoRT. On the other hand, in distributed technological systems, participants may include not only mechatronic components, but also, for example, self-sufficiency electrical devices with built-in intelligence. That is, devices that do not contain precise mechanics (which is a sign of mechatronic devices). By self-sufficiency we mean the devices ability to solve independently part of a distributed technological problem, Also it is necessary to exchange information with other participants through the events or messages generation to

978-1-7281-2572-5/20/$31.00 ©2020 IEEE

solve the entire problem. Conventionally, such devices can be called Functional Networking Components (FNC). This class of devices will include devices that have the ability to receive and process information (the presence of a computer with network ports), having a physical nature (mechanics) and capable of affecting the surrounding physical environment.

The paper is organized as follows. Section 2 analyzes related to event-based cooperation work. Section 3 discusses the problems of software control of distributed technology systems. An example implementation of an event and message generator is discussed in 4. Finally, the conclusion is given in section 5.

II. Related Work

Over the last decade the rapid development of microelectronics and information technology created the basis for the emergence of autonomous network components both in the manufacturing sector and in everyday life. Among the representatives of new network devices, autonomous cars, unmanned aerial vehicles and autonomous sea vessels can be distinguished [6]. The appearance of these devices has further exacerbated the problem of safe interaction between them and the distributed control problem. A separate problem is the interaction of these devices with a person.

The book, which has already become a classic on the topic of "Distributed Event-based Systems" [7], summarized the event-based approach of interaction between the components of a distributed system. Event-style provides an easier way to integrate autonomous, heterogeneous components into complex systems that can be scaled. Event-based architecture is becoming an integral part of the design of large-scale distributed systems, and many applications and their underlying infrastructures include event-based communication mechanisms. In [8–10], methods for controlling autonomous network devices using the theory of multi-agent control systems are analyzed. The authors proposed solutions that use the tools of predictive control in the framework of the proposed model, distributed control and cooperation of devices through events.

In [11], the cooperation possibilities of technological subsystems through the general timing system of accelerators complex are analyzed. Subsystems are clusters of functional devices in which events and messages arrive at a specific time clock. This allows us to synchronize the devices operation within the cluster. The proposed solutions allow the exchange of events and messages through specialized controllers in real time. The controller is a reconfigurable computing platform built on the basis of Field Programmable Gate Array (FPGA) as the main processing elements.

An example of using an intelligent power converter as an FNC device is presented in [12–16]. The authors substantiated the use of a deterministic local network as part of the FNC communication ports for the automation of fast technological processes when a real-time reaction of the FNC calculator to events is required. In this case, the use of a LAN for interaction purposes is not always justified, since the LAN is usually used for system purposes: downloading configurations, implementing cloud technologies, etc.

In conditions of strong electromagnetic interference from technological equipment and external disturbances, the use of wireless technologies to solve these problems is not always feasible. Therefore, additional fast wired local communication is necessary at the level of the control board, which would not occupy the resources of the microcontroller. In addition, for the reliability of the transmitted data and to increase the overall reliability of the systems, it is necessary to use a special encoding at the signal level to transmit events and messages. This solution requires the use of a combined computer as part of the FNC, that is, together with a traditional controller (for example, programmable logic controller (PLC), controllers based on the ARM core), it is necessary to use a computer on a platform, such as programmable matrices (FPGA).

III. Hardware & Software FNC

Considering the FNC as a network component that performs technological function, we primarily analyze the FNC mechanisms that allow us to process input information from sensors and generate events and messages. The software choice to develop distributed applications plays a great role.

A. FNC Software

Traditionally, the automation of technological systems used calculators on the PLC platform, which are part of subsystems and components. The programming of these calculators was successfully aided by the International Electrotechnical Commission (IEC) 61131-3 standard. The adoption of this standard unifies the development languages of control applications for PLC, which made it possible to port developed projects to PLC of various manufacturers. With this model of managing distributed interacting components, any change in the control software in one PLC can result in editing the rest of the software.

In 2005, IEC adopted the first version of the new IEC 61499 standard [17], which defined a way to build control systems of distributed technological process. The architecture of IEC 61499 is based on the definitions of IEC 61131-3. The architecture is based on Function Block (FB) with advanced interface capabilities. One of the main extensions to FB is the event interface, which allows us to explicitly define FB.

Each FB may contain several algorithms encapsulated, to which there is no direct access from other FBs. That is, event processing and algorithm execution in FB are not explicitly related to each other.

The control system, developed on the basis of the IEC 61499 standard, is a set of devices that communicate with each other through a communication network. The control system implements the functions described using applications. These applications can be distributed among several devices, which can be used as PLC, programmable automation controller (PAC) [18] or digital computers on the FPGA platform.

Figure 1 shows the IEC 61499 application model, which consists of function blocks FB1, FB2 ... FBn, connected by

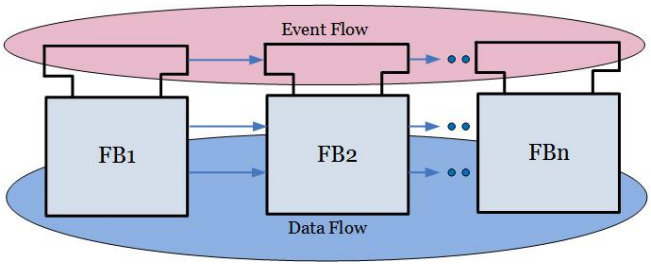

Fig. 1. Data and event flows in the application model IEC 61499

a network through which event and data streams circulate. FBs may be in certain states defined by the FB functioning algorithm. The transition from one state to another can be triggered by an event coming from a neighboring FB. All configuration changes of the event flows of a distributed system are performed at the upper level (not in each FB), which allows simple reconfiguration and scaling.

FNC programming, which uses FPGAs, requires the use of hardware description language (HDL) and an environment for designing and debugging projects on FPGAs. In [19], the advantages of using the FPGA platform for real-time synchronization of mechatronic modules through Industrial Ethernet Communications are analyzed. Standard Ethernet is non-deterministic due to the basic Carrier Sense Multiple Access with Collision Detection CSMA / CD. All industrial Ethernet protocols solve this problem by applying modifications to the Ethernet protocol and frame processing at the OSI-2 level. Modified Layer 2 protocols, by definition, require special equipment, such as Application-Specific Integrated Circuit (ASIC) or FPGA. The main industrial Ethernet protocols currently in use are Modbus/TCP [20], PROFINET RT/IRT [21], Ethernet POWERLINK [22], EtherCAT [23] and Sercos III [24].

Block diagram example of the implementation of the Ethernet POWERLINK protocol on an INTEL FPGA device of the Cyclone IV family is shown in Figure 2. In this configuration, the POWERLINK slave consists of the following main hardware components: quartz crystal oscillator (50 MHz), Ethernet physicals (PHYs), RAM, PROM, debugging interface (JTAG), diagnostic LEDs.

The Ethernet POWERLINK protocol implemented on FPGA provides maximum system flexibility and performance. An additional advantage of this implementation is the ability to embed IP cores of a mechatronic engine control system or a fast-local trunk controller [11].

B. Generation events and messages

One of the main devices for organizing FNC interaction is an event and message generator (EMG). The generator's tasks include converting information from sensors into encoded messages and events. In addition, EMG generates synchronization series for the implementation of the FNC measurement and control functions. Consider the simplified generator structure used in conjunction with the controller as part of the technological subsystems of the charged particle accelerator on the FPGA platform [11]. EMG can solve a wide range of tasks in synchronizing high-resolution processes and managing process parameters in accordance with a given law, namely:

- Generate events and messages in the established format for transmission both over global and local networks.
- Generate single clock pulses or regular clock series.
- Form a sequence of regular synchroseries with different discreteness.

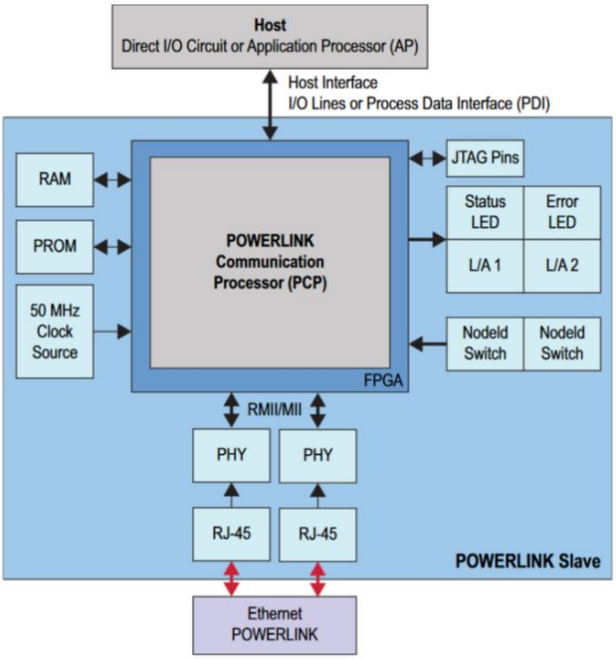

Fig. 2. An example of the POWERLINK Communication Processor implementation in the form of Soft-IP-Core on the integrated microcontroller NIOS II [19]

- Distribute the generated single pulses or regular synchronization series to the given outputs.

The EMG module can be considered as a digital device consisting of two parts: an operating unit and a control machine. Any command or operation performed in the operation unit is described by some program (microprogram) and is implemented in several clock cycles, in each of which one or several microoperations are performed. The time interval for the performance of micro-operations will be considered the working beat or tact of a digital device.

If we consider a microprogram device as a finite-state machine, then it functions as a Mealy machine with output signals delayed by several clock cycles. Arguments of transition functions are input variables recorded in a micro-command and setting the automaton state by their values. The value of the input and output signals and the firmware state of the control device must be unchanged during the cycle and changes only in pauses. The microcommands used in the EMG are microcommands with direct addressing. In this case, the micro-command does not contain the operand address, but the operand itself (or several operands, for example OPND'T ', OPND'N'), which reduces the processing time of the micro-command. The micro-command code and operands on which certain actions will be performed is vector Each vector consists of 4 control 16 - bit words located in the firmware memory (Table I).

The EMG control part includes: FSM, Program Counter, Microcommand Memory. The operating part of the EMG consists of the following components: Clock Generator,

Counter_T (period between pulses of the clock series), Counter_N (number of pulses), multiplexers and output masks.

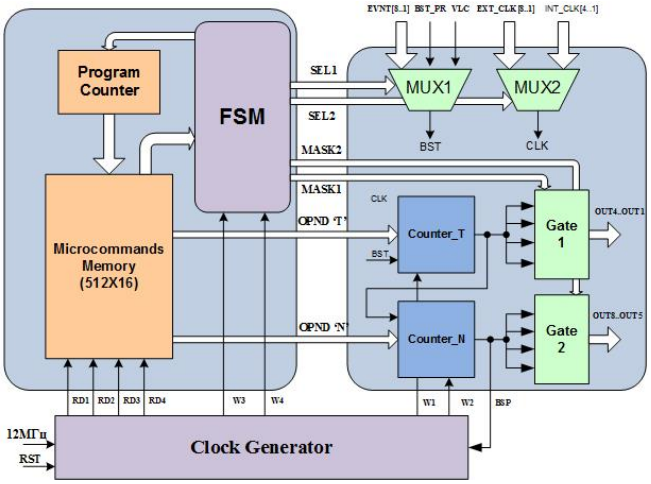

Fig.3. Block diagram of the event and message generator

TABLE I. GENERATION EVENTS AND MESSAGES

№	Encoding of events and messages		
	Program memory address	*Parameter*	*Vector*
1	Address1	Code OPND'T'	Vector 1
2	Address2	Code OPND'N'	Vector 1
3	Address3	Code SEL1, SEL2, MASK1, MASK2	Vector 1
4	Address 4	Code Microcommand	Vector 1
….	……	……	…..
511	Address 511	Code parameter SEL1, SEL2, MASK1, MASK2	Vector 128
512	Address 512	Code Microcommand	Vector 128

The memory of micro commands consists of two blocks (256x16 words), in each of which a separate program can be recorded. This allows us to divide the generated events into two groups and quickly switch them. Program switching is carried out by an external or internal PSW pulse - program switching (not shown in the figure). Dividing the memory into two independent blocks helps to use EMG channels when working with FNC in the "inside batch programming" mode. If it is intended to use both memory blocks as a unit, then the PSW signal is disabled.

The program initialization (the initial vector loading) is done either by an external RST pulse (restart), or by a special command from the controller (BST_PR). This starts the Clock Generator, which generates signals for reading (RD1..RD4) codes from the memory of microcommands and writing (W1..W4) codes in the corresponding operational registers.

The Microcommands execution sequence defined by vectors is set by Program Counter (PC). When a program is initialized, a zero address is always entered in the PC. Each vector may contain a transition address, which is loaded into the Program Counter from the control word of the Microcommand code of the current vector. If there is no transition address, then the PC is incremented by the required value, and the next vector is loaded. The jump address is used when executing microcommands that require changing the program course.

These microcommands include JMP and PLP. The JMP Microcommand, which is an unconditional Microcommand, performs actions related to the continuation of the program not from the next address, but from the transition address. The micro-command PLP (program loop) is necessary to execute the selected set of micro-commands a specified number of times. This operation allows us to save micro-command memory by setting the beginning of the selected set in the transition address. In each channel, it is possible to organize a program loop, which is a procedure for executing a set of microcommands a specified number of times.

The RES (reset) Microcommand, contained in a vector, resets all the registers and switches to the starting memory address. Thus, the firmware completes and enters the standby mode of the next start.

IV. Conclusion

The definition of a new class of network devices Functional Networking Components (FNC) is proposed. The FNC have the ability to receive and process information (the presence of a computer with network ports), have a physical nature (mechanics), and are able to affect the physical environment.

The methods and approaches of event-based FNC cooperation are analyzed. A FNC key component on the FPGA platform in event-based cooperation in a distributed technological system is the EMG event and message generator. The requirements and tasks are formulated, which include the information converting from sensors into encoded messages and events. In addition, the EMG problem is the generation of synchronization series for the implementation of the measuring and control functions of the FNC. The EMG diagram and the operation algorithm are proposed.

References

[1] V.Komarov, G.Antonichev, L.Kim, V.Kokovin and etc. "Modernization of U-70 general timing system", Proceedings of ICALEPCS-2005, Geneva, Switzerland, October 10-14, 2005.

[2] V. Kokovin "Analysis of the dependence of algorithms interrelated MANAGEMENT PROCESSES", *System analysis in science and education,* no.4, 2015, pp. 23-27, http://sanse.ru/.

[3] B. Javed, M. W. Iqbal, H. Abbas, "Internet of Things (IoT) Design Considerations for Developers and Manufacturers", *ICC2017.*

[4] IIoT Companies (Online) Available: https://www.postscapes.com/iiot-companies/ (accessed Dec. 10, 2019).

[5] P. P. Ray, "Internet of robotic things: Concept technologies and challenges", *IEEE Access*, vol. 4, pp. 9489-9500, Jan. 2017.

[6] (Online) Available: https://www.wsj.com/articles/norway-takes-lead-in-race-to-build-autonomous-cargo-ships-1500721202 (accessed Dec. 10, 2019).

[7] Gero Mühl , Ludger Fiege , Peter Pietzuch, Distributed Event-Based Systems, Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[8] R. P. Jain, A. Alessandretti, A. P. Aguiar, J. B. De Sousa, "Cooperative moving path following using event based control and communication", *Proc. 13th APCA Int. Conf. Autom. Control Soft Comput.*, pp. 189-194, Jun. 2018.

[9] Nguyen T. Hung, Antonio M. Pascoal, "Cooperative path following of autonomous vehicles with model predictive control and event triggered communications", *6th IFAC Conference on Nonlinear Model Predictive Control*, vol. 50, pp. 8692-8697, 2018.

[10] E. Garcia, Y. Cao, H. Yu, A. Giua, P. Antsaklis, D. Cas-beer, "Decentralised event-triggered cooperative control with limited

communication", *International Journal of Control*, vol. 86, no. 9, pp. 1479-1488, 2013.

[11] Kokovin V.A., Komarov V.V. "Timing Network Controller of IHEP accelerators complex" *Instruments and Systems: Monitoring, Control and Diagnostics,* Moscow 2005, no. 6, P.15-20.

[12] V. Kokovin, V. Diagilev, S Uvaysov and S. Uvaysova, "Intelligent power electronic converter for wired and wireless distributed applications", In Proc. of the IEEE International Conference SED-2019, 2019, pp. 1-5, doi: 10.1109/SED.2019.8798455.

[13] Alexander Kusko and Marc T. Thompson, *Power Quality in Electrical Systems*, New-York: McGraw-Hill, 2007.

[14] V. Diagilev, V. Kokovin and S. Uvaysov, "Power converter" Russian Federation Patent RU153221, 10 Jul., 2015.

[15] V. Diagilev, V. Kokovin, S. Uvaysov and S. Uvaysova, "Computer simulation of the power converter with harmonic wave output", *Information Technologies*, vol. 22, no. 4, pp. 261-266, 2016.

[16] V. Diagilev, V. Kokovin and S. Uvaysov, "Power converter", Russian Federation Patent RU134717, 20 Nov., 2013.

[17] International Standard IEC 61499. Function blocks for industrial-process measurement and control systems. Part 1: Architecture / International Electrotechnical Commission. – Geneva, 2005. – 245 p.

[18] *Programmable Automation Controller.* (Online) Available from: http://www.cannonautomata-products.com/programmable-automation-controller.html (accessed Dec 10, 2019)

[19] S. Germanos, Synchronizing Mechatronic Systems in Real Time Using FPGAs and Industrial Ethernet Communications. 2015. (Online) Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01249-synchronizing-mechatronic-systems-reatime-using-fpgas-and-industry.pdf (accessed Dec 10, 2019).

[20] Modbus Organization. (Online) Available from: http://www.modbus.org/ (accessed Dec 10, 2019).

[21] PROFIBUS and PROFINET International. (Online) Available from: https://www.profibus.com/ (accessed Dec 10, 2019).

[22] Ethernet POWERLINK Standardization Group (EPSG). (Online) Available from: https://www.ethernet-powerlink.org/ (accessed Dec 10, 2019).

[23] EtherCAT Technology Group. (Online) Available from: https://www.ethercat.org/default.htm (accessed Dec 10, 2019).

[24] SErial Real-time COmmunication System. (Online) Available from: https://www.sercos.org/ (accessed Dec 10, 2019).