

Analysis of possible SDN use in the rapid prototyping process as part of the Industry 4.0

D. MAZUR^{1*}, A. PASZKIEWICZ², M. BOLANOWSKI², G. BUDZIK³, M. OLEKSY⁴

¹Department of Electrical and Computer Engineering Fundamentals, The Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

²Department of Complex Systems, The Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

³Department of Machine Design, The Faculty of Mechanical Engineering and Aeronautics, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

⁴Department of Polymer Composites, The Faculty of Chemistry, Rzeszow University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

Abstract. The article refers to the idea of using the software defined network (SDN) as an effective hardware and software platform enabling the creation and dynamic management of distributed ICT infrastructure supporting the rapid prototyping process. The authors proposed a new layered reference model remote distributed rapid prototyping that allows the development of heterogeneous, open systems of rapid prototyping in a distributed environment. Next, the implementation of this model was presented in which the functioning of the bottom layers of the model is based on the SDN architecture. Laboratory tests were carried out for this implementation which allowed to verify the proposed model in the real environment, as well as determine its potential and possibilities for further development. Thus, the approach described in the paper may contribute to the development and improvement of the efficiency of rapid prototyping processes which individual components are located in remote industrial, research and development units. Thanks to this, it will be possible to better integrate production processes as well as optimize the costs associated with prototyping. The proposed solution is also a response in this regard to the needs of industry 4.0 in the area of creating scalable, controllable and reliable platforms.

Key words: rapid prototyping, software-defined networking, industry 4.0, computer networks.

1. Introduction

The rapid prototyping (RP) process (including mechanical parts) is currently one of the most important aspects of the design and manufacturing process of new products launched on the market. Globalization and high competition forces manufacturers to develop innovative solutions immediately. It also involves the need to cooperate with other business entities and scientific units [1, 2]. The analysis of processes taking place on the contemporary market of producers clearly shows that at present it is less and less possible to carry out the full design and prototyping process in the area of one company or a single branch within a given consortium. Often units involved in prototyping are geographically dispersed forcing the use of modern communication technologies to ensure fast and reliable data exchange between them. A perfect example of such relationships are producer clusters, special economic zones, etc. The ideal solution for integrating dispersed resources are therefore modern computer networks.

It should also be noted that currently, when we are at the threshold of another industrial revolution, it is necessary to ensure compatibility of new solutions with industry 4.0 [2]. A di-

agram illustrating some selected elements that should be integrated with each other within Industry 4.0 is presented in Fig. 1.

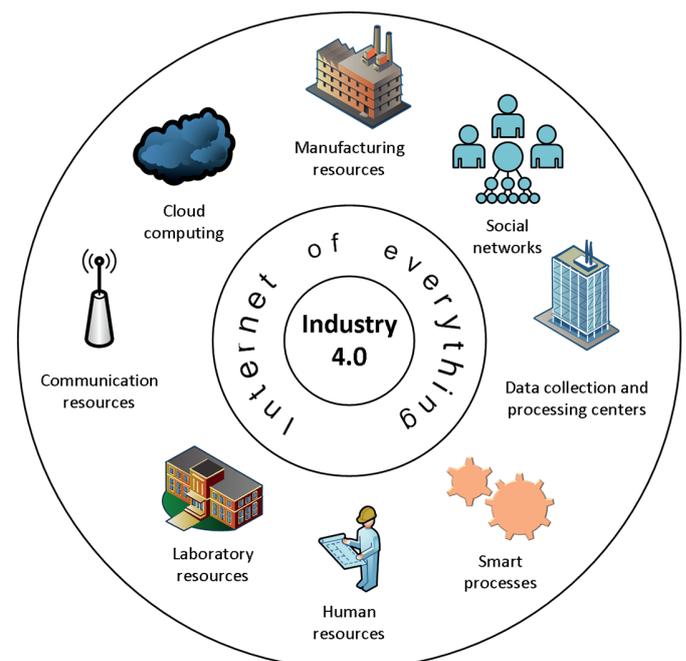


Fig. 1. Industry 4.0

*e-mail: mazur@prz.edu.pl

Manuscript submitted 2018-02-25, revised 2018-04-22, initially accepted for publication 2018-04-26, published in February 2019.

Therefore, integration of manufacturing processes going beyond individual production plants has already become a fact and thus it is necessary to create and develop solutions enabling cooperation between various economic entities not only at the level of IT systems integration but also technology and production. The chosen development direction in this area should increase the efficiency of prototyping and production processes. This can be achieved by using a new approach to support the process of collecting and analyzing data from various, distributed sources (machines, devices). Thanks to this, it will be possible to control of the integrated processes on the basis of solutions using the features of complex systems, e.g. feedback loops. To achieve this, it is necessary to take into account the following issues.

- **Reliable and secure communication infrastructure.** It should be intended to integrate the distributed, heterogeneous environment used in the rapid prototyping process. Nowadays, there are many threats related to the functioning of the ICT infrastructure, as well as the protection of industrial secrets. Therefore, such infrastructure must guarantee the security of transmitted data and ensure quick and reliable reconfiguration of logical connections, and in many cases also physical connections. This involves both the use of appropriate hardware solutions as well as the development and implementation of the necessary security mechanisms, guaranteeing the integrity and confidentiality of data transmitted between the various elements that are part of the prototyping and production process.
- **Monitoring of the manufacturing process.** In the modern high-tech industry, the quality of manufactured products and sub-assemblies, both final and prototypes, models, etc. plays an important role. This is particularly important in a distributed environment based on remote work [3]. Therefore, in the process of rapid prototyping it is necessary to use tools supporting the process of monitoring and control of manufacturing processes located in remote locations. Considering the diversity of manufacturing technologies, this issue requires in many cases an individual approach. Of course, one can look for and try to implement also universal solutions. An example of this can be registration and then analysis of images in high resolution. This approach allows for the initial verification of the conformity of the real object with the digital model [4]. However, this issue is not discussed by the authors in this paper.
- **Control and synchronization of work.** Personalized activities in this area based on technical staff are characterized by low efficiency, which is also associated with relatively long response time, low flexibility in managing available resources, and thus limits the possibility of using a methodology of continuous work on the prototype e.g. follow-the-sun or RBAC [5, 6]. Thus, it is necessary to develop a methodology for managing a distributed prototyping process, taking into account the limited participation of technical personnel. A central and automatic mechanism for controlling the prototyping process based on a program arbitrator may be an effective solution in this area. This architecture also allows the implementation of

the advanced control mechanisms, such as those described in the papers [7–9].

The process of distributed rapid prototyping is extremely complex. Thus, solutions to a number of problems should be found before its complete implementation. The authors of this paper focused on proposing a model of rapid distributed prototyping with particular emphasis on the control of the communication environment, which is built on the basis of the software defined networking model.

The second chapter will describe the general scheme of the rapid prototyping process and the proposed network model of distributed prototyping using computer networks. The third chapter presents the model of using the SDN idea to carry out tasks in the area of the distributed rapid prototyping system. The fourth chapter is devoted to the presentation of the results of ongoing work, including the presentation of the developed physical and logical infrastructure of the system. The summary will present the directions of further research and the potential possibilities of the proposed model in the production environment.

2. The process of rapid prototyping in the infrastructure of computer networks

The rapid prototyping process consists of several basic elements (stages) which, depending on the technology used, may slightly differ from each other [10]. However, its basic outline is presented in Fig. 2. The first stage of rapid prototyping is always associated with conceptual work and the preparation of an initial 3D or 2D model. In the next steps, a digital model is developed, usually with specialized CAD applications, both of general purpose and dedicated to specific solutions. At this stage, the data regarding the modeled object has been already verified and possible corrections are made in relation to the initial assumptions obtained at the stage of concept development. It should be remembered that any changes introduced at this stage save time and limit possible costs related to later corrections. It should be noted that currently there are many methods of physical object production in the rapid prototyping process, for example, stereolithography (SLA), layering and curing of the photopolymer (PolyJet), fused deposition modelling (FDM), selective laser sintering of metallic powders (SLS), three-dimensional printing (3DP) [11–13]. However, in each of these cases it is required to “transfer” (export) the data describing the modeled object to the physical production devices. Then the process of physical creation of the final object begins, which involves precise specification of technological parameters, calibration of devices, etc. A very important aspect of this stage includes quality control, which may take place at its end or during the whole production process. If the final object does not meet expectations the return to one of the previous stages should take place and the appropriate adjustments should be done. Sometimes this requires verification of the very concept and initial assumptions.

The contemporary nature of production systems, in particular those implemented in the Industry 4.0 architecture, has the characteristics of a distributed system. The process of information exchange between nodes is crucial for this class of

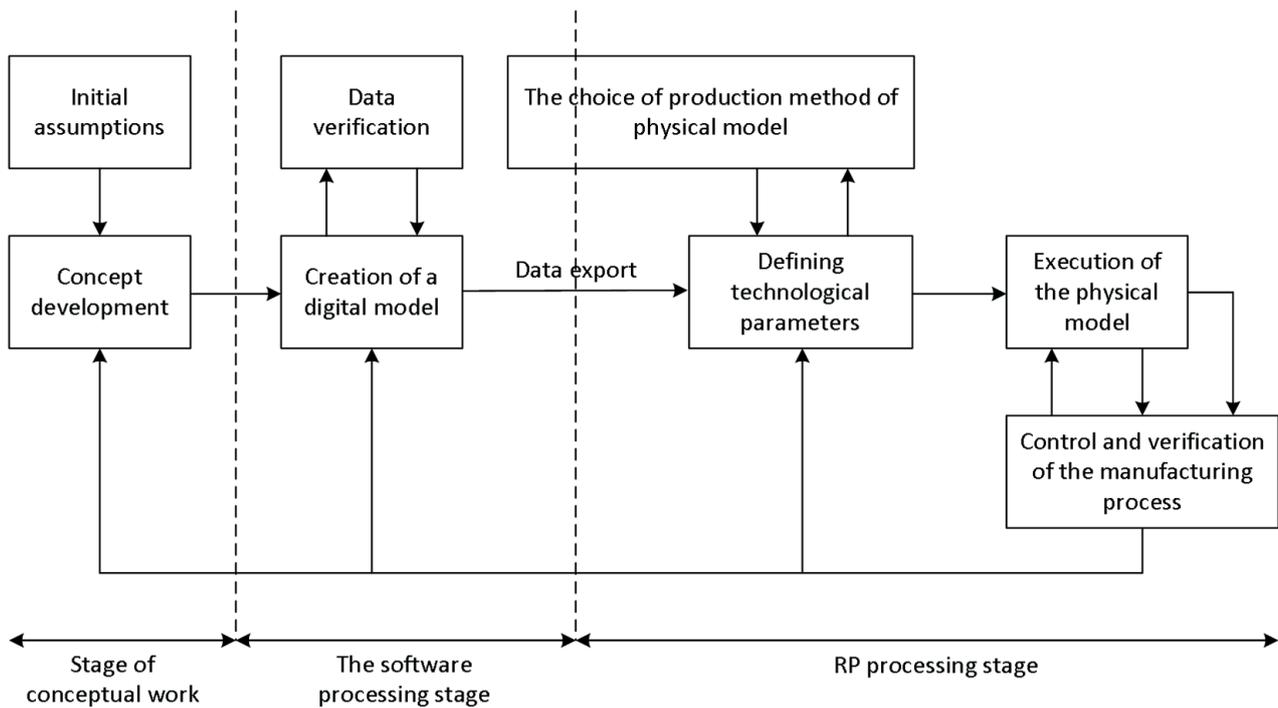


Fig. 2. Diagram of the rapid prototyping process

systems. Interestingly, a node in the case of rapid prototyping systems corresponds to the individual stage in Fig. 2. Until now, information exchange between particular stages was implemented in a centralized architecture and its implementation was not a serious challenge. Often, systems of this class were also characterized by homogeneity of the applied solutions. One of the basic requirements of Industry 4.0 is the ability to combine resources from various locations within one task, in which devices constructed by various manufacturers supporting various communication standards are located. This task can be implemented using an efficient and reliable computer network. Modern computer networks have commonly mechanisms corresponding to the requirements of industrial systems, including: QoS, encryption mechanisms, adequate level of reliability, clock synchronization protocols, etc.

An exemplary diagram of a classic network infrastructure allowing the integration of distributed elements involved in the rapid prototyping process has been presented in Fig. 3. Thanks to the use of broadband computer networks it is possible to allocate individual components in remote locations while maintaining the appropriate transmission quality parameters.

However, the classic approach to computer network management has strongly reactive nature, which is characterized by, among others:

- **A reactive management model.** In this model, individual actions are initiated by end users or production processes supervised by technicians. After receiving the message, the technical staff managing the information system (including a computer network) designates a service window within which actions related to the reconfiguration of the connection network will be carried out. This classic approach to man-

agement prevents quick reconfiguration of the system. It also significantly raises the amount of time needed to assemble a cluster of production equipment dedicated to the implementation of a given project or task execution. Additionally, for activities carried out in follow-the-sun mode, it is necessary to employ a team managing the network in 24-hour mode.

- **Closed architecture of automatic management systems.** There are solutions available on the market that allow dynamic change of parameters of computer networks. Thanks to this, it becomes possible to actively manage the infrastructure, including automatic reconfiguration of connections to set up a cluster of devices. However, these systems

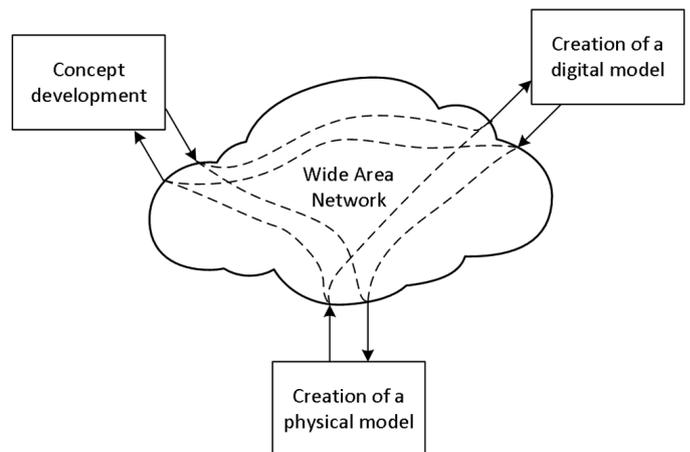


Fig. 3. Diagram of network infrastructure supporting the rapid prototyping process

are very expensive and usually dedicated to a homogeneous device environment. Unfortunately, their application in a real distributed environment of network devices from different manufacturers is limited, and in some cases even impossible. At some stage, attempts were made to create their own dedicated management systems. The main problem during their construction was the lack of a unified, standardized management protocol common to network devices from different manufacturers. The costs of implementing and maintaining such solutions were also too high.

The above restrictions significantly hinder and sometimes even prevent the creation of an efficient and reliable distributed communication system supporting rapid prototyping processes. The use of software defined networking in the computer network environment may be helpful here.

3. The concept of SDN network in the context of rapid prototyping

The concept of an SDN network has been described in detail in publications [14–16]. The general approach to managing the communication environment distinguishes it from the classical model. Figure 4 presents a model of a simple communication system that connects several L1 to Ln locations. In each location, the communication system is created by network devices

(ND) that connect elements used in the rapid prototyping devices infrastructures (RPDI) process. Individual locations can be seen as separate departments of a given company, teams of engineers, various enterprises, etc. The role of a computer network is to ensure cooperation between them. Therefore, the communication environment must change quickly and create secure communication channels on demand. In the classic topology shown in Fig. 4a, each network device operates with its own autonomous control system (Control Plane) and data transmission (Data Plane). In the optimal case, the whole environment is managed from one place by a specialized network management system (NMS) application, which collects information from devices, presents them and reactively manages them in the case of a homogeneous structure. The data flows are marked with a continuous line, and the route of sending control communication is marked with a red dotted line. The reactivity of the management approach in this case is usually associated with the concept of SLA (service level agreement). The work group or data channel set up time can be described with the following formula:

$$T_s = t_z + t_{SLA} + t_p + \sum_{i=1}^n t_{r_i}, \quad (1)$$

where: T_s means the total duration of the channel configuration process or reconfiguration environment, t_z – the time required to

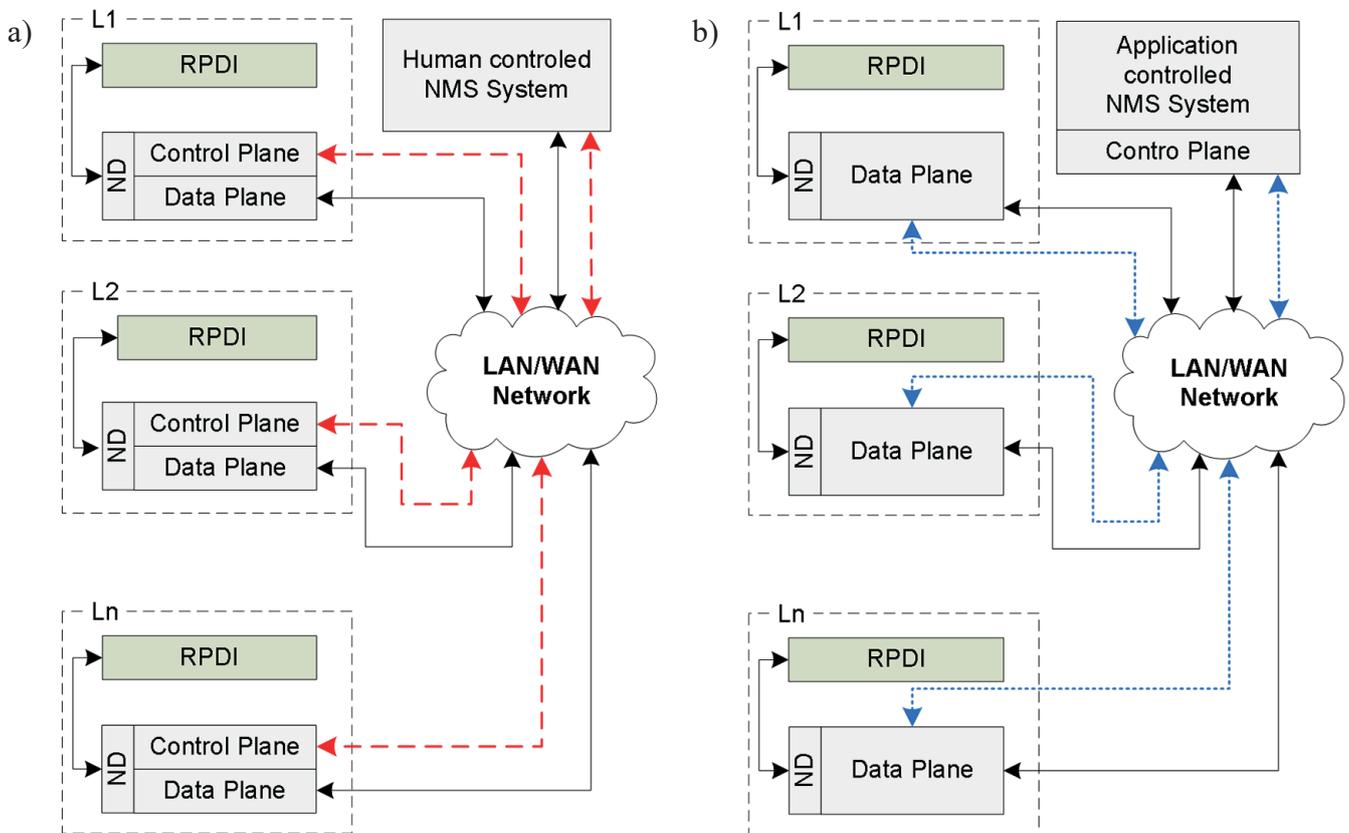


Fig. 4. Control models of connection network: a) classic; b) SDN

report the need for reconfiguration (from several minutes to several hours), t_{SLA} means the time that the IT team (or maintenance company) has guaranteed in the contract to complete the task (4 to 24 hours), t_p – the time needed to gather configuration data and their presentation to the IT personnel (if the data analysis is included, this time can range from several dozen minutes to several hours), $\sum_{i=1}^n t_{r_i}$ – time needed to reconfigure individual devices in given locations (few minutes per device).

Figure 4b presents the assumptions of the SDN architecture. The data flows are marked with a continuous line, and the blue dotted line marks the routes of transmission of the control communication. It should be noted that in the classical case, as a rule, there are several dozen incompatible protocols of data collection and control in the network. In the case of the SDN model, the Control Plane layer is transferred from the device to the controller in which it is implemented as a software. The controller injects flows directly to the devices. One protocol (OpenFlow) [17] is used to control network devices (even in the case of a heterogeneous environment). The system architecture, and in particular its NMS system allows its direct integration with applications. Thanks to this, the API can be made available to the end user to control network resources using pre-prepared patterns. In this case, we can talk about an active network management model. In many cases, there will be no need for IT staff to reconfigure the network (no need to include SLA time). These activities will be implemented automatically. Taking the above into consideration, formula (1) will take the form:

$$T_s = t_{zu} + t_p + \sum_{i=1}^n t_{ra_i}, \quad (2)$$

where: t_{zu} – is the time needed to independently initiate infrastructure and topological changes in the network, t_{ra_i} – time needed to reconfigure devices with the assumption of flow level management (flow injection). It should be noted that $t_z \gg t_{zu}$ i $t_{r_i} \gg t_{ra_i}$.

Thanks to the use of SDN model in a distributed rapid prototyping environment, it becomes possible to significantly accelerate and simplify the process of decision-making and data migration needed to complete the subsequent stages of work.

4. RDRP model

Considering the above-described issues, the authors proposed a new hierarchical model called the remote distributed rapid prototyping model (RDRP) (Fig. 5a). This model allows referring to all mechanisms, processes and technologies accompanying rapid prototyping in the environment of distributed IT networks and in the computing cloud. Components of a distributed system can be individual devices, computers but also entire production lines. Let us assume that each of these independent elements will be called a RDRP entity. As a rule, entities are manufactured by different manufacturers and subcontractors. The use of the model allows to clearly define which elements must be implemented in a given entity in order to be

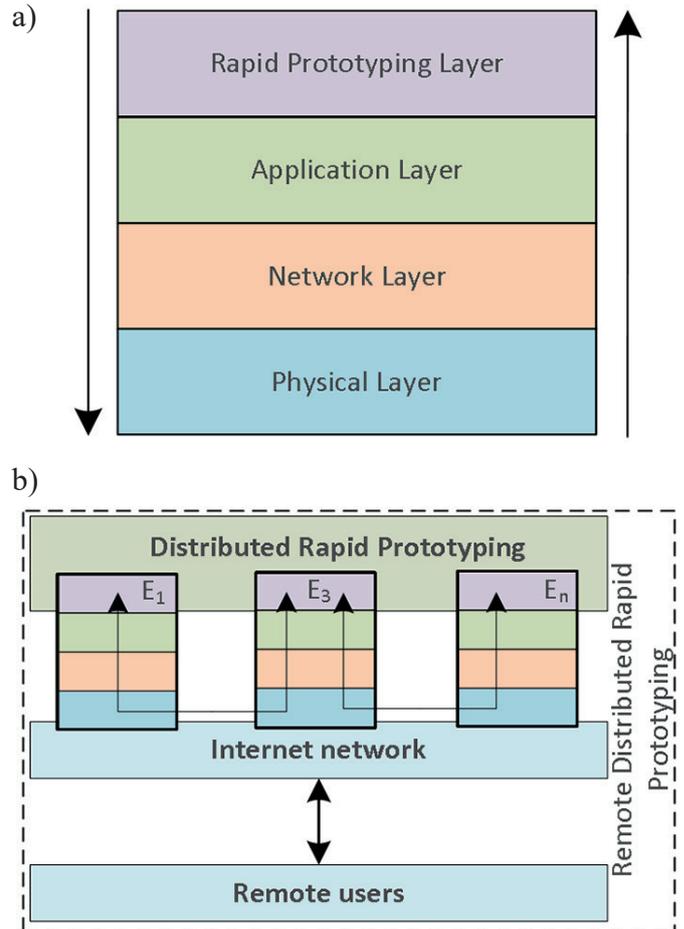


Fig. 5. Industry 4.0 a) Remote Distributed Rapid Prototyping Model; b) RDRP model from the point of view of the distributed system

able to communicate with other entities in RDRP. The producer of a given entity element focuses only on fulfilling the requirements of a given layer. Processes residing in individual layers of the RDRP model communicate with their counterparts in the same layer located in different entities (Fig. 5b). Thanks to the proposed model, manufacturers of appropriate devices, e.g. for the physical layer, no longer have to produce complicated network control systems for their devices. This is what the companies creating systems at the application layer are dealing with. Communication for them is provided by specialized devices appropriate for the network layer. Thus, rapid prototyping processes communicate with each other among the entities deployed in the distributed network infrastructure. In the same way, applications or network protocols communicate with each other on the appropriate layer. In contrast, data flow at individual nodes is carried out from top-down and bottom-up, with a restriction on the layer being processed on a given node, of course. As a result, the rapid prototyping layer approach from individual entities (see Fig. 5b) creates one distributed prototyping system that is made available to remote users, thus creating remote distributed rapid prototyping (RDRP). This approach is beneficial for cost, security and implementation time.

The proposed RDRP model consists of 4 layers:

1. **the physical layer** consists of real components that create both network infrastructure (intermediate communication devices) as well as end devices, including workstations, manufacturing devices (e.g. 3D printers), etc., as well as wired and wireless communication infrastructure necessary to ensure continuity of data transmission. Currently, most industrial devices have network interfaces. For those that do not have them, systems for data collection are created that have the ability to communicate by IP protocol. Adaptation of devices to work in this layer seems to be an easy task. In addition, elements that are part of the Internet of Things may function in this layer
2. **the network layer** is responsible for managing network mechanisms. At the level of this layer, network device configurations are created. This enables reliable and secure communication, including mechanisms for quality assurance (QoS), creation of logical topologies, and defining parameters of secure (including encrypted) communication channels. Communication protocols such as IP also function in this layer. It should be clearly stated that this layer is non-hardware based. Network management systems will reside here whose main task is to create an API for the application layer to manage the communication environment. It should also be noted that virtual communication devices such as those used in the network functions virtualization model [18] will be located outside this layer in the physical layer.
3. **the application layer** is responsible for communication between individual applications on the remote devices connected to the computer network. Data is collected, processed and visualized in this layer. The applications residing in this layer are also used to model prototype objects and optimize their parameters and technological processes. The key to this layer is the detailed definition of standards and formats used by each application to ensure interoperability. Of course, for a given RDRP instance, the set of protocols used may be unique. At this stage, we can use the concept of protocol compliance domains for which a specific set of interpreting parameters will be defined. At present, the team is conducting research in this area, and results will be published in the forthcoming articles.
4. **the rapid prototyping layer** provides a broad view of all processes related to the prototyping of individual elements, the flow of information between components that are part of the rapid prototyping process, as well as the control of all stages of this process. Thanks to this, it is possible to take an abstract look at the whole process and to visualize both technological and business flows between independent partners taking part in it. It should be emphasized that this layer is closely related to the model of rapid prototyping shown in Fig. 2.

The presented model has character of open systems. As a result, it is possible to continuously create new applications, develop new protocols, develop and introduce new manufacturing technologies in the process of rapid prototyping as well as what has already been mentioned the interaction of many different components in a high- heterogeneous environment.

The model presented above has a unique character because it combines the features of a technical model (such as for example in the ISO/OSI model [19]) with an abstract model. The hybrid nature of the model allowed for its better adaptation to the market requirements and new trends noticeable in the industry.

4.1. SDN networks for RDRP model. The first tests carried out showed clearly that the key to the proper functioning of the entire RDRP system is to ensure an adequate communication layer. Among many available architectures, the SDN model was chosen because its application enables the implementation of communication functionalities in the two lowest layers of the RDRP model. The benefits of choosing the SDN architecture are described in Section 3. In this part, the authors want to present practical conclusions from the implementation of SDN in the actual laboratory structure. Figure 6 illustrates the functional scope of individual layers of the RDRP model and relations between them using the SDN technology.

4.2. Implementation of the RDRP model based on SDN architecture in a laboratory environment. The communication environment for the proposed RDRP model was developed based on the resources of the laboratory of converged systems, the Department of Complex Systems and the Laboratory of Rapid Prototyping in Rzeszów University of Technology. Alcatel-Lucent OmniSwitch 6860E as well as the OmniSwitch 6900 were used in the communication physical layer (Fig. 7). The SDN controller was developed using Open Source software, including FloodLight [20] software and the controller architecture for complex and distributed network environments, which was described in detail in [21].

A model based on the theory of complex systems was used to create software for managing the prototyping environment [22–25]. Taking into consideration the above-mentioned technological conditions in Fig. 8, the architecture of the RDRP system has been presented, including the implemented communication system based on SDN.

It was assumed that three teams would perform their work remotely. Ultimately, based on the adopted model, the number of teams would be limited only by the availability of resources and in particular the possibility of their allocation in a given allocation unit (see Fig. 9). During the implementation of the work the following design assumptions were adopted:

- The system should ensure full isolation of resources used by individual T_1 , T_2 and T_3 teams. This isolation can be achieved, among others, by assigning units to various dedicated VLANs in the 802.1q or Q in Q standard. The essence of the operation of such a system is, among others ensuring full confidentiality of transmitted information and their consistency.
- For the purposes of the experiment, technical resources from R_1 to R_4 are available through a computer network and communicated with the SDN controller. Among them, we can distinguish two types: sequential devices – available only in a given unit of time only for one team and parallel devices – available simultaneously for a larger number of teams. In

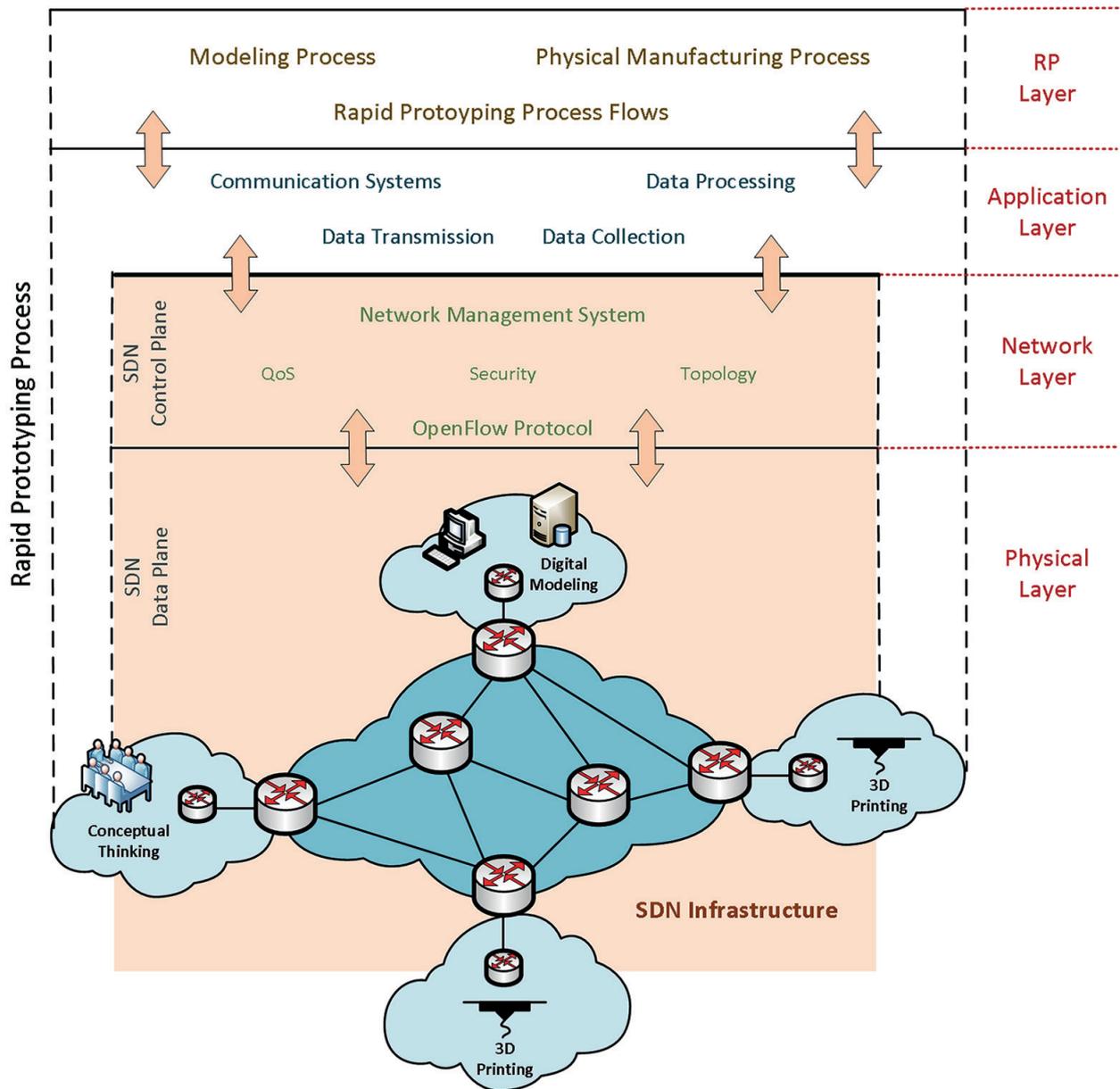


Fig. 6. Correlation of the SDN architecture with the RDRP model

the case of the R_1 resource, parallelization was achieved through the use of virtualization platforms. Thanks to this, CAD modeling software was launched on separate system instances, which allowed the independent (parallel) operation of separate teams, using a common hardware platform.

- Team requests are registered by the system via the website and supported by the scheduler who allocates resources analyzing their availability. The scheduler is dedicated to optimize the use of resources based on the analysis of planned tasks. At this stage T_1 , T_2 and T_3 are also treated as resources with a specific availability time.
- The operation of a given T_n team can be interrupted, then the current state must be stored in the dedicated memory area F_n and restored after the team has again obtained access to the system. This solution enables effective scheduling

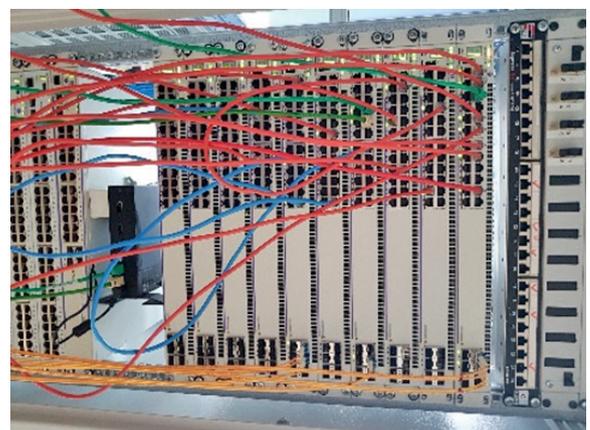


Fig. 7. SDN infrastructure in laboratory

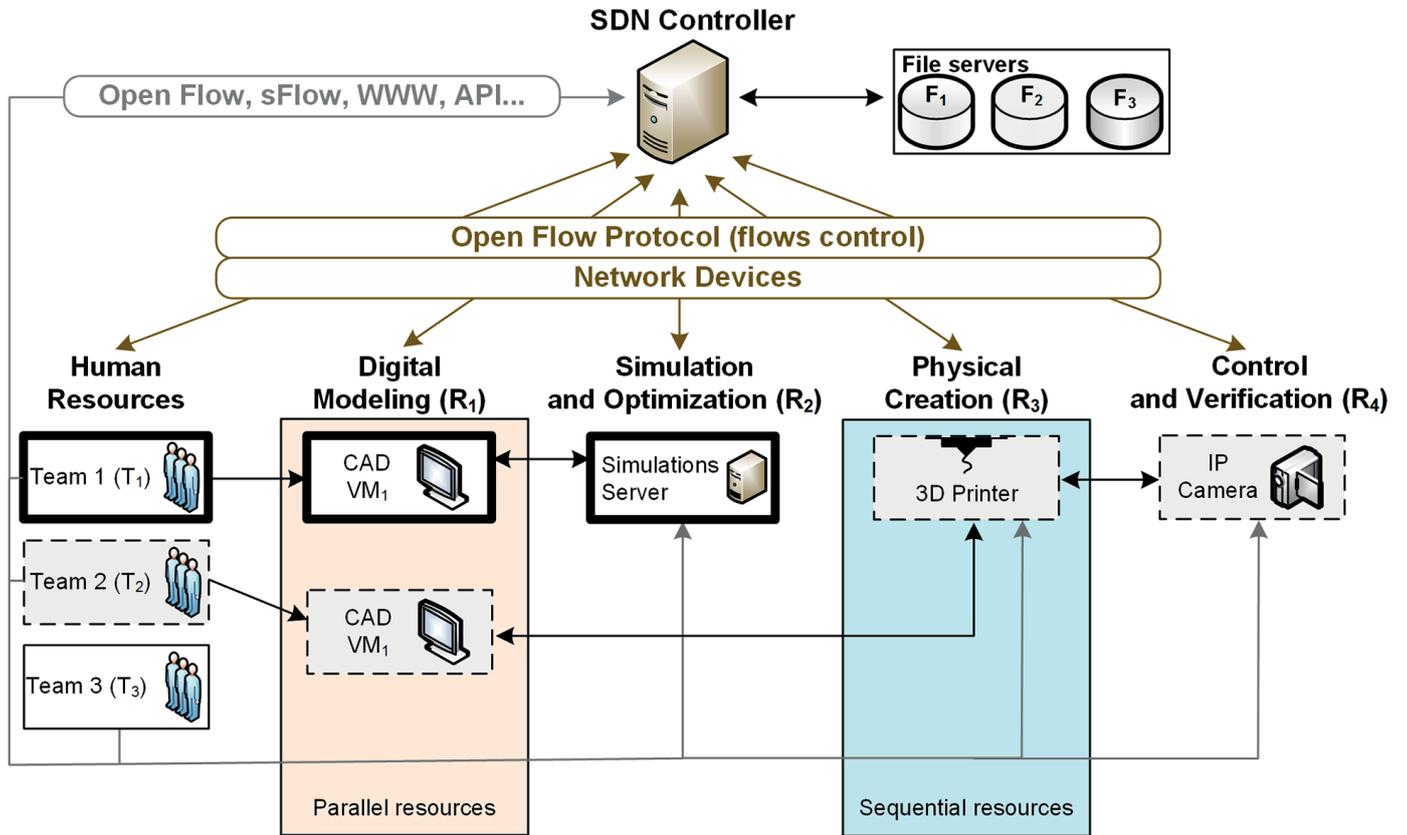


Fig. 8. Diagram of the implemented communication system for the RDRP model based on SDN

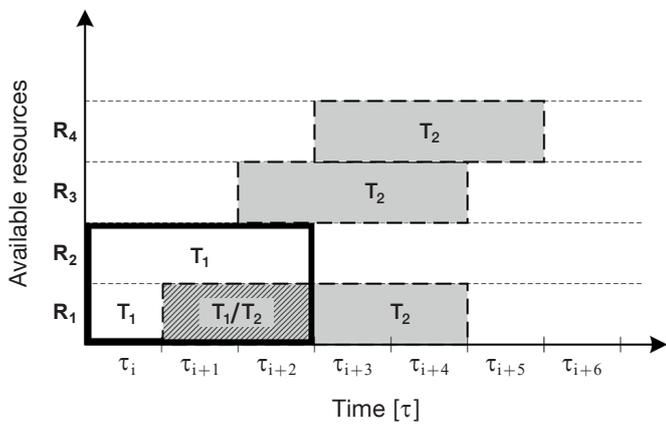


Fig. 9. Allocation of resources in time

of individual teams' work as well as more efficient use of available resources. An exception are resources that must be available until the job is completed (e.g. machine tool, 3D printer, etc.).

- The system must enable the detection of anomalous situations both in the case of industrial equipment as well as in the case of network traffic.

In the next step, the system was implemented and launched at the communication layer. During the work, the research fo-

cused mainly on the methods of automatic cluster creation for teams. An exemplary scheme of these operations is presented in Fig. 8. Two teams T_1 and T_2 used the system simultaneously, where: $T_1 \leftarrow (R_1, R_2, F_1)$ and $T_2 \leftarrow (R_1, R_3, R_4, F_2)$. Detailed allocation of resources in time τ including their sharing is shown in Fig. 9. In the case of the T_1 team, the bold line indicates the appropriate resource allocation, while in the case of the T_2 team, it is represented with dotted line and gray filling. During the work, it was assumed that the time unit τ is globally definable for the whole system and can take discrete values. The research has shown that the correct selection of the time unit value is of great importance for the efficiency of resource allocation and minimization of their inactivity time. Further research will focus on the possibility of dynamic allocation of time.

The entire system was coordinated by applications for handling requests and time allocation. The application is at the initial stage of development and the allocation of resources was carried out statically according to previously prepared patterns.

4.3. Results. As a result of a series of experiments related to the automatic allocation of resources in the RDRP model, the following conclusions were made:

- The use of SDN architecture is the most effective in consistent communication environments that have SDN devices. When using the model in a WAN environment, it may be

necessary to use a hybrid model (Pseudo SDN) or use tunneling techniques to make a part of the incompatible infrastructure “transparent” for SDN [21].

- The use of SDN architecture fully meets all the requirements set by the first and second layer of the RDRP model. Additionally, it is possible to easily integrate these layers with the application layer.
- It is necessary to develop an algorithm for automatic allocation of resources. The algorithm must be real-time or near-real-time. The optimization tasks that this algorithm will solve are well known. However, in order to shorten the time of its operation it is possible to use mechanisms borrowed from genetic algorithms [26] or algorithms based on graph association (neural networks, clustering methods, k -partite graphs), etc.
- It should also be remembered that the management of the planning process when allocating resources in the RDRP model also refers to the existence of the phenomenon of incomplete information, which is characteristic for the majority of project processes. Therefore, it is necessary in the future to develop the methods of taking this phenomenon into account for the process of allocation of resources for the proposed model.
- Simulations have shown that for the proposed architecture it is possible to use the methods of detecting anomalies in network traffic described in the works [27]. At the same time, using the appropriate adapters, it is also possible to detect anomalies in the operation of industrial equipment [28]. Information about these anomalies could be collected and gathered in a dedicated controller module, e.g. SDN, and then used for the current control of the communication infrastructure and the rapid prototyping process itself. In the future, it would be possible to develop mechanisms of prediction and counteracting anomalies for RP.

5. Conclusions

The dynamic development of industrial clusters, the integration of research centers with industry, the emergence of new start-up companies forces the creation of new models and technical solutions not only supporting design and manufacturing processes inside them, but also (and perhaps above all) supporting cooperation between them. Therefore, the authors developed a new remote distributed rapid prototyping model (RDRP). It is characterized by openness as well as heterogeneity in the area of communication and design-manufacturing infrastructure.

Of course, as indicated in Section 4 there are many areas in which further work should be carried out. They are related to, among others, network integration of various solutions not connected with each other so far and in the long term the need to integrate with Industry 4.0 and IoT. However, after the first tests in a controllable laboratory environment, the suitability, flexibility and scalability of both the RDRP model and the SDN were confirmed to increase the potential of rapid prototyping processes. Moreover, the proposed architecture allows to im-

plement of many various algorithms, for example in the image recognition area [29] which can be used in the automation of the control and verification process (Fig. 8).

REFERENCES

- [1] H. Lasi, P. Fettke, H.G. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0”, *Business & Information Systems Engineering*, vol. 6, issue 4, 239–242 (2014).
- [2] R. Strange and A. Zucchella, “Industry 4.0, global value chains and international business”, *Multinational Business Review*, vol. 25, issue 3, 174–184 (2017).
- [3] W. Bozejko, P. Rajba, and M. Wodecki, “Stable scheduling of single machine with probabilistic parameters”, *Bull. Pol. Ac.: Tech.*, vol. 65, Issue 2, 219–231, DOI: <https://doi.org/10.1515/bpasts-2017-0026>, (2017).
- [4] Z. Kuś, “Analysis of dynamical properties of object tracking system elements”, *Bull. Pol. Ac.: Tech.*, vol. 64, Issue 3, 479–489, DOI: <https://doi.org/10.1515/bpasts-2016-0053>, (2016).
- [5] J. Kroll, B.J. da Silva Estácio, J.L.N. Audy, and R. Prikładnicki, “An Initial Framework for Researching Follow-the-Sun Software Development”, *Global Software Engineering (ICGSE)*, 2014 IEEE 9th International Conference on, 2014.
- [6] Y. Fu, Y. Liu, and D. Liu, “An environment-based RBAC model for internal network”, *Computer Communication and the Internet (ICCCI)*, 2016 IEEE International Conference on, DOI: 10.1109/CCI.2016.7778884, (2016).
- [7] J. Klamka, “Controllability of dynamical systems. A survey”, *Bull. Pol. Ac.: Tech.*, vol. 61, Issue 2, 335–342, DOI: <https://doi.org/10.2478/bpasts-2013-0031>, (2013).
- [8] T. Kaczorek, “Stability of fractional positive continuous-time linear systems with state matrices in integer and rational powers”, *Bull. Pol. Ac.: Tech.*, vol. 65, Issue 3, 305–311, DOI: <https://doi.org/10.1515/bpasts-2017-0034>, (2017).
- [9] T. Kaczorek and K. Borawski, “Stability of positive nonlinear systems”, *Methods and Models in Automation and Robotics (MMAR)*, 2017 22nd International Conference on, 564–569, DOI:10.1109/MMAR.2017.8046890, (2017).
- [10] S.H. Choi and S. Samavedam, “Modeling and optimization of Rapid Prototyping”, *Computers in industry*, vol. 47, issue 1, 39–53 (2002).
- [11] M. Sobolak and G. Budzik, “Experimental method of tooth contact analysis (TCA) with rapid prototyping (RP) use”, *Rapid Prototyping Journal*, vol. 14, issue 4, 197–201 (2008).
- [12] P. Rokicki, G. Budzik, K. Kubiak, J. Bernaczek, T. Dziubek, M. Magniszewski, A. Nowotnik, J. Sieniawski, H. Matysiak, R. Cygan, and A. Trojan, “Rapid prototyping in manufacturing of coremodels of aircraft engine blades”, *Aircraft Engineering and Aerospace Technology*, 86(4), (2014).
- [13] D. Pham and S.S. Dimov, *Rapid Manufacturing: The Technologies and Applications of Rapid Prototyping and Rapid Tooling*, Springer-Verlag London, DOI: 10.1007/978-1-4471-0703-3, 2001.
- [14] S. Singh and R.K. Jha, “A Survey on Software Defined Networking: Architecture for Next Generation Network”, *Journal of Network and Systems Management*, vol. 25, issue 2, 321–374, DOI:10.1007/s10922-016-9393-9, (2017).
- [15] D. Kreutz, F.M.V. Ramos, P.E. Verissimo, Ch.E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey”, *Proceedings of the IEEE*, vol. 103, issue 1, DOI: 10.1109/JPROC.2014.2371999, (2015).

- [16] H. Kim and N. Feamster, "Improving network management with software defined networking", *IEEE Communications Magazine*, vol. 51, issue 2, 114–119, DOI: 10.1109/MCOM.2013.6461195, (2013).
- [17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks", *ACM SIGCOMM Computer Communication Review*, vol. 38, issue 2, 69–74 (2008).
- [18] G. Faraci, A. Lombardo, and G. Schembra, "A building block to model an SDN/NFV network", *Communications (ICC), 2017 IEEE International Conference on*, 114–119, DOI: 10.1109/ICC.2017.7997430, (2017).
- [19] ISO/IEC 7498-4:1989, "Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing", International Organization for Standardization, (1989).
- [20] I.Z. Bholebawa and U.D. Dalal, "Performance Analysis of SDN/OpenFlow Controllers: POX Versus Floodlight", *Wireless Personal Communications*, vol. 98, issue 2, 1679–1699 (2018).
- [21] M. Bolanowski and A. Paszkiewicz, "Methods and Means of Creation Application to Control a Complex Network Environment", *Software Engineering Research for the Practice*, Polish Information Processing Society, 151–160 (2017).
- [22] J.A.C. Ladyman, J. Lambert, and K. Wiesner, "What is a complex system?", *European Journal for Philosophy of Science*, vol. 3, issue 1, 33–67 (2013).
- [23] F. Grabowski, "Nonextensive model of self-organizing systems", *Complexity*, vol. 18, issue 5, 28–36 (2013).
- [24] F. Grabowski, A. Paszkiewicz, and M. Bolanowski, "Computer Networks as Complex Systems in Nonextensive Approach", *Journal of Applied Computer Science*, vol. 21, No. 2, 31–44 (2013).
- [25] A. Paszkiewicz and M. Bolanowski, "Software Development for Modeling and Simulation of Computer Networks: Complex Systems Approach", *Towards a Synergistic Combination of Research and Practice in Software Engineering*, Springer International Publishing, 193–206 (2017).
- [26] Z. Michalewicz and C.Z. Janikow, "Genetic algorithms for numerical optimization", *Statistics and Computing*, vol. 1, Issue 2, 75–91, DOI: <https://doi.org/10.1007/BF01889983>, (1991).
- [27] M. Bolanowski and A. Paszkiewicz "The use of statistical signatures to detect anomalies in computer network", *Lecture Notes in Electrical Engineering*, vol. 324, Springer International Publishing, 251–260, DOI: 10.1007/978-3-319-11248-0_19, (2017).
- [28] P.Y. Sevilla-Camacho, J.B. Robles-Ocampo, J. Muñoz-Soria, and F. Lee-Orantes, "Tool failure detection method for high-speed milling using vibration signal and reconfigurable bandpass digital filtering", *The International Journal of Advanced Manufacturing Technology*, vol. 81, Issue 5–8, 1187–1194, DOI: 10.1007/978-3-319-11248-0_19, (2015).
- [29] Z. Gomolka, B. Twarog, and J. Bartman, "Improvement of image processing by using homogeneous neural networks with fractional derivatives theorem", *Discrete and Continuous Dynamical Systems – Series A*, vol. 31, 505–514, (2011).