

Use of Dynamic Models in Cognitive Cyber-Physical Systems [†]

Michael Chervontsev ¹, Alexey Subbotin ^{1,*}, Alexander Vodyaho ¹ and Nataly Zhukova ^{1,2}

¹ Department of Computer Science and Engineering, Saint-Petersburg State Electrotechnical University, St. Professor Popov, 5, St. Petersburg 197376, Russia

² Saint-Petersburg Federal Research Centre of the Russian Academy of Sciences, 14th Line V.O., 39, St. Petersburg 199178, Russia

* Correspondence: ansubbotin@stud.etu.ru; Tel.: +7-921-365-6535; Fax: +7-812-440-7701

[†] Presented at the 15th International Conference “Intelligent Systems” (INTELS’22), Moscow, Russia, 14–16 December 2022.

Abstract: This article discusses an approach to the development of large-scale cognitive cyber-physical systems characterized by a high level of structural, functional and architectural dynamics. The main idea of the proposed approach is the use of several modern paradigms to build cyber-physical systems, such as continuous architecture, agile architecture, digital twins and digital threads. A three-level model of a cognitive cyber-physical system is proposed. In order to ensure the required level of flexibility of the system, such possibility should be laid at earlier stages of the life cycle, i.e., at the development stage. At the upper level, the system is described in terms of a continuous architecture; at the middle level, the system is represented as a system with agile architecture, which is described as a multi-level relatively finite automaton, and at the lower level, structural and functional models are used that illustrate the system in the process of functioning. This article provides examples of using the proposed approach.

Keywords: cognitive cyber-physical systems; dynamic models; continuous architecture; agile architecture; digital twins; digital threads



Citation: Chervontsev, M.; Subbotin, A.; Vodyaho, A.; Zhukova, N. Use of Dynamic Models in Cognitive Cyber-Physical Systems. *Eng. Proc.* **2023**, *33*, 14. <https://doi.org/10.3390/engproc2023033014>

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 9 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the distinctive features of the modern stage of development of society is very rapid changes in people’s lives and production methods. Modern companies and organizations adapt their strategy, business models, products and services, as well as business processes and information systems, to increase the level of their digitalization with the help of intelligent services and products with digital improvement. The potential of the Internet and related digital technologies, such as the Internet of Things, artificial intelligence, data analyses, cloud computing, mobile systems and cyber-physical systems (CPS), involves strategic factors contributing to the development of digital platforms with rapidly developing ecosystems of intelligent services for digital products [1,2]. Currently, the CPS paradigm is widely used for building real-life systems. One can define CPS as systems based on the integration of entities of a different physical nature when separate subsystems of a different physical nature function as a single whole [3].

Nowadays, another point of interest for IT developers is cognitive systems. In the IT industry, the concept of “cognitiveness” came from cognitive psychology, which studies the processes occurring in the human brain during information processing. This concept later formed the basis of one of the branches of artificial intelligence, which is engaged in the development of artificial cognitive systems taking into account advances in neurophysiology and cognitive psychology. As the simplest cognitive system, we can consider a system that can have certain knowledge about itself and the external world, the functioning of which is realized in the form of achieving goals. We can say that a cognitive system should have at least a model of itself and models of the external world (context) presented in the form of

knowledge. The following definitions can be proposed: cognitive systems can be attributed to systems that are able to perceive information about the state of the external world and their own state, working with a model of themselves and a model of the external world; systems that do not use these models can be defined as intelligent systems.

One of the main distinctive features of modern CPS is the variability of their structure and behavior [4]. Thus, we can say that modern CPS may have the ability of evolution, i.e., they can be considered as developing systems that may have such typical features as (i) a spontaneous change of the system state; (ii) counteraction (reaction) to the influence of the environment (other systems), leading to a change in the initial state of the environment; and (iii) permanent changes in the structure and behavior of the system. If a developing system evolves at the expense of its own resources, then such systems are called self-developing (self-sufficient developing).

Practically all self-developing cognitive CPS are Software Intensive Systems (SWIS) [5], which use mechanisms of working with knowledge.

2. Background

The CPS conceptual model consists of five levels: physical, network, data storage, processing and analytics and an application layer. Large-scale CPS are most often implemented on fog and edge computing platforms [6].

The mainstream of the CPS evolution is associated with the permanent complication of the implemented functionality, and nowadays many modern CPS can be considered as ambient intelligence systems. A significant part of modern CPS can be attributed to cognitive systems.

In relation to the IT domain, the term “cognitiveness” is most often associated with the derived concepts of cognitive modeling, cognitive systems and cognitive architectures and systems that implement cognitive behavior.

Cognitive modeling can be defined as a decision-making support process taking into account the mutual influence of various events and the relationship between them. Cognitive analyses, using the apparatus of cognitive models and technologies, promptly solve such tasks as building a model of the situation, assessing the impact of external and internal factors on possible scenarios of the situation, and identify trends in the development of situations. The cognitive modeling technology is usually used to solve management problems under unforeseen circumstances.

Classical systems with cognitive architectures use human-inspired heuristic algorithms, which are found by psychologists and biologists who are often part of the development team. The history of the creation of this kind of cognitive system comprises at least 40 years. For the most part, these are experimental systems. When constructing them, concepts such as patterns of situations and fusion are most often used [2]. For the most part, these are also decision-making support systems. As an example of using this approach, we can consider the concept of the Cognitive Internet of Things [7].

The main idea of this paradigm is that the modern Internet of Things is just a collection of a wide variety of devices equipped with status sensors and actuators. To manage and maintain such a complex infrastructure in a proper condition, it is necessary to endow it with intelligence, i.e., the ability to perceive information about its own state and, if necessary, issue control actions for the purpose of reconfiguring the structure. The cognitive system here is the Internet of Things management system, which can be built as a distributed intelligence system. Within the framework of this paradigm, no specific solutions are given; it is determined only what should be done, but not how the mechanisms of cognition should be implemented.

3. Problem Statement

A significant part of modern self-developing cognitive CPS are big and complex distributed systems with a high level of not only structural and functional but also architectural variability. One of the key problems associated with the construction of self-developing

cognitive CPS is the problem of ensuring their manageability at all levels. To solve this problem, in turn, it requires having sufficient information about the current state of the system, which is usually presented as a set of models.

Designing self-developing cognitive CPS is also quite a difficult problem. Very often, the use of traditional SWIS design approaches [8] does not give the desired results. The development of new approaches to the design of self-developing cognitive CPS is required.

4. The proposed Approach

The main idea of the suggested approach is that it is proposed to use several modern paradigms for the construction of CPS, such as continuous architecture [9], agile architecture [10], digital twins [11] and digital threads [12]. The proposed approach can be considered as an adaptation of the digital threads' paradigm in relation to cognitive CPS.

The implementation is supposed to be based on a Model-Based Approach [13] and assumes the use of a three-level CPS life cycle model (LC).

A three-level model of self-developing cognitive CPS. In order to provide the required level of CPS capabilities such as flexibility, this opportunity should be laid at earlier stages of the CPS LC, i.e., at the development stage. This model can be used as the basis of a development viewpoint or encapsulated into a separate architectural viewpoint [8].

The proposed model describes the LC of continuous agile architecture at all five LC stages on three levels.

The structure of the model is shown in Figure 1.

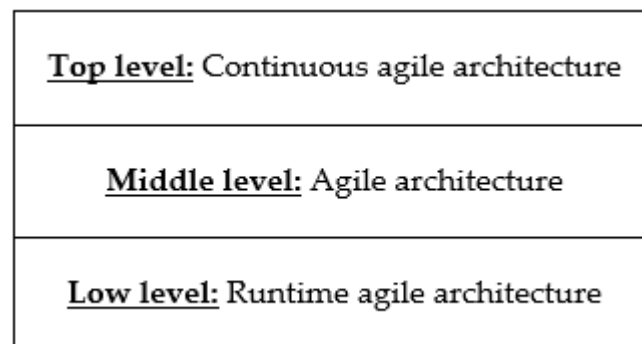


Figure 1. Three-level model.

At the top level, CPS are described in terms of a continuous architecture [9].

At the middle level, the system is represented as an agile architecture and is described as a multilevel relatively finite state automaton [14].

At the low level, structural and functional models describing the system in the process of operation are used.

These models describe CPS at different LC stages. The top level describes all LC stages. The middle level refers mainly to the stages of operation and modernization. The low level relates mainly to the stage of operation.

Top level. At this level, the process of transformation of the architectural representations of the CPS during the transition between the individual stages of the LC cycle is described (Figure 2).

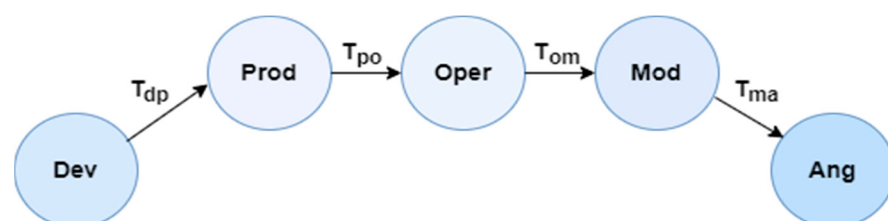


Figure 2. Transformation of artifacts.

In general, five phases of the life cycle can be distinguished: Development (Dev), Production (Prod), Operation (Oper), Modernization (Mod) and Annihilation (Ang). The chain of transformations can be represented as

Art Dev--> Tdp--> Art Prod--> Tpo--> Art Oper--> Tmo--> Art Mod--> Tma-->--> Tmo--> Art Ang.

At each phase of the LC, a certain set of artifacts (Art) is formed. Artifacts can belong to the following classes:

Art classes: = <Model | Code | Physical Entity>. Each of the phases uses its own set of artifacts.

One can define four basic types of transformations:

Mi--> Mi-1, M--> Code, M--> PhE, Mi1--> Mi2, where M are models and PhE are physical entities.

In general, at each stage of the LC, its own set of entities is used, which are interconnected by transformation mechanisms.

At the top level, one can discuss a single model. At the middle level, each phase uses its own artifacts. Each phase uses its own agile architectures (AA). Each of the AA architectures is matched with a set of runtime architectures (RTAA).

Each phase uses its own artifact systems, but some common features can also be distinguished.

Artifacts, the classification of which is shown in Figure 3, can be divided into three groups: entities, models and metrics.

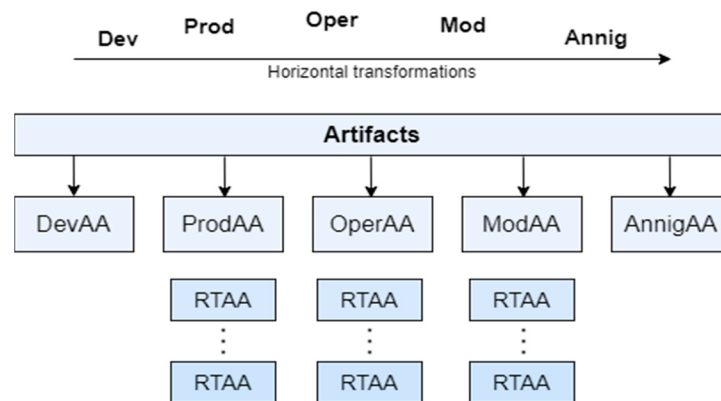


Figure 3. Artifact system structures.

Entities can be physical (PE) or virtual (VE). Physical entities can be either physical elements of the CPS, or their physical models. Both CPS elements and CPS models can act as virtual ones. A virtual entity can concede as a code or as a business process.

Currently, models are becoming one of the key elements of CPS. Models can be both physical and virtual (PM and VM). In this paper, physical models are not considered. A complete classification of the models can be found, e.g., in [13]. From the point of view of the subject of research, models can be divided into models describing a class of systems (VCM) and models describing instances of a class (VIM).

The model and the entity being modeled can be in the following relations: (i) model implementation $M \rightarrow E$; (ii) model building $E \rightarrow M$; and (iii) synchronization of the model and the entity state $M \leftrightarrow E$.

Middle level. At this level, the mechanisms of structural, functional and architectural reconfiguration are supported. This level describes the operational phase.

At the middle level, the system can be described as a multilevel relatively finite state automaton, the vertices of which correspond to architectural states (As), and the transitions are illustrated by the fitness functions (FF) [10].

This model can be used both on exploitation and the modernization phase.

In the first case, we are discussing maintaining agility mechanisms in the runtime mode, and in the second case, about maintaining these mechanisms in the process of CPS modernization.

Low level. At the low level, structural and functional mining mechanisms are supported, i.e., tracking the current structural and functional (architectural) state of the observed CPS.

The architectural state of the observed CPS can be described using different models depending on the goals.

Approaches to implementation. The implementation of self-organizing cognitive CPS on the fog and edge computing platforms is associated with solving a number of problems, the central of which are the following: (i) it is obvious that cognitive CPS should be focused on working with knowledge; however, working with ontologies and knowledge graphs requires sufficiently powerful computing resources that are not always available even at a fog level, and really full-scale support of mechanisms for working with knowledge can only be organized at the cloud level; (ii) neural networks can be considered as an effective tool for implementing cognitive functions, but their training can really be organized only at the cloud level.

The generalized policy related to the implementation of the cognitive functions of the CPS can be formulated as follows. Functions related to managing slow business processes, such as self-organization management, can be implemented in terms of ontologies and knowledge graphs mainly in a cloud environment; neural networks and federated learning algorithms can be used to implement functions with higher performance requirements.

The maturity models. The proposed model approach is not always possible and appropriate to be applied at full scale. In some cases, it is more appropriate to apply it partially or only at the design stage. In this context, by analogy with the well-known methods of assessing the maturity of CMMI [15], one can discuss the maturity levels of the application of the suggested model approach. Five maturity levels can be distinguished: (i) level 0—the model approach is not applied; (ii) level 1—the model approach is applied only at separate stages, e.g., at the development stage; (iii) level 2—the model approach is used to a limited extent, both at the development and runtime, e.g., it can be used only to control structural dynamics or only to control dynamic business processes; (iv) level 3—the model approach is used to solve both structural dynamic management problems and dynamic business process monitoring, but the possibilities of building models are limited, e.g., one can only tune the properties of individual model elements, or one can add only elements of certain types to the model; and (v) level 4—all models can be synthesized in the runtime, no restrictions are imposed on the complexity and structure of models and model transformation mechanisms are used.

5. Case Study

The proposed approach was used by the authors in the construction of a number of real systems. It should be noted that in order to solve real problems, it was not required to use it at full scale.

Below is a brief description of real projects in which the proposed approach was used.

Case 1. A cable TV network maintenance system. This project solves the problem of maintenance of a complex distributed system. The main task was to create cognitive CPS based on the existing cable TV network, with the abilities of self-repair and self-optimization, i.e., implementing elements of cognitive behavior. When solving this problem, it turned out to be possible to limit ourselves only to structural dynamic models that are focused on supporting the operation phase [16].

Case 2. Production system. The purpose of the development was to modernize the production system with mobile entities at the sensory level, built on the platform of the Industrial Internet of Things. There are two key subtasks in this project: collecting data at the sensory level in conditions of a very high level of electromagnetic interference and the task of building a corporate knowledge graph. This system is focused mainly on

work in the runtime and uses structural and functional models operating at a low level. The use of top-level models allows for increasing flexibility and simplifying the process of integration into the corporate information system. The presence of a dynamic model illustrating the production system in the form of a knowledge graph opens up possibilities for the construction of cognitive production systems. This example can be considered as the use of the proposed approach for the construction of dynamic digital twins [17,18].

Case 3. In the third case, the model approach was used to solve the problem of building an educational trajectory master level curriculum in the IT domain. This is the task of synthesizing business processes based on the model. This task is also interesting because educational systems can be considered as one of the varieties of socio-cybernetic systems where it is required to work with competence models. This indicates the possibility, with certain improvements related to the adaptation of the model, of using this approach to solve data processing problems in socio-cybernetic systems, for example, social networks.

Case 4. Digital twin of an electric piano. This example illustrates the possibility of constructing a dynamic digital twin. The idea is as follows. The piano often breaks down and there is a special connector for connecting to a computer for diagnostics. Modern electric pianos have great capabilities, can change the timbre of the voice through a connected microphone, have good sound-reproducing speakers that are not inferior to professional equipment for artists of the musical genre, have additional accompaniment functions and have a training mode for beginners to learn to play the piano. Many electric pianos have a special diagnostic input, for which the authors of the article developed a digital twin (Figure 4).

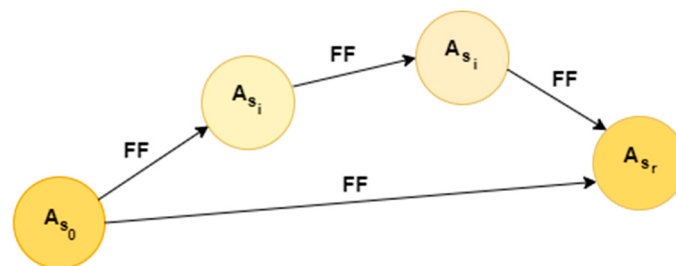


Figure 4. Middle level model.

The program has a menu through which it is possible to download and save the model (File) and screenshots for subsequent sending to the statistics server and various settings (Figure 5). On the left of the form, there is a static image of the object (electric piano), which can be hidden by pressing the “Hide” button. On the right is a schematic image of the object, which can be moved horizontally and vertically through the blue circle. It is possible to zoom in front of the magnifying glass image (plus and minus buttons), manually set the size by pressing the “Set” button and reset to the original dimensions, which are presented by default, by pressing the “Reset” button. On the left, you can select the event levels, Easily, Danger and Seriously, and see the current diagnostic status “Good”; meaning everything is fine and the components are in good working order. Below is the configuration of sensors and signal sources, Keys, Pedals, Speakers, Functions and Connectors, with the possibility of additional configuration using the “More...” button. The log with additional functions is displayed on the right: clear, save to a file, open a file with a log and open it on the site in a table with a search using the “Web” button. Only the last 20 lines of the log are displayed on the form element to avoid memory overflow errors.

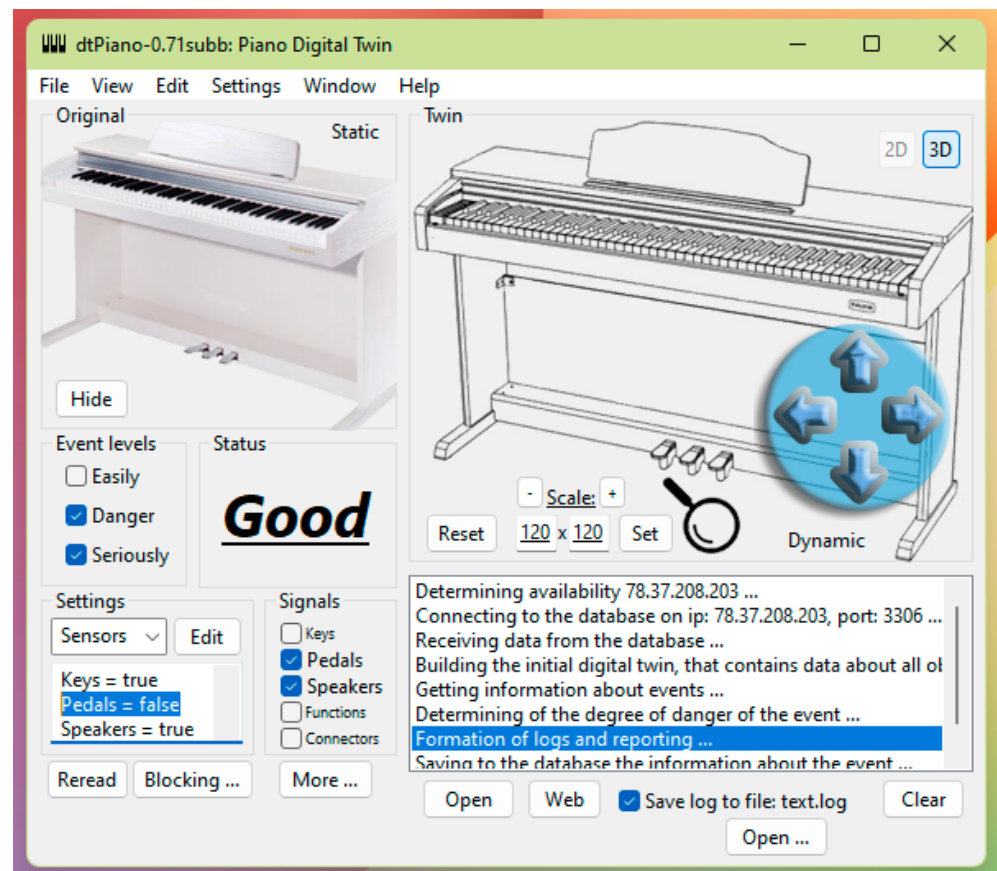


Figure 5. Digital Twin for Troubleshooting.

6. Conclusions

The above-described approach can be considered, first of all, as an approach to the development of large-scale cognitive heterogeneous CPS with a high level of structural, functional and architectural dynamics, built on fog computing platforms. The key idea of the developed approach is to use a system of dynamic models of the CPS, which is kept up-to-date throughout the entire system LC.

Implementation of the suggested approach allows for solving a number of important problems, such as increasing the level of CPS intelligence and increasing the level of service availability, to reach a new level of complexity of the created CPS. The presence of model knowledge about the past states of the system allows for determining the root causes of events and predicting future states of the system to implement proactive management. In addition, the availability of knowledge about past states allows for using learning mechanisms.

A necessary and sufficient condition for using the proposed model approach is a sufficiently high complexity of the structure and behavior of CPS that have high structural dynamics and adaptive behavior.

Currently, work on the further development of the suggested approach is being carried out in three directions: (i) creating new models; (ii) expanding the scope of the approach, in particular, for its application for the analysis of social networks; and (iii) development of new algorithms of model synthesis and model transformations.

Author Contributions: Conceptualization, A.V. and N.Z.; methodology, A.V.; software, A.S.; validation, M.C., A.V. and N.Z.; formal analysis, A.V.; investigation, N.Z.; resources, A.S.; data curation, A.S.; writing—original draft preparation, A.S.; writing—review and editing, A.V.; visualization, A.S.; supervision, N.Z.; project administration, N.Z.; funding acquisition, N.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sanfelice, R.G. Analysis and Design of Cyber-Physical Systems. In *A Hybrid Control Systems Approach//Cyber-Physical Systems: From Theory to Practice*; Rawat, D., Rodrigues, J., Stojmenovic, I., Eds.; CRC Press: Boca Raton, FL, USA, 2016; ISBN 978-1-4822-6333-6.
2. Kotseruba, I.; Tsotsos, J.K. 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artif. Intell. Rev.* **2018**, *53*, 17–94. [\[CrossRef\]](#)
3. Calinescu, R.C.; Camara Moreno, J.; Paterson, C. Socio-Cyber-Physical Systems: Models, Opportunities, Open Challenges. In *Proceedings of the 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems*, Montreal, QC, Canada, 28 May 2019.
4. Capilla, R.; Bosch, J.; Kyo-Chul, K. (Eds.) *Systems and Software Variability Management*; Springer: Berlin/Heidelberg, Germany, 2013.
5. Anthony, J.L. *Architecting Software Intensive Systems. Practitioner's Guide*; Taylor & Francis Group, LLC.: Abingdon, UK, 2009; 453p.
6. Buyya, R.; Srirama, S.N. (Eds.) *Fog and Edge Computing Principles and Paradigms*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2019; 490p.
7. Wu, Q.; Ding, G.; Xu, Y.; Feng, S.; Du, Z.; Wang, J.; Long, K. Cognitive Internet of Things. A New Paradigm beyond Connection. *IEEE Internet Things J.* **2014**, *1*, 129–143. [\[CrossRef\]](#)
8. Bass, L. *Software Architecture in Practice*, 3rd ed.; Bass, L., Clements, P., Kazman, R., Eds.; Addison-Wesley: Upper Saddle River, NJ, USA, 2013; 661p.
9. Ford, N.; Parsons, R.; Kua, P. *Building Evolutionary Architectures*; O'Reilly Media: Sebastopol, CA, USA, 2017; 272p.
10. Bloomberg, J. *The Agile Architecture Revolution: How Cloud Computing, REST-Based SOA, and Mobile Computing Are Changing Enterprise IT*; Wiley & Sons, Inc.: Hoboken, NJ, USA, 2013; 278p.
11. van der Valk, H.; Hunter, J.; Rabe, M.; Otto, B. Digital twins in simulative applications: A taxonomy. In *Proceedings of the 2020 Winter Simulation Conference*, Orlando, FL, USA, 14–18 December 2020. [\[CrossRef\]](#)
12. Digital Thread. Available online: <https://searcherp.techtarget.com/definition/digital-thread> (accessed on 1 September 2022).
13. Weillkiens, T.; Lamm, J.; Roth, S.; Walker, M. *Model-Based System Architecture*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2016; 375p.
14. Osipov, V.; Stankova, E.; Vodyaho, A.; Lushnov, M.; Shichkina, Y.; Zhukova, N. Automatic Synthesis of Multilevel Automata Models of Biological Objects. In *Proceedings of the Computational Science and Its Applications—ICCSA 2019: 19th International Conference*, Saint Petersburg, Russia, 1–4 July 2019; Springer: Cham, Switzerland, 2019; Volume 11620.
15. CMMI. Available online: <http://www.sei.cmu.edu/> (accessed on 1 September 2022).
16. Osipov, V.; Zhukova, N.; Vodyaho, A. About one approach to multilevel behavioral program synthesis for television devices. *Int. J. Comput. Commun.* **2017**, *11*, 17–34.
17. Vodyaho, A.; Abbas, S.; Zhukova, N.; Chervoncev, M. Model Based Approach to Cyber–Physical Systems Status Monitoring. *Computers* **2020**, *9*, 47. [\[CrossRef\]](#)
18. Vodyaho, A.; Osipov, V.; Zhukova, N.; Chernokulsky, V. Data Collection Technology for Ambient Intelligence Systems in Internet of Things. *Electronics* **2020**, *9*, 1846. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.