Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 217 (2023) 969–978

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Industry 4.0 and Smart Manufacturing

Conceptual Thoughts on Biointelligent Embedded Systems and Operating Systems Architecture

Arber Shoshi[ab]*, Robert Miehe[bc], Thomas Bauernhansl[bc]

[a]*Graduate School of Excellence advanced Manufacturing Engineering (GSaME), Universität Stuttgart, Nobelstr. 12, 70569 Stuttgart, Germany*
[b]*Fraunhofer-Institute for Manufacturing-Engineering and Automation IPA, Nobelstr. 12, 70569 Stuttgart, Germany*
[c]*Institute of Industrial Manufacturing and Management (IFF), Faculty of Engineering Design, Production and Automotive Engineering, University of Stuttgart, Allmandring 35, 70569 Stuttgart, Germany*

Abstract

Biointelligent systems are among the most important enablers for a sustainable transformation of industry. The convergence of bioware, hardware and software generates completely new system architectures that allow products to be manufactured and utilized more decentralized, autonomous, and demand-driven. With the increasing amount of (heterogeneous) production data required for this and the need for real-time capability, the importance of edge computing is growing. To fulfill the demand for edge computing, the requirement for intelligent high-end embedded systems, together with an operating system, for future production has grown. After a series of recent publications describing the basic principles of the necessary information technology networking of biological and technical systems, we introduce the concept of a biointelligent, embedded system, discuss its need for an operating system and place it in a larger system context. By comparing the use of embedded operating systems in digital product systems, necessities of an operating system in biointelligent systems are derived, e.g. managing resources, guaranteeing real-time processing, mastering complexity, ensuring usability and increasing security. Upon the evaluation of the fulfillment of existing embedded operating systems to these necessities, we derive white spots for future research.

Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing

Keywords: biointelligent; embedded systems; operating system; CPS

* Arber Shoshi.
E-mail address: arber.shoshi@gsame.uni-stuttgart.de

1877-0509
Peer-review under responsibility of the scientific committee of the 4th International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2022.12.294

1. Introduction

The past decades have seen the introduction of increasing amounts of technology into our everyday lives [1]. As a result, great distances became bridgeable, hard labor was largely eliminated, and communication across continents became a matter of course. At the same time, this was accompanied by an enormous increase in CO2 emissions and the destruction of nature, resulting in global warming [2]. Beyond all doubt, it is necessary to rethink products and production in order to ensure their sustainability.

In this context, biointelligence was introduced recently as an innovation path towards a sustainable technology future [3]. Its goal is to bring together life, engineering, and computer science to create an innovation space for companies and promote the transition to a sustainable economy. The convergence of these previously separate disciplines enables new system architectures that allow products to be manufactured and utilized in a more decentralized, autonomous, and demand-driven manner, e.g. via small-scale non-expert bioproduction units, so-called smart biomanufacturing devices (SBMDs) [4]. By utilizing biological resources harvested on-site, these systems replace large supply chains and increase resource efficiency. Initial approaches to these technologies include the production of advanced therapy medicinal products in hospitals [5], the production of personalized foods in restaurants [6], and the effective utilization of biogenic waste materials in decentralized biorefinery [7] or bioprinting concepts [8, 9].

Biointelligent production by SBMDs requires efficient control and governance of these systems, culminating in autonomous, intelligent behavior [10]. This inevitably results in growing complexity and an increasing amount of (heterogeneous) data generated during production. To counteract these effects, increasingly intelligent embedded systems (ES) are being used, supporting a trend from centralized to decentralized process computing, i. e., edge computing [11–13]. This is intended to process certain information directly at the ES and perform the inference - i.e., the conclusion - directly at the edge of the system and no longer in an overall centralized manner. This type of process control requires powerful ES that, unlike their predecessors, are more high-performing, more complex, and possess an embedded operating system (EOS).

Common universal operating systems (OS) are intended to coordinate the complexity and resources of a system (especially a computer system). EOS, on the other hand, are gaining additional importance in the context of the digitalization of industrial processes since they can guarantee a real-time capability of the system [12, 13]. Future biointelligent system architectures will additionally strengthen the trend of more decentralized information processing [10]. After a series of recent publications describing the basic principles of the necessary information technology networking of biological and technical systems, a number of questions remain unanswered:

- What role can embedded systems play in biointelligent systems?
- Are biointelligent embedded systems conceivable? If so, what would be their characteristics?
- Is there a need for a novel OS architecture?
- What are white spots for research and development?

2. Methodology

By identifying essential questions for the use of digital embedded systems in current production, a keyword search string was developed for a literature search in the context of biointelligent systems. The results were checked by means of existing studies. From the criteria derived from literature, existing solutions for purely digital systems were examined and their possible degree of fulfillment was determined with regard to an application in biointelligent systems (BIS). Thereupon, gaps for further research activities were derived. Fig. 1 shows the procedure of this paper.

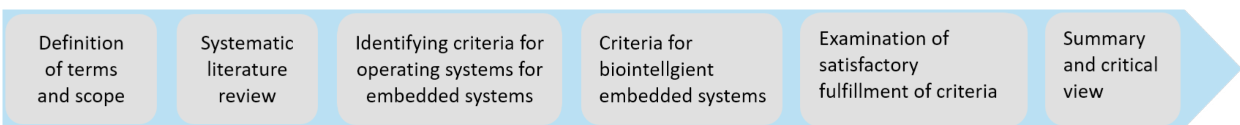

Fig. 1: Research process of this paper

3. Fundamentals of Operating Systems

An exact definition of the term OS is barely possible due to the many proprietary programs and their functions. Generally, an OS is understood as the interface between the user's hardware and the application software. An OS is a number of computer programs that manage a computer's system resources (e.g., RAM, hard disk space, input and output devices, etc.) and make them available to the application programs [14]. It can be found in almost all digital or electronic devices operated by software, why its tasks differ depending on the device's field of application [15]. Nonetheless, specific fundamental tasks are common to almost all OSs. Fig. 2 illustrates the generic structure of the relationship between the user, the application, the hardware, and the OS as the interface in between.

The fundamental tasks of an OS include process, file, and memory management. It regulates, among other things, which process is carried out, which storage capacity is allocated to this process, and where the data is stored and managed. A process is a currently executing program and provides information such as state and control information, register contents, stacks, etc. During the coordination of processes, memory management is essential. This ensures that a program does not interfere with memory that is already being used by another program. Hence, a crash in one

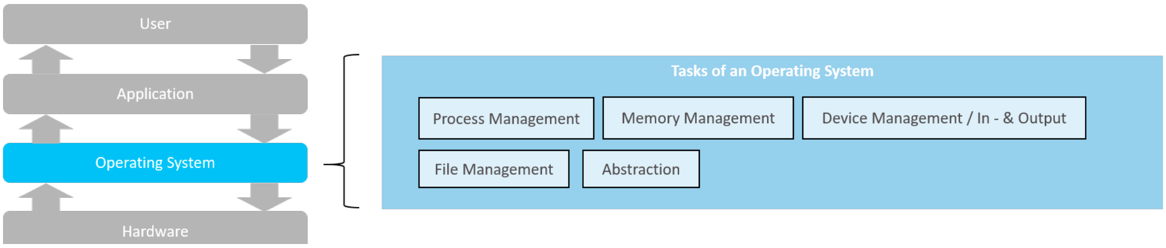

Fig. 2 Relationship between the individual stakeholders and the tasks of an OS, according to Brause (2017).

program cannot shut down other programs or the entire system. The device management and abstraction allow devices from different manufacturers to be made usable through device drivers [15]. Another task of the OS is to hide the complexity of the hardware, e.g., by providing a graphical user interface.

An OS has certain elements that are necessary for the operation of a computer, e.g., utilities, translation and organizational programs, and user interfaces, while leaving the developer a certain amount of freedom in positioning the corresponding functions [15]. These can be provided either in the OS's kernel or shell. The kernel is directly connected to the system's hardware and includes the most important functions, e.g., parts of the processor, memory and device management. Programs that run in the kernel, or Ring 0, are prioritized and have unrestricted access to the processor's memory. Operating Systems provide different level of access to resources, by applying a hierarchical protection domain, or protection rings Programs within a particular ring can only interact with resources and other programs in bordering rings. This layered model creates security in that a random program cannot access the kernel unhindered thus crashing the system [15]. All other programs, i.e., programs that access via the user shell, do not have this unrestricted access. Earlier computer systems, such as old ESs, used single tasking to compute a redundant task quickly and efficiently. New high-end ESs and all other computer systems use multitasking to run multiple processes quasi-simultaneously, thereby increasing efficiency [15–17]. Therefore, an OS is a program that makes a technical system run. Most people think of the PC as the computer system with the most common OS Windows, MacOS or Linux, whereas Linux is strictly speaking a kernel and not an operating system. However, the most common computer systems are small, embedded components integrated into larger systems, which is why they are called embedded systems. ES are computer systems designed to perform one or more designated functions within a larger system. They cannot take over other functions and are intended to perform their functions within a time period so that the larger main system functions properly [18]. ES are either powered by proprietary or open source EOS or even have no EOS at all, if the task of the ES is simple [19]. The most widespread are Windows CE, QNX, VxWorks, Embedded Linux and RIOT, the last two being open source systems [15].

Such EOSs were developed to efficiently control computer systems that have simple routine tasks and therefore only limited processor capability. For this purpose, an EOS is much leaner than a universal OS and must control the given resources as efficiently as possible in order to be able to meet e.g. certain time specifications and achieving real-time. This real-time ability also has a major significance in the use of automation robots, which are becoming increasingly indispensable in production and logistics [20]. The necessity of OSs in modern production is inevitable. In a production with countless sensors and actuators, optimal functionality over their entire lifetime cannot be guaranteed with an embedded software. This and the management of complexity require the use of a specific manufacturing OS (MOS) [21].

MOSs usually handle various control systems that are specifically developed for production and are highly proprietary, but they usually contain the same tasks and goals. These OSs describe control, monitoring and supervision over an entire production process, including its material and information flows. Such a MOS is, for example, 'FabOS' developed by the Fraunhofer Institute for Manufacturing Engineering and Automation IPA [22] or the openMOS is developed within the Horizon 2020 EU's research and innovation funding program [23]. Another approach of a production site's holistic control are digital twins, which are virtual models of a real system that enable bidirectional information exchange between the real and virtual systems, including its various data, information and energy flows [24–26]. To realize such a perfect cyber image, digital twins are based on cyber physical systems (CPS). They are smart, interconnected ESs that help control a complete production system through actuator, sensors, data processing, and a real-time exchange of all information needed for a production based on Internet-of-Things (IoT) [27, 28].

Depending on the operation site, e.g. in the field of life science, the demands on ESs can be smaller or larger. The more complex a product and its tasks are, the more complex is the underlying software [29]. For example, a pacemaker is expected to always react in the same pattern within a certain time span, rather than reacting to incoming sensor data. However, if one considers systems that also have non-deterministic behavior, i.e. react to changes in the environment, a possibility of an information exchange between technical and biological components must be created by using adaptive biology-technology interfaces (BTI). Thus, interactions of technical components with cellular systems and vice versa are realized by means of physical principles (e.g. electrical, chemical, mechanical, and optical) [10].

Implementing a BTI on a small intelligent subsystem in order to monitor and control a certain area of the main system inevitably results in the question whether a biointelligent embedded system (BIES) is conceivable. For this purpose, a more precise differentiation between an ES, CPS, BIES, and BIS must be made at this point. Fig. 3 shows on the left the categorization of ES in the context of Internet of Things (IoT). ES represent a subset of CPS, e.g. by

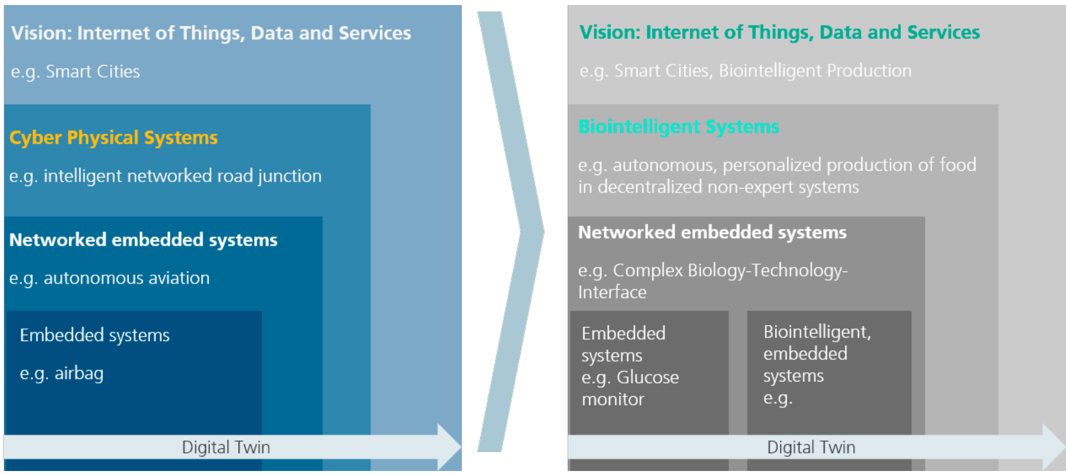

Fig. 3: Classification of embedded systems and biointelligent embedded systems [1].

using sensors for autonomous driving or flying, which only have a specific function and would be useless outside the main system, i.e. in a different environment. Likewise, a CPS will be seen as a subsystem of a vision, e.g., IoT [30]. In contrast, a BIES would represent a subsystem of a BIS. Two types of BIES are conceivable. First, if an ES is integrated as a classical ES in a BIS, it becomes a BIES only by virtue of its environment. An example is a spectroscopic sensor for in-line bioprocess monitoring [31]. However, an ES may especially be considered biointelligent if it contains within itself the principles of biointelligence by having biological components and intelligent system behavior. An example of this would be a biobased soft sensor that can provide online measurement in a substrate and learn at the same time. As such, this sensor would be a BIES, would have a specific function and could only meaningfully perform its task within the main system. To further investigate and understand the need for a BIES, it can be considered in the context of the biointelligent production process.

4. Role of a biointelligent embedded system

The biological transformation is preceded by the digital transformation in industry [32, 33]. Both transformations are currently ongoing, with the digital transformation being more advanced and enabling the biological transformation in the first place. It has to be stated that a BIS is not achieved simply by adding a biological component but involves much greater complexity [3, 10].

Fig. 4 illustrates parallels and discrepancies between a traditional (digital) and a biointelligent production and provides the basis for comparing the two systems. The highest levels of consideration in this diagram are digital or biointelligent production. Digital production can be divided into different digital, networked, and agile production sites or factories. These are places where several steps of the value creation process are accumulated. One level deeper, such

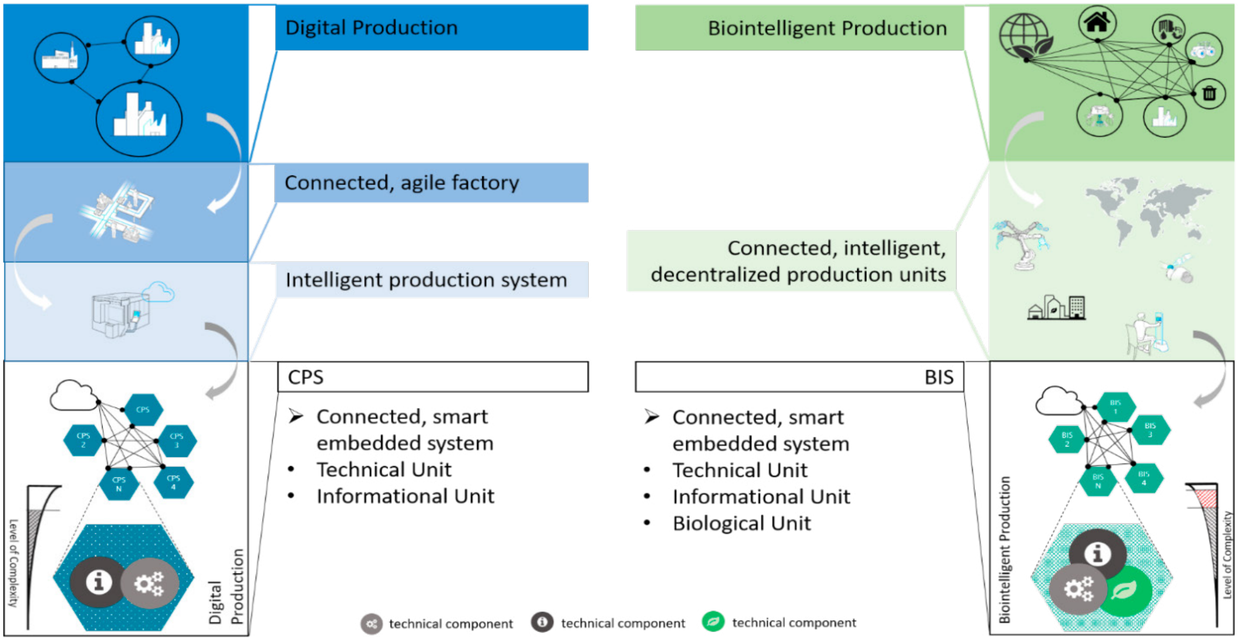

Fig. 4: Different levels of digital and biointelligent production

a factory can be characterized as a bundle of intelligent production systems or production units. However, biointelligent production is increasingly promoting the type of on-site production, as it can be a sustainable production method in many aspects [34, 35]. The production units have to be as decentralized as possible, with the goal of proximity to the user or the location of use. This is expressed in the idea of manufacturing by the consumer, where the finished product is not delivered, but the raw material is provided to the customer for in-house production [36].

Such a distribution of production units entails a higher complexity and thus a higher coordination effort. The fact that a BIS requires an intelligent response not only to technical processes but also to biological processes further drives complexity, which is illustrated in the bottom level of the figure. In digital production, CPSs that integrate technical and informational components play a fundamental role in realization. The counterpart for biointelligent production here are BIS. Both systems can thus be seen as basic pillars for realizing the respective productions: CPS for the digital production system and BIS for the biointelligent production system. Their subsystems, ES and BIES, are assumed in this paper to have equivalent functions and thereby support the systems CPS and BIS, respectively, to the same extent. Now comparing ES and BIES could help to analyze a necessity for an OS for BIES.

In a literature review, criteria can be found that determine whether a CPS should be integrated into a digital production system with or without an OS from a software technical perspective. Subsequently, these criteria must be examined to determine whether they are also applicable to a BIS. If they are mostly required by a BIS, it can be concluded that a BIES requires an OS. Following this, existing EOS will be examined to see to what extent they fulfill these criteria already in order to find gaps for future research and development activities. The decision of whether an EOS is necessary and which one is suitable for the system is made on the basis of various criteria. They explore the tasks and specifications of the ES or the main system by using guidelines that ES developers also use in decision-making. Table 1 summarizes 12 guiding questions as a result of the literature review. These questions are used to investigate whether an EOS would make sense from a software engineering point of view. These questions help to determine how complex the functions of the subsystem (ES) are and thus, according to the buttom-up method, allow us to conclude how complex the suprasystem (CPS) is. The more the criteria have to be fulfilled, the more complex the system is and the higher the need for an EOS. If the same questions are now asked of a BIS, the tasks of the BIES can now be inferred according to the top-down principle. The more the criteria for a BIS have to be fulfilled, the more complex the tasks of the BIES will be. If the tasks of a BIES are very complex, an OS is required to ensure the correct functionality of the system.

The right part of the table shows the ratings of existing EOS that describe the extent to which this criterion is satisfactorily met. The rating is represented by the use of Harvey balls. The more the ball is filled with black color, the higher the importance of the criterion for BIS and the more the analyzed EOS meets the criterion. It is found that many factors favor the use of an EOS, with mastering complexity, system extensibility, custom I/O devices, and ensuring ease of use.

Table 1: Requirements and performance levels of EOSs in BISs [4, 10, 14, 37–46].

	BIS	Embedded Linux	QNX	Windows CE	VxWorks	RIOT
How important is real time?	◕	◕	◕	●	●	●
Is multitasking mandatory?	●	●	○	○	○	●
Should the system be scalable?	●	●	●	●	●	●
Is the system complex?	●	●	◕	□	●	□
Should the system be portable?	●	●	●	●	●	●
How important is unlimited memory size?	●	●	●	●	●	□
Is the CPU power limited?	□	◔	□	□	□	□
Is device power consumption an issue?	□	□	□	□	□	◕
Has the system obscure or custom peripherals?	●	●	●	●	●	□
Is a memory management unit (MMU) required?	●	●	●	●	◕	○
Is application security an issue?	□	●	●	●	●	●
Is usability required?	●	□	◔	□	●	●
Open Source	●	●	○	○	○	●

One of the first criteria for using an EOS is the question of real-time capability, multitasking ability, and portability. Since an EOS can efficiently distribute and manage a system's resources, this can enable the quasi-simultaneous processing of multiple programs, but also guarantee that a program's computation will be performed in a given time. This is important if the system does not allow delayed reactions since otherwise catastrophic consequences can be expected for the system and the environment. However, this is only of limited importance for a BIS since biological processes often do not occur within a few milliseconds, and a slight delay of the system is acceptable. As a rule, real time is divided into two main categories: hard and soft real time. The major difference between the two is the significance of the consequences of exceeding the time interval to be met. If this interval is exceeded and the consequences for the environment, people or the plant itself are minor, it is called soft real time. However, if serious consequences are to be expected, e.g. the endangerment of humans, it is called hard real time, since this must be met under all circumstances [10]. On the other hand, the guarantee that a reaction will be carried out at least within a certain time window, even when several programs are executed simultaneously, is advantageous for a BIS since it enables the precise control of the entire system, similar to the CPS [47]. Portability is needed because basic elements of ESs often contain valuable intellectual property, for instance, sophisticated and finely tuned control algorithms. Therefore, components should be reusable and transferable, i.e., portable across platforms [45]. Thus, it can be concluded that portability is also an important aspect of BIES that aim at achieving the highest possible level of decentralization.

Other criteria that an EOS advocates are complexity, scalability, and the presence of many different peripherals. In a complex system, the EOS helps with resource allocation and thus efficient operation of the system, as well as scalability and integration of new system components [48]. By creating a standardized interface and different drivers, components can be added to the overall system more easily. Thus, the system can be adapted to future changes faster and easier, should there be the need to add new peripheral I/O devices. Thus, in a BIS that is intended to be modular, flexible, and adapt to future environmental conditions quickly and efficiently [35], there is no question that an EOS is important for BIES.

If memory size and CPU power consumption are limited, implementing an EOS would be counterproductive since it would consume memory for execution and load the CPU with calculations. However, with increasingly powerful higher-end embedded CPUs in more complex ESs, the load is acceptable. Such systems also partly require a Memory Management Unit (MMU), which creates a separation between process memory and main memory and thus allows, for example, the swapping out of unneeded memory or the isolation of processes from one another. Memory protection tasks are also carried out by the MMU, locking individual memory areas for code or further writing, and thus programs cannot access the memory of other programs, or the function of the EOS is endangered in itself by other programs. The creation of security is also important for a BIS's smooth functioning. The function of the system should be ensured at all times if possible. However, the system can also be endangered by external factors, which is why the security question must be asked. If the system needs to be protected from outside influence, it is advisable to implement an EOS that blocks access to certain areas in the system kernel, which is also partly achieved by using an MMU. The necessity for a BIS, in this case, is coupled to the area of application, similar to CPS. If the consequences of crashing BIES are high, it is recommended to use an EOS.

Another aspect is the usability of the system. An EOS should increase or simplify the usability of a system for the end-user. The EOS should also enable a layperson to use and modify the system by enabling communication with the system via human interfaces rather than programming in machine language [15]. Miehe et al. describe a future-oriented biointelligent production system as a decentralized system that can also be used by non-experts. Thus, in the sense of personal manufacturing, in-situ production at the end customer should be enabled. In the area of simplified use of a system, the success story of the OS for the personal computer speaks for itself in making a computer usable for everybody. Such a feature could create an OS for a biointelligent manufacturing system by enabling biointelligent manufacturing for a broad audience.

In this case, the criterion "open source" applies more to the assessment of already existing EOSs than to answer the question of whether it is necessary or not. For a BIS, an open source OS should be sought. This is intended to achieve, among other things, wider dissemination, higher use in research and innovation activities, and higher security of the software. The advantages of this concept are evident in open source software such as Linux, with a wide range of uses and a strong community of developers [49].

5. Conclusion

The biological transformation of production is dedicated to the goal of making today's production sustainable and future-proof. According to the principles of a biological transformation, an increasing complexity of production is inevitable, which is why supporting concepts are required to counteract this. One concept can be the use of BIES that perform certain subtasks and functions autonomously and thus relieve the overall system. We already see this trend in digital production through edge computing [11, 13]. ES in BISs have not yet been investigated so far, which is why a more precise definition and the necessity is essential. BIES are considered as such under two conditions: either they are integrated in a main biointelligent system or they carry within themselves the principles of biointelligence. Another question investigated in this paper is whether the concept of a BIES requires an OS for proper use. For this purpose, the role of a BIES was compared to that of an intelligent ES in a digital production system, with the result that both would take a similar role, as an enabler of their system. Within a literature review, criteria were extracted that assist ES developers in determining whether or not an EOS is needed. These criteria were then applied to a biointelligent production system, with the result that a BIES requires an OS for proper deployment. Then, existing EOS were examined to see how they would fulfill these criteria. The result shows that these could partially cope with the tasks of a BIS well.

This work is intended to give a rough concept of a BIES, but it must be defined more precisely. For example, a smart wearable that interacts with a human could be a BIES according to this concept, as part of a larger intelligent biological system. They can be considered ES, but then the question arises whether their "embedding" in the human system makes them a BIES or should this characterization be limited to production environments only? This needs to be further explored and defined in the future. Likewise, the criteria developed here can only be considered as an initial guide for the implementation decision on an EOS for a BIES, as they only partially address aspects of biological systems. Furthermore, the developed criteria and their evaluation are to be regarded as subjective.

ESs are becoming increasingly complex, even without including a biological component, and the behavior of a biological component is different from a technical component. They are autopoietic, adaptive and variable. Therefore, in addition to resource coordination and a "real-time guarantee", an EOS must also be able to collect new and other information and pre-process it for the rest of the system. Thus, components on the control and field level of a manufacturing system should be able to easily implement and understand new elements of information intake (e.g., new sensors). Due to the necessity of biointelligent system to be adaptable to changing environments, they have to be able to be retrofit by nonprofessionals. However, some white spots for future research remain:

- How to cope with update ability and adaptability to recognize new data structures of far more complex and ever-changing biological systems?
- How to effectively integrate new building blocks (e.g. sensor) in order to adapt to changes and thus increase adaptability?
- How to enable security for possible applications in households or non-classic manufacturing environments (e.g., hospitals) with the highest possible level of networking with other BIES?
- How to enable universal usability similar to Windows, MacOS or Linux?

Acknowledgement

This work was supported by the Landesministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg (Ministry of Science, Research and the Arts of the State of Baden-Wurttemberg) within the Nachhaltigkeitsförderung (sustainability support) of the projects of the Exzellenzinitiative II.

References

[1] Kumar S, Tiwari P, Zymbler M. Internet of Things is a revolutionary approach for future technology enhancement: a review. J Big Data. 2019;6:1–21. doi:10.1186/s40537-019-0268-2.

[2] Al-Ghussain L. Global warming: review on driving forces and mitigation. Environ. Prog. Sustainable Energy. 2019;38:13–21. doi:10.1002/ep.13041.

[3] Bauernhansl T, Brecher C, Drossel W-G, Gumbsch P, Hompel M ten, Wolperdinger M, editors. Biointelligenz: Eine neue Perspektive für nachhaltige industrielle Wertschöpfung : Ergebnisse der Voruntersuchung zur biologischen Transformation der industriellen Wertschöpfung (BIOTRAIN). Stuttgart: Fraunhofer Verlag; 2019.

[4] Miehe R, Full J, Scholz P, Demmer A, Bauernhansl T, Sauer A, Schuh G. The Biological Transformation of Industrial Manufacturing-Future Fields of Action in Bioinspired and Bio-based Production Technologies and Organization. Procedia Manufacturing. 2019;39:737–44. doi:10.1016/j.promfg.2020.01.437.

[5] Salmikangas P, Schuessler-Lenz M, Ruiz S, Celis P, Reischl I, Menezes-Ferreira M, et al. Marketing Regulatory Oversight of Advanced Therapy Medicinal Products (ATMPs) in Europe: The EMA/CAT Perspective. In:

Regulatory Aspects of Gene Therapy and Cell Therapy Products: Springer, Cham; 2015. p. 103–130. doi:10.1007/978-3-319-18618-4_6.
[6] Ma Y, Zhang L. Formulated food inks for extrusion-based 3D printing of personalized foods: a mini review. Current Opinion in Food Science. 2022;44:100803. doi:10.1016/j.cofs.2021.12.012.
[7] Full J, Miehe R, Kiemel S, Bauernhansl T, Sauer A. The Biological Transformation of Energy Supply and Storage – Technologies and Scenarios for Biointelligent Value Creation. Procedia Manufacturing. 2019;39:1204–14. doi:10.1016/j.promfg.2020.01.349.
[8] Hoch E, Tovar GEM, Borchers K. Bioprinting of artificial blood vessels: current approaches towards a demanding goal. Eur J Cardiothorac Surg. 2014;46:767–78. doi:10.1093/ejcts/ezu242.
[9] Wenz A, Borchers K, Tovar GEM, Kluger PJ. Bone matrix production in hydroxyapatite-modified hydrogels suitable for bone bioprinting. Biofabrication. 2017;9:44103. doi:10.1088/1758-5090/aa91ec.
[10] Miehe R, Fischer E, Berndt D, Herzog A, Horbelt J, Full J, et al. Enabling bidirectional real time interaction between biological and technical systems: Structural basics of a control oriented modeling of biology-technology-interfaces. Procedia CIRP. 2019;81:63–8. doi:10.1016/j.procir.2019.03.012.
[11] IDS. Edge- & Cloud-Computing: Nachhaltige Edge-Intelligenz. Automation und Digitalisierung. 2022;24:18–27.
[12] Lee SH. Real-time edge computing on multi-processes and multi-threading architectures for deep learning applications. Microprocessors and Microsystems. 2022:104554. doi:10.1016/j.micpro.2022.104554.
[13] Hildebrandt A, Landhäußer W, editors. CSR und Digitalisierung: Der digitale Wandel als Chance und Herausforderung für Wirtschaft und Gesellschaft. 2nd ed. Berlin, Heidelberg: Springer; 2021.
[14] Tan SL, Tran Nguyen BA. Survey and performance evaluation of real-time operating systems (RTOS) for small microcontrollers. IEEE Micro. 2009:1. doi:10.1109/MM.2009.56.
[15] Brause R. Betriebssysteme. Berlin, Heidelberg: Springer; 2017.
[16] Mandl P. Grundkurs Betriebssysteme. Wiesbaden: Springer Fachmedien; 2014.
[17] Baun C. Operating Systems / Betriebssysteme. Wiesbaden: Springer Fachmedien; 2020.
[18] Altenburg J. Embedded systems engineering: Grundlagen - Technik - Anwendungen. München: Hanser; 2021.
[19] Media O. How to Choose an Embedded Operating System - Embedded Computing Design. 23.06.2022. https://embeddedcomputing.com/technology/software-and-os/how-to-choose-an-embedded-operating-system. Accessed 23 Jun 2022.
[20] Tilley J. Automation, robotics, and the factory of the future. 07.09.2017. https://www.mckinsey.com/business-functions/operations/our-insights/automation-robotics-and-the-factory-of-the-future. Accessed 3 May 2022.
[21] Venkat S, Clyburn M, Campbell B. Energy Harvesting Systems Need an Operating System Too. In: SenSys '20: The 18th ACM Conference on Embedded Networked Sensor Systems; 16 11 2020 19 11 2020; Virtual Event Japan. New York,NY,United States: Association for Computing Machinery; 2020. p. 15–21. doi:10.1145/3417308.3430274.
[22] Lukas M, Stock D, Csiszar A. FabOS: Towards an open, distributed, real-time-capable, and secure operating system for production. Procedia CIRP. 2021;104:962–7. doi:10.1016/j.procir.2021.11.162.
[23] openMOS. Home | openMOS. 08.06.2022. https://www.openmos.eu/. Accessed 8 Jun 2022.
[24] NASA Office of the Chief Technologist. NASA Modeling, Simulation, Information Technology & Processing - TA11. https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf. Accessed 10 May 2022.
[25] Heidel R, Hoffmeister M, Hankel M, Döbrich U, editors. Industrie 4.0: The reference architecture model RAMI 4.0 and the Industrie 4.0 component. 1st ed. Berlin: Beuth Verlag; VDE Verlag; 2019.
[26] Sauer M, Schaber V, Schel D, Schell O, Schier M, Schleipen M, et al. Details of the Asset Administration Shell: Part 1 - The exchange of information between partners in the value chain of Industrie 4.0. Specification. 2020. https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf. Accessed 12 Jun 2022.
[27] Wu X, Goepp V, Siadat A. Concept and engineering development of cyber physical production systems: a systematic literature review. Int J Adv Manuf Technol. 2020;111:243–61. doi:10.1007/s00170-020-06110-2.
[28] Leng J, Wang D, Shen W, Li X, Liu Q, Chen X. Digital twins-based smart manufacturing system design in Industry 4.0: A review. Journal of Manufacturing Systems. 2021;60:119–37. doi:10.1016/j.jmsy.2021.05.011.
[29] Hee YH, Ishak MK, Mohd Asaari MS, Abu Seman MT. Embedded operating system and industrial applications: a review. Bulletin EEI. 2021;10:1687–700. doi:10.11591/eei.v10i3.2526.

[30] Geisberger E, Broy M. AgendaCPS: Integrierte Forschungsagenda ; Cyber-Physical Systems. Berlin, Heidelberg: Springer; 2015.
[31] Claßen J, Aupert F, Reardon KF, Solle D, Scheper T. Spectroscopic sensors for in-line bioprocess monitoring in research and pharmaceutical industrial application. Anal Bioanal Chem. 2017;409:651–66. doi:10.1007/s00216-016-0068-x.
[32] VDMA, editor. Biologisierung der Industrie 2035.: Zukunftsbilder für den Maschinen- und Anlagenbau. Frankfurt am Main: VDMA; 2021.
[33] Petschow U, Ferdinand J-P, Dickel S, Flämig H, Steinfeldt M, Worobei, Anton. Dezentrale Produktion, 3D-Druck und Nachhaltigkeit: Trajektorien und Potenziale innovativer Wertschöpfungsmuster zwischen Maker-Bewegung und Industrie 4.0. Berlin: Institut für ökologische Wirtschaftsforschung; 2014.
[34] Miehe R, Buckreus L, Kiemel S, Sauer A, Bauernhansl T. A Conceptual Framework for Biointelligent Production—Calling for Systemic Life Cycle Thinking in Cellular Units. Clean Technol. 2021;3:844–57. doi:10.3390/cleantechnol3040049.
[35] Miehe R, Bauernhansl T, Beckett M, Brecher C, Demmer A, Drossel W-G, et al. The biological transformation of industrial manufacturing – Technologies, status and scenarios for a sustainable future of the German manufacturing industry. Journal of Manufacturing Systems. 2020;54:50–61. doi:10.1016/j.jmsy.2019.11.006.
[36] Do N. Integration of design and manufacturing data to support personal manufacturing based on 3D printing services. Int J Adv Manuf Technol. 2017;90:3761–73. doi:10.1007/s00170-016-9688-8.
[37] Kočíš T, Srovnal V. Operating Systems for Embedded Computers. IFAC Proceedings Volumes. 2003;36:359–64. doi:10.1016/S1474-6670(17)33774-6.
[38] Cheng X, Gong Y, Wang X. Study of Embedded Operating System Memory Management. In: 2009 First International Workshop on Education Technology and Computer Science; 2009. p. 962–965. doi:10.1109/ETCS.2009.753.
[39] Sally G, editor. Pro Linux embedded systems: Your complete guide to developing embedded Linux systems. Berkeley, Calif.: Apress; 2010.
[40] Baccelli E, Gundogan C, Hahm O, Kietzmann P, Lenders MS, Petersen H, et al. RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. IEEE Internet Things J. 2018;5:4428–40. doi:10.1109/JIOT.2018.2815038.
[41] Pothuganti K, Haile A, Pothuganti S. A comparative study of real time operating systems for embedded systems. International Journal of Innovative Research in Computer and Communication Engineering. 2016:12008–13. doi:10.15680/IJIRCCE.2016.
[42] Jabeen Q, Khan F, Hayat MN, Khan H, Jan SR, Ullah F. A Survey: Embedded Systems Supporting By Different Operating Systems. 11.05.2016. https://arxiv.org/pdf/1610.07899. Accessed 11 Jun 2022.
[43] Roussel K, Song Y-Q, Zendra O. RIOT OS Paves the Way for Implementation of High-performance MAC Protocols; 2015.
[44] Sumalan T, Lupu E, Arsinte R. Real Time Operating System Options in Connected Embedded Equipment for Distributed Data Acquisition. Carpathian Journal of Electronic and Computer Engineering. 2018;11:35–58. doi:10.2478/cjece-2018-0016.
[45] Sehestedt S, Giannopoulou G, Monot A, Wahler M. Virtualizing Embedded Firmware to Boost Innovation Cycles. In: 2019 IEEE International Conference on Software Architecture Companion (ICSA-C); 3/25/2019 - 3/26/2019; Hamburg, Germany. Piscataway, NJ: IEEE; 2019. p. 218–225. doi:10.1109/ICSA-C.2019.00045.
[46] Miehe R, Horbelt J, Baumgarten Y, Bauernhansl T. Basic considerations for a digital twin of biointelligent systems: Applying technical design patterns to biological systems. CIRP Journal of Manufacturing Science and Technology. 2020;31:548–60. doi:10.1016/j.cirpj.2020.08.006.
[47] Broy M, editor. Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme. Berlin: Springer; 2010.
[48] Tanenbaum AS, Bos H. Modern operating systems. Boston (Mass.): Pearson; 2015.
[49] Jung J. 30 Jahre Linux – Ein Rückblick. 08.06.2022. https://www.zdnet.de/88401723/30-jahre-linux-ein-rueckblick/. Accessed 28 Jun 2022.