INTRODUCTION TO THE SPECIAL ISSUE ON CYBERPHYSICAL SYSTEMS

JANUSZ ZALEWSKI[1,2]

[1] *Professor Emeritus of Software Engineering*
Florida Gulf Coast University,
10501 FGCU Blvd S, Fort Myers, Florida 33965, USA

[2] *Ignacy Mościcki State Professional College*
Narutowicza 9, 06-400 Ciechanów, Poland

DOI: https://doi.org/10.34808/tq2021/25.2/0

Cyberphysical computer systems are those, which in addition to their computational functions interact with the environment, where an environment is understood as any entity external to the computer that supplies measurement data and/or receives control signals. This area of research has grown exponentially in the last decade or so, due to several contributing factors, such as, an increase in collecting massive amounts of data for decision making, the actual need for use of sensors in nearly every aspect of our lives, the necessity of making remote operations that include physically moving parts, such as robotic arms, and finally, the emergence of Internet of Things, which appears to combine and integrate all related aspects of computing.

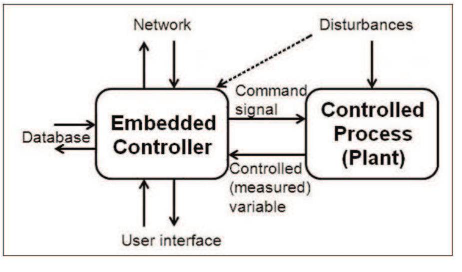

Figure 1. Abstract view of a cyberphysical system

To illustrate it graphically, one can produce a diagram shown in Figure 1, which is an abstraction presenting encapsulated functionality and all major interactions of a cyberphysical system. It can be essentially viewed as an extension of a traditional embedded system, stemming from a single loop feedback control system pattern [1], with added networking functionality, which amounts to the word “cyber”. As one can see on this diagram, the interactions themselves involve so many aspects of computing, including computer networks, databases, human-computer interaction, measurement and control, protection against disturbances, such as threats and hazards, data processing functions notwithstanding, that to give a full account of what is happening in this research area one would need not only a single issue of a journal or a book, but probably entire book series. Therefore, this issue focuses on one particularly important aspect of cyberphysical systems research, which is data acquisition and handling.

A good starting point for it is provided by the first article in this issue, authored by Miroslaw Malek, concerning “Monitorology” understood as the “Art of Observing the World”. It is exactly how the activity of data acquisition and handling – once simple and straightforward – has grown, so he could now invent this new term “monitorology”. In this view, Prof. Malek introduces a hierarchy of computing functions reflecting the layers of processing the data acquired by embedded (read: cyberphysical) systems – activities that lead to extracting gradually: information from data, knowledge from information, and finally, wisdom from knowledge. This leads the author to formulating a number of challenges, which the computing disciplines are facing to harness the untamed proliferation of big data.

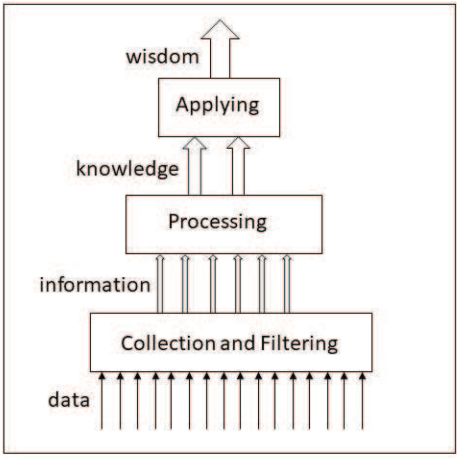

Figure 2. Illustration of transformation processes from data to wisdom

For the purpose of organizing articles in this issue, we have adopted, with Prof. Malek’s consent, his diagram illustrating the hierarchy of data flows, which is reflected in a somewhat transformed picture presented in Figure 2. The first step in handling big data is always their collection and storage, before the transformation of raw data into information can be done. In this issue, this aspect is covered by an article on wireless sensor networks, titled “WSN architecture for building resilient, reactive, and secure wireless sensing systems”, authored by Paweł Gburzyński.

The next stage in the data transformation process, with lots of data on hand, is to convert these *data into information*. This is much easier said than done, since this kind of transformation is crucial to the existence of human beings not only to technical systems. Therefore, in this issue we are focusing only on one feature of this stage, which is – generally speaking – concerning two fundamental software aspects of a computing device used in information processing: operating system and programming language. These aspects are covered, respectively, by Bo Sandén in “Designing multithreaded software based on concurrency in the problem domain”, which addresses modeling of concurrent systems by using his event-sequence modeling, and by Emil Vassev and Mike Hinchey, who – in the article titled “Prototyping Self-healing Behavior for NASA Swarm-based Systems with ASSL” – provide details of their Autonomic System Specification Language (ASSL) and how to apply it.

As these two papers touch the software product from the process perspective, as reflected in the beginnings of their titles, “Designing” and “Prototyping”, respectively, it makes sense to expand more on the software development issues to have a more complete picture of this transformation stage. Two issues are important in this regard, the entire development process and testing. The former is tackled by Franco Ciccirelli and Libero Nigro in their paper titled “A Development Methodology for Cyber-Pysical Systems Based on Deterministic Theatre with Hybrid Actors”, and the latter by Vadim Mukhin et al. in a paper titled “Improved Method of Testing Distributed System Interfaces Using Simulation Tests”[1].

It is always a challenge to anyone – and to software engineers, in particular – to transform *information into knowledge* or even to claim such an attempt being successful. While this stage certainly has multiple aspects, all involving elements of artificial intelligence (AI), machine learning, etc., our goal for this issue was relatively moderate, which resulted in covering the topic by two articles from two important and complementary perspectives: safety and security, which are both intertwined. The paper by Fernando Gonzalez et al. addresses the issue of achieving safety through security by building a “Model for Intelligent Protection of Critical Computer Systems”. The AI aspects concern detecting faults and vulnerabilities in an electric grid and are covered gradually in stages, by applying threat monitoring, data analysis and state prediction for decision making. The paper by Nary Subramanian, titled “Root Cause Analysis of Cybersecurity Incidents on Pipelines Using the NFR Approach”, approaches a cybersecurity problem using the Non-Functional Requirements (NFR) method. It relies on identifying the root causes of incidents by applying their traceability to the system

1. This paper closes Vol. 25, No. 2. The remaining five papers are published in Vol. 25, No. 3 of TASK Quarterly

requirements, identification of synergistic and conflicting operational goals, and historical record-keeping.

Finally, the last transformation in the model from Figure 2 is the most difficult even to relate to, as there is no credible theory or practice of applying or converting knowledge to wisdom. The two papers selected for this issue reflect just these two extremes from the entire spectrum of topics, one is concerning the theory and another concerning the practice. Jan van Katwijk, in his paper titled "Design and Implementation of an Open Source Software Digital Audio Broadcasting (DAB) Decoder" presents a professional approach to practical software development, based on the example of writing open source code. On the other hand, in another paper of this category, titled "Weighted Laplacians of Grids and Their Application for Inspection of Spectral Graph Clustering Methods" by M. Kłopotek et al., the authors outline their study on a plain theoretical subject that could potentially benefit configurability of large-scale cyberphysical systems, such as the Internet of Things.

The issue like this would not be complete if someone did not take a look into the future. What happens if cyberphysical systems grow further, even beyond the Internet of Things, forming larger conglomerates known as systems of systems? What kind of knowledge is necessary to design them? We are lucky to have this topic covered in an article by Ricardo Sanz and coworkers. They do exactly this, look into the structure and details of knowledge required at this level, investigating what role knowledge can and should play to build functionalities of this kind of systems.

Overall, it is our hope that this collection of papers in a special issue brings to the readers a representative cross-section of current research and practices in cyberphysical systems, and reflects the state-of-the-art in this important area of computing. It is also worth noting that one specific subject, *cyberphysical systems education*, fitting well into the top layer in Figure 2 or crossing all layers of the hierachy – depending on the point of view – is deliberately left out from this issue, perhaps making a good topic for a follow-up presentation.

References

[1] Sanz R and Zalewski J 2003 *Pattern-based control systems engineering*, *IEEE Control Systems* **23** (3) 43