Challenges in providing sustainable analytic of system of systems with long life time

Daniel Hallmans
Hitachi-ABB Power Grids
Ludvika, Sweden
daniel.hallmans@hitachi-powergrids.com

Kristian Sandström, Stig Larsson
RISE Research Institutes of Sweden
Västerås, Sweden
kristian.sandstrom@ri.se
stig.larsson@ri.se

Thomas Nolte
Mälardalen University
Västerås, Sweden
thomas.nolte@mdh.se

***Abstract*—Embedded systems are today often self-sufficient systems with limited communication. However, this traditional view of an embedded system is changing rapidly. Embedded systems are nowadays evolving, e.g., an evolution pushed by the increased functional gain introduced with the concept of System of Systems (SoS) that is connecting multiple subsystems to achieve a combined functionality and/or information of a higher value. In such a SoS the subsystems will have to serve a dual purpose in a) the initial purpose that the subsystem was originally designed and deployed for, e.g., control and protection of the physical assets of a critical infrastructure system that could be up and running for 30-40 years, and b) at the same time provide information to a higher-level system for a potential future increase of system functionality as technology matures and/or new opportunities are provided by, e.g., greater analytics capabilities. In this paper, within the context of a "dual purpose use" of a) and b), we bring up three central challenges related to i) information gathering, ii) life-cycle management, and iii) data governance, and we propose directions for solutions to these challenges that need to be evaluated already at design time.**

***Index Terms*—embedded systems, SoS, analytics, data gathering, long life time**

I. INTRODUCTION

Historically, most embedded systems have been designed and built to be more or less self-sufficient units with limited connectivity. These systems have been engineered, verified and validated to serve one purpose, for example, to Control and Protect (C&P) a well defined process, during its life cycle, with limited or predefined communication outside the system. Resent years' evolution in hardware technology, e.g., faster communication and greater bandwidth, and larger data storage, opens for new system functionalities. If it is combined with new analytics possibilities, e.g., utilizing Machine Learning (ML), are together introducing an opportunity to provide Cyber Physical Systems (CPS) with new high value information leveraging data combined from different (sub)systems, e.g., a new sensor interface. Such System of Systems (SoS) are constructed by a collection of functionality that use resources and capabilities from other subsystems to create a new and more complex systems which offer more functionality and performance than simply the sum of the included subsystems. Today, initiatives pushing in this direction include: Smart Manufacturing, Industry 4.0 [1], and Smart Grids [2] .

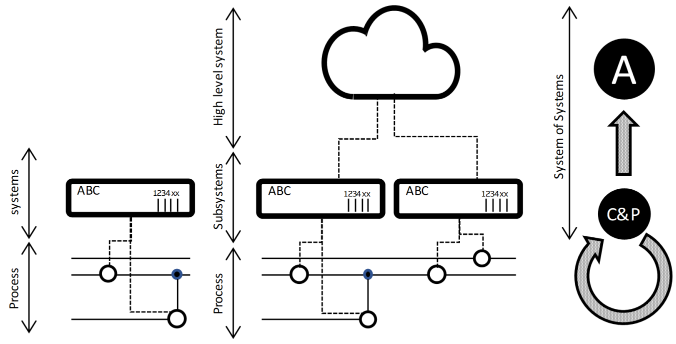

Fig. 1. Left: stand alone system. Middle and Right: Two subsystems with a local C&P task and at the same providing information to a high level analytic function (A), and by that creating a SoS.

The dual purpose use of the subsystems give them two separate tasks, i.e., i) the original C&P of the process that it is directly connected to and ii) to provide the higher level system with information, Fig. 1, e.g., providing an up-to-date Digital Twin of the system.

The architectural complexity arises when the design and deployment of the different subsystems and the higher level systems are not developed at the same time. Making it impossible to synchronize and agree on requirements among the system and its subsystems. Moreover, the system and its different subsystems may not share the same life cycle. For example, subsystems with an expected long life-cycle, e.g., up to 30-40 years, once deployed having a long time between updates. At the same time the higher level system is likely to evolve more frequently, and by that also requiring new sets of information to be available. At design time of the original system, future requirements may only be partially understood with ideas of what the system should provide when new and/or more powerful technology becomes available, but the requirements will also evolve over time making it difficult to impossible to foresee what is needed in the future.

In Fig. 2 the distributed subsystems are represented by discrete systems and the higher level systems as a cloud. Note that the higher level systems do not need to be cloud instances, rather a system on a higher level running on a server, an edge node, or in the fog or cloud. The subsystem still has the "local" functionalities with an input (I), a process of information (P)

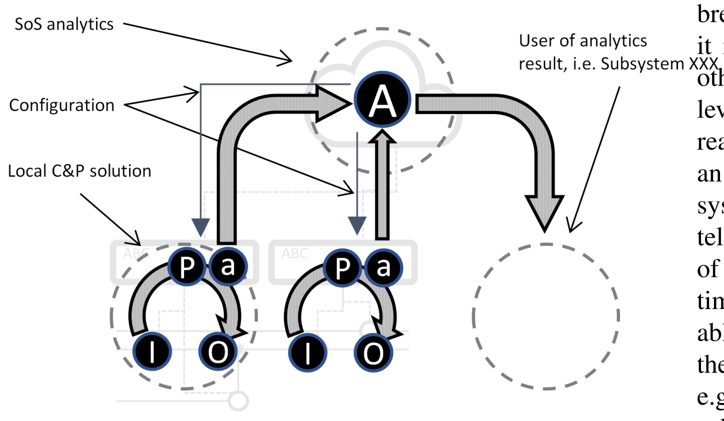

Fig. 2. Local C&P functionality, IPO, with high level processing in the cloud.

and an output (O), all needed to handle the local process. In parallel the subsystem also need to provide information (a), based on a configuration from the upper level system, for creating the SoS/CPS analytics functionality (A). The result could be fed back to the initial subsystems or to yet another system, as depicted in Fig. 2.

With new high performing and low latency communication technologies such as 5G, even more of the initial subsystem functionality could be moved into the cloud. For example, one can consider running part of the control loops in a remote location. However, gathering of information from the physical system may has to be performed locally, to provide information needed to the ”dual” purposes of C&P and analytics.

There are already today several types of systems where the users can improve and/or extend its functionality by creating a SoS. Some of these systems are critical infrastructures, such as a substations in energy transmissions, power plants, trains, and health care systems. Within these systems their local information could have a high value in a new use-case, or only to get information to understand and improve the local process. In an electrical transmission system one such example could be a protection relay, i.e., an Intelligent Electronic Device (IED), where the inputs (I) would be voltages and currents measured at different positions in a substation, processing (P) the information to, for example, make sure that there is not a too high current registered, and if that is the case take an action (O), e.g., tripping a high voltage circuit breaker. Another example of a subsystem could be an Anti-lock Break System (ABS) available as a standard functionality in most cars. It measures the speed of all 4 wheels (I), processing (P) the information, i.e., detecting a sudden locking of a wheel, and then taking an action (O) to reduce the hydraulic pressure connected to the break system for the corresponding wheel.

Both examples above represent a vital part of their respective installation with a limited communication, mainly predefined, to the outside world. From a SoS perspective, the information included in the substation protection system, e.g., voltages, currents, frequency, temperature, protective near misses, and in the ABS system, e.g., vehicle speed, high speed breaking, loss of grip,is valuable information. Especially when it is combined with information from other systems such as other vehicles, traffic lights, etc., it would then generate a new level of understanding of the present situation, e.g., was the reason for the breaking of the vehicle a traffic light or is it an indication of a possible upcoming traffic situation? Similar systems exist already today that are utilizing information from telephones, e.g., Google maps[1] and Waze[2], used by millions of users every months. They are collecting map data, travel times, and traffic information from users and by that being able to provide routing and real-time traffic updates, based on the information that all users send anonymous to the server, e.g., speed and location. By adding information from more subsystems, e.g., the ABS, in combination with new analytics functions, e.g., by using Machine Learning on large data sets to solve complex relations, information to drivers could be extended even further with for example ”road traction” information, based on the ABS data from vehicles that have already passed the corresponding section of the road. Since the value of the new information, in some cases lifesaving, would be high, also the push from the potential 3:rd party company or governments, to get access to the needed information, will be high. Most probably the value of saving a number of lives per year, or reducing the number of incidents, will have a higher value compared to the increased effort and complexity in the subsystem to retrieve and provide the information without jeopardizing any of the initial/original functionality, e.g., of an ABS system.

A large extent of future systems, and also to some extent present systems, cannot avoid to handle both its local stand alone functionality, and at the same time, with a high data quality, provide information that enable future SoS functionalities. Therefore, in this paper we analyze important challenges when designing a system where information also should be available for a potential future SoS implementation. The availability of information should also be done in a sustainable way, e.g., not needing a costly rebuild of the system.

The outline of the paper is as follows: Section II describes major challenges with designing a subsystem that should be able to handle future analytic requirements. Section III investigates and analyze key challenges related to such a design. Related work is presented in Section IV. Finally, Section V concludes the paper and point out future work.

II. PROBLEM DESCRIPTION

When designing a system that should be able to provide information to future analytic needs, not fully known at design time, we highlight three major challenges:

1) The quality of the analytic result has a direct dependency to the *quality of the used information*, e.g., from a sampling perspective, thereby requiring the subsystem to provide such information in competition with the subsystem base functionality.

---

[1] https://en.wikipedia.org/wiki/Google_Maps

[2] https://en.wikipedia.org/wiki/Waze

2) Development of new analytic functions, in the higher levels of the system, are likely to follow/align with *different life-cycles* than the subsystems themselves. This will lead to that new features on system level are likely to require new data sets from the subsystem level more often than the frequency of regular updates of the subsystems.
3) Last but not least is the *access and understanding of the information available* in all of the subsystems. If every device around us would provide an data access API how could we then know what information it is actually providing us with or how to configure it? Who is the owner of the information?

III. ANALYSIS

Given the problem description as outlined in the previous section we have divided the analysis of the problem into three parts: A. Impact on subsystems, B. Different life-cycles, and C. Understanding and access to information. In the following each part will be investigated and discussed in more detail.

A. Impact on subsystems

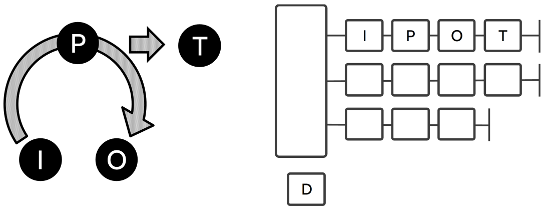

Fig. 3. Process with I, P and O, with adding of transient store of high frequency data (T). Right part shows corresponding tasks in a scheduler with three different priority levels. Additional a fault recorder (T) is added to the scheduled tasks and a Debug interface (D) outside of the scheduler.

Providing generic access of information from a system must be planned already at design time. Most of the systems in use today already have some kind of API that gives access to, e.g., none real-time data such as debug information (D in Fig. 3). Other examples include pre-compiled settings for data transfer to, e.g., an HMI, or built in data gathering that is more used for fault tracing once an instance has triggered the storage of information, e.g., a Transient Fault Recorder (TFR) (T in Fig. 3) used in a substation IED to store data at a high frequency. Configured to store a pre and post fault recording of information, e.g., current, voltages, calculated values, outputs, at a high sampling rate when a specific event occurs, e.g., an action of a protection, to be used for post analysis of the fault. To make the gathering more generic and open will require some guessing and ability to change what type of data to store, since all use-cases are not known at design time.

Building an analytic function will require access to signals with different characteristics, e.g., time stamped information, different measured accuracy, differences with respect to number of samples per second, requirements on concurrent sampling in different systems, all requirements depending on the intended analytic function. A temperature that changes very slow would not require as accurate time stamping or sampling as an event that changes value extremely fast and at the same time needs to be compared to similar measurements in other systems. To fulfill this type of requirements the scheduled tasks in the subsystems also need to be able to handle the gathering of analytic information at the same level as its initial C&P tasks (I, P and O), see Fig. 4, giving the analytics function an equal importance when it comes to hardware and software resources as the C&P, motivated by the potentially high added value introduced by the SoS context.

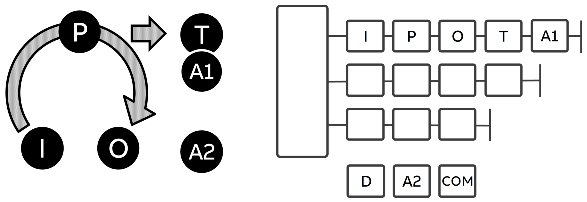

Fig. 4. Expanding the process with an analytics gathering module, A1 and A2, and a configuration and communication module, COM.

To the initial scheduler, depicted in Fig. 4, a data gathering and pre-processing task (A1) is added on the same priority level as the sampling and processing of data to make sure that every single sample can be handled if needed. In our example an additional task is also added, (A2), that operates more on free time, as a complement for really slow changing signals to not affect the scheduling priority. Additionally communication and configuration (COM) has to be added to handle transfer of information and configuration of the data set to collect, since the data set will not be static during the life cycle of the system. Different analytic functions will also require the data to be collected in different ways, e.g., burst of high frequency data, time stamped (supported by a time synchronization module), maximum/minimum/median value during a time period, filtered values or pre-processed values to reduce the communication overhead, etc.. If COM is located at the scheduling level also real-time streaming of analytic data could be implemented, giving a constant stream of data at the same frequency as the data gathering.

If we are only interested in the sampled values (I) and not the internal values generated by the process (P) we could think of having a separate parallel hardware, i.e., an IoT-Sensor[3] that samples the same signal independent on the C&P functionality. A positive effect of such a technical solution would be that any changes to the analytic gathering would not affect the C&P functionality. Negative, however, is a higher cost, potentially larger space requirements along with the inability to support any use case involving improving or using the internal process functionality (P) if the same solution is not also replicated in the parallel hardware.

Sharing hardware, or providing separate hardware, will limit the amount of information that can be gathered / processed

---

[3] https://en.wikipedia.org/wiki/Internet_of_things

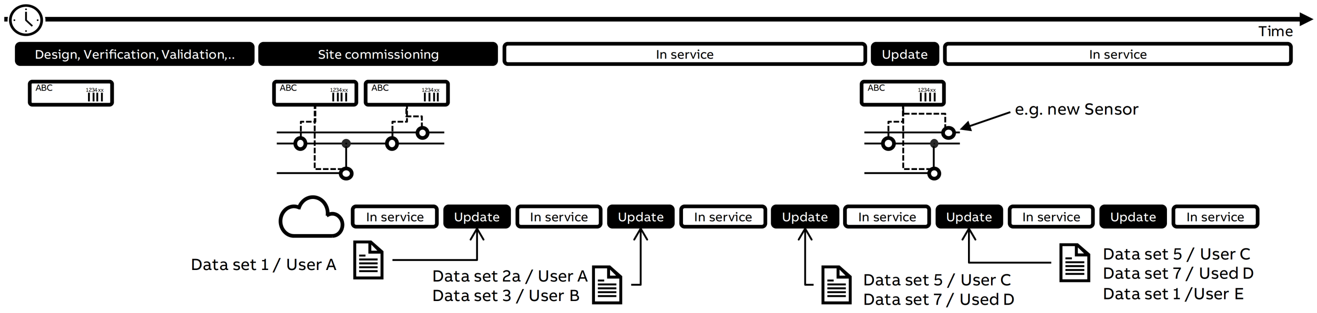

Fig. 5. Subsystem life cycle and update frequency, top, vs. high level system feature update, lower part of figure.

due to hardware limitations, e.g., CPU cycles, memory, bandwidth, before it will affect and/or interfere with the original functionality of the system. If the critically of the high level system functions, i.e., SoS analytics, is of an equal or higher importance than the subsystem, then the analytic part needs to be prioritized. If there for example is shortage of hardware resources during a short period of time due to a sudden cache eviction, instead an upper boundary of number of analytics functions needs to be set already at design time and also be in use in the initial verification and validation of the system, to avoid any future problems when the analytics functions are used by other systems. Similar limitations also apply for COM since the collected information also needs to be transferred to the SoS level.

B. Different life cycles

Building an embedded system to be used in a critical installation, e.g., a power transmission system, a vehicle, or any system with a high level of complexity, involves a large number of activities spread out during a number of years, e.g., design, manufacturing, subsystem verification and validation, system testing, and type testing according to standards. Initial design requirements are set years before the system is deployed on site or the vehicle is delivered to the customer. Updates to the system is after that seldom needed to be done, e.g., bug fixes, new features, or they are done at given maintenance intervals, with a few years in between.

The SoS analytics, on the other hand, is enabled and is starting to evolve once the system is taken into use. When the analytics process starts, insights from the gathered data from the different subsystems, will most likely lead to changes in requirements of new data sets, e.g., new signals to be collected, new preprocessing algorithms, changes in sampling frequencies. All these changes in requirements must be, in the end, delivered from the subsystems, generating new configurations that have to be downloaded, see an example lift cycle in Fig. 5. A typical data science framework [3] is *CR*oss *I*ndustry *S*tandard *P*rocess for *D*ata *M*ining (CRISP-DM), including six phases (business understanding, data understanding, data preparation, modeling, evaluation and deployment), all depicted in Fig. 6. CRISP-DM describes the data science life cycle, including an iterative process when it comes to preparing, using and evaluation the information used to be able to update future sets of data to gather.

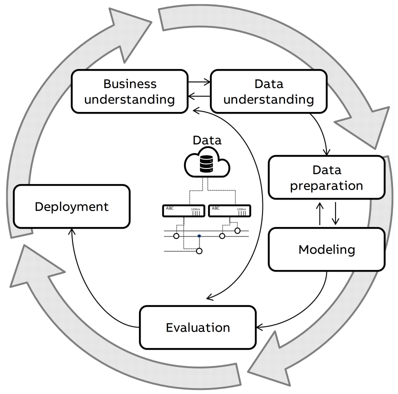

Fig. 6. CRISP-DM framework describing the iterative process of data science.

The analytics update frequency will be higher compared with the frequency that the previously mentioned underlying embedded system will change, giving a much shorter life cycle and in most cases also updates without a service stop to reduce the spent latency until a new analytics feature can start to be used, resulting in a need for the system to be prepared for this type of updates already at design time. One note when it comes to life-cycles, from a cyber security perspective, is that new vulnerabilities are found every year and must be handled, raising a need for dynamic and reoccurring updates if the communication channel with the world outside the system should be allowed to be used.

C. Data governance, understanding and access to information

Data is sometimes defined as being of high quality if it meets the requirements for its intended use. When using data for machine learning in a CPS this also means understanding the data with respect to the physical system, including temporal aspects. As an example a raw sensor value might be of little value if one does not know that it is a temperature reading in degrees Celsius, of a fluid from a given pipe, with some resolution, measured at a specific time.

A challenge related to long lived systems is that the intended use (of data) might be poorly understood, and therefore defining and providing a sufficiently detailed model of the data might be challenging. A description not only need to include the physical parameters, e.g., vehicle speed in km/h, but also the way the value could be sampled, e.g., 1kHz with a second order low pass filter with a cut off frequency of 0.1 Hz, Maximum and Minimum value during last hour Additionally the quality of the measurement, e.g., time stamping with an accuracy of 0.01 second, need to be understood. Since the systems are not designed at the same time a guessing and configurable solution may have to be implemented when it comes to what data and how, since all data points can not be stored for ever and at all times.

Limiting the number of data points that can be measured opens up for the next challenge, the governance of what data to measure. Is it the subsystem owner, e.g., vehicle owner, or initial manufacturer or SoS builder that can demand a specific set of data, when for example the interest is in saving lives in a traffic situation. Are we by using the vehicle, or other devices that are connected, signing off to the SoS that they can use the created information, since the initial manufacturer built in ability to gather information.

IV. RELATED WORK

Within the area of CPS, Digital twins and SoS there are a lot of research ongoing, with connections to the challenges of ”dual use”. ISO/IEC/IEEE 21839 [4] is describing the life cycle considerations of a System of Interest (SoI), i.e., the subsystem, that is part of a SoS. The SoI need to function effectively in the operational or business environment it is located in and at the same time be able to contribute to the SoS functionality. During the different life cycle stages of the SoI, e.g., development, utilization, support and retirement, considerations need to be taken for the SoS interaction, interoperability and technical trade-offs. [5] combined the utilization, support and retirements stages, as described in ISO/IEC15288 [6], into an evolution stage that describes more of a long-term effect of a “dual purpose” use in a system that evolves during a longer time. In [7] [8] challenges around the control perspective, e.g., with an external control loop, system predictability to be able to handle real time tasks, and adaptability, is highlighted and will be key challenges when handling the access of information. [9] is describing the CPS from an Industry 4.0 perspective with challenges in closely monitored and synchronized between physical factory floor and the cyber computational space. [10] brings up the question of information retrieval in a cyber-physical production system as a challenge.

Using of data in higher level systems brings up another set of key challenges connected to cyber security, [11] gives an overall study based on different use-cases, e.g., smart-grids and smart-vehicles. [12] [13] uses Vehicles as a good example of the complexity in a CPS with 50-70 electronic control modules, where the electronic break control module used for ABS systems is one, and how the connectivity will be a challenge from a CS perspective.

For analytics the gathering of data will be the key challenge and it can be done in different ways, as described in [14] as implemented in the original hardware, additional hardware and smart sensors at a higher cost, highlighting some of the challenges when it comes to getting information from a subsystem. [15] also added the aspect of long life cycles and adequate data is mostly unavailable critical infrastructure when building a digital twin. Data communication and communication middle layer comes back in several of the papers, e.g., [16] with a data layer to be able to handle problems with heterogeneous systems, and different communication methods, e.g., OPC-UA and Automation ML in [17].

V. CONCLUSION AND FUTURE WORK

With an increased focus on the possibilities and insights that SoS setups can give us, we are moving into a situation where self-sufficient systems provide information to a system on a higher level. Vehicles, substations, trains, planes, etc. are not just responsible to handle the task that they initially was designed for, but they must also enable systems and functions on a higher level that has a completely different life cycle and purpose.

The initial system was designed, built and tested during a number of years until it was deployed into service. Once deployed and in service the SoS analytics data gathering start, providing information to the higher level systems. With this dual purpose need, C&P of local assets vs. SoS analytics, we see three central challenges,

i) at subsystem level, i.e., scheduling of tasks, the data gathering need to be implemented in such a way that it does not disturb the C&P functionality but at the same time also allow for gathering of information with a quality that is enough for future analytics purposes.

ii) The sub system will have a long life cycle, some times up to 30-40 years, with few updates during the time period. At the same time the analytics requirements will change during time giving a much higher update frequency. Giving a requirements to the subsystem to be able to handle new analytics requirements without updating the original system.

Last, *iii)*, the data governance need to be handled with access and modeling of the information available.

In the three areas that we have analyzed we see several important future research challenges that need to be investigated, e.g., on the embedded side we need to be able to have the two different systems, C&P and data gathering, to work hand in hand without disturbing each other while they are sharing the same information. An example of possible way forward for the communication part could be by using Time Sensitive Networks (TSN) to handle the communication prioritization and by that allow for a shard communication solution with a deterministic guarantee. Can we align the different life-cycles better by using e.g. DevOps, with a challenge that all devices are probably not from the same manufacturer and that all functions are not available for adaptation later on during the life-cycle. Is there a universal language, or should it be developed, to describe all connected systems configuration and data attributes for each data point and also knowing who is the owner of the information? Can different reference frameworks, e.g., the reference Architectural Model Industry 4.0 (RAMI

4.0) [18] or for the Industrial internet of thing [19], be used to address some of the challenges.

In summary, we see that a large extent of future, and also present, self-sufficient systems cannot avoid handling both the local functionality, and at the same time, with high quality, provide information to enable SoS functionalities. To allow this to work in a system that is built to last for 30-40 years without significant updates, the SoS interaction and gathering of information has to be in preparation already at design time, even if the future requirements are not fully known.

ACKNOWLEDGEMENTS

This work was supported by the Knowledge Foundation (KKS) via the projects ITS-EASY and FIESTA.

REFERENCES

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, 2016.

[3] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1. Springer-Verlag London, UK, 2000.

[4] "Iso/iec/ieee international standard – systems and software engineering – system of systems (sos) considerations in life cycle stages of a system," *ISO/IEC/IEEE 21839:2019(E)*, pp. 1–40, 2019.

[5] D. Hallmans, M. Jägemar, S. Larsson, and T. Nolte, "Identifying evolution problems for large long term industrial evolution systems," in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*. IEEE, 2014, pp. 384–389.

[6] I. ISO, "Iso/iec 15288: Systems and software engineering—system life cycle processes," *ISO, IEC*, pp. 24 748–1, 2008.

[7] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems." *KSII Transactions on Internet & Information Systems*, vol. 8, no. 12, 2014.

[8] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*. IEEE, 2008, pp. 1–9.

[9] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters*, vol. 3, pp. 18–23, 2015.

[10] L. Monostori, "Cyber-physical production systems: Roots, expectations and r&d challenges," *Procedia Cirp*, vol. 17, pp. 9–13, 2014.

[11] Y. Jiang, S. Yin, and O. Kaynak, "Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond," *IEEE Access*, vol. 6, pp. 47 374–47 384, 2018.

[12] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915, 2017.

[13] S. H. Bouk, S. H. Ahmed, R. Hussain, and Y. Eun, "Named data networking's intrinsic cyber-resilience for vehicular cps," *IEEE Access*, vol. 6, pp. 60 570–60 585, 2018.

[14] N. Jazdi, "Cyber physical systems in the context of industry 4.0," in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, 2014, pp. 1–4.

[15] M. Borth, J. Verriet, and G. Muller, "Digital twin strategies for sos 4 challenges and 4 architecture setups for digital twins of sos," in *2019 14th Annual Conference System of Systems Engineering (SoSE)*. IEEE, 2019, pp. 164–169.

[16] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-cps: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88–95, 2015.

[17] T. Bangemann, M. Riedl, M. Thron, and C. Diedrich, "Integration of classical components into industrial cyber–physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 947–959, 2016.

[18] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," *ZVEI*, vol. 2, no. 2, p. 4, 2015.

[19] M. Weyrich and C. Ebert, "Reference architectures for the internet of things," *IEEE Software*, vol. 33, no. 1, pp. 112–116, 2015.