MÄLARDALEN UNIVERSITY
SWEDEN

PhD Thesis Proposal

Meeting Technical Challenges in Long-lived Industrial Systems Considering Emergent Trends

Daniel Hallmans
School of Innovation, Design and Engineering (IDT)
Mälardalen University
daniel.hallmans@mdu.se
September 2022

Main Supervisor:	**Thomas Nolte**
Co-supervisors:	**Kristian Sandström**
	Stig Larsson

Abstract

This is the PhD proposal of Daniel Hallmans.

List of publications

Papers included in the Thesis:

- **Paper A**: *Challenges in providing sustainable analytics of systems of systems with long lifetime.* Daniel Hallmans, Thomas Nolte, Kristian Sandström, Stig Larsson, SoSE 2021

- **Paper B**: *Design considerations introducing analytics as a "dual use" in complex industrial embedded systems.* Daniel Hallmans, Thomas Nolte, Kristian Sandström, Stig Larsson, Niclas Ericsson, ETFA 2021

- **Paper C**: *Analysis of the TSN standards for utilization in long-life industrial distributed control systems.* Daniel Hallmans, Thomas Nolte, Mohammad Ashjaei, ETFA 2020

- **Paper D**: *Clock synchronization in integrated TSN-EtherCAT networks.* Daniel Hallmans, Thomas Nolte, Mohammad Ashjaei, Daniel Bujosa Mateu, Alessandro V. Papadopoulos, Julian Proenzaz, ETFA 2020

- **Paper E**: *Consistent sensor values on a real-time Ethernet network.* Daniel Hallmans, Thomas Nolte, Kristian Sandström, Stig Larsson, In Proceedings of the 12th IEEE World Conference on Factory Communication Systems, WFCS, 2016, May

- **Paper F**: *Identified challenges and opportunities with cyber security standard compliance in combination with a long-expected lifetime.* Johan Malmström, Daniel Hallmans, Jeremy Morgan, Cigre 2022

Papers not included in the Thesis

- **Paper 1**: *Identifying evolution problems for large long term industrial evolution systems.* Daniel Hallmans, Marcus Jägemar, Thomas Nolte, Stig Larsson. In Proceedings of the 38th IEEE International Conference on Computers, Software& Applications (COMPSAC), 2014, July.

- **Paper 2**: *Industrial requirements on evolution of an embedded system architecture.* Daniel Hallmans, Thomas Nolte, Stig Larsson. In Proceedings of the 37th IEEE International Conference on Computers, Software & Applications (COMPSAC), 2013, July.

- **Paper 3**: *A method for handling evolvability in a complex embedded system.* Daniel Hallmans, Thomas Nolte, Stig Larsson. In Proceedings of the 18th IEEE International Conference on Emerg-ing Technologies & Factory Automation (ETFA), 2013, Sep-tember.

- **Paper 4**: *A method and industrial case: Replacement of a FPGA component in a legacy control system.* Daniel Hallmans, Thomas Nolte, Stig Larsson, Kristian Sandström. In Proceedings of the 13th IEEE International Conference on Industrial Informatics (INDIN), 2015, July.

- **Paper 5**: *Challenges and opportunities when introducing cloud computing into embedded systems.* Daniel Hallmans, Thomas Nolte, Stig Larsson, Kristian Sandström. In Proceedings of the IEEE International Conference on Industrial Informatics (INDIN), 2015, July.

- **Paper 6**: *GPGPU for industrial control systems.* Daniel Hallmans, Thomas Nolte, Markus Lindgren, Kristian Sandström. In Proceedings of the 18th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA), Work-in-Progress Session, 2013, September.

- **Paper 7**: *Towards using the graphical processing unit (GPU) for embedded systems.* Daniel Hallmans, Thomas Nolte, Mikael Åsberg. In Proceedings of the 17th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA), Work-in-Progress Session, 2012, September.

- **Paper 8**: *Applicability of using internal GPGPUs in industrial control systems.* Markus Lindgren, Kristian Sandström, Thomas Nolte, Daniel Hallmans. In Proceedings of the 19th IEEE International Conference on Emerging Technology and Factory Automation (ETFA), 2014, September.

Contents

1 Introduction

Process and power industries are these days more and more dependent on complex embedded control systems to meet an increasing demand when it comes to efficiency and effectiveness of their product portfolio. When these products are delivered and installed at the customer they become components in a highly complex system and are expected to remain operational for many years, in some cases up to 30-40 years. Hence, such systems are subject to challenges inherent of long life-cycles.

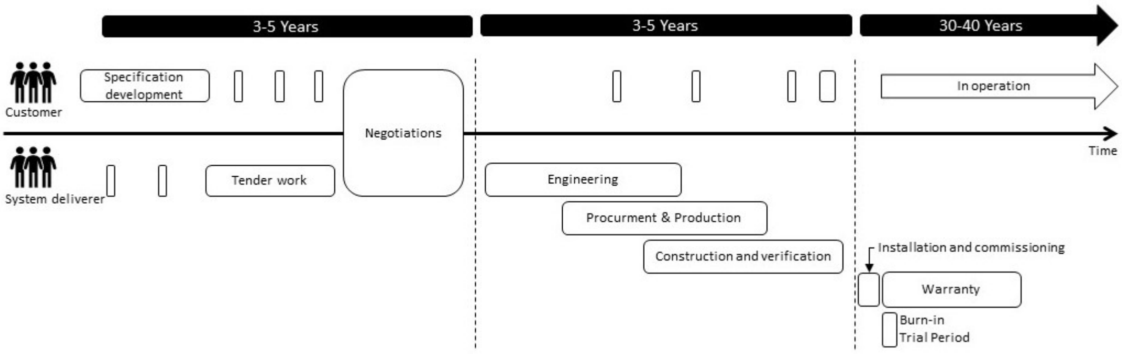

Figure 1: Example of a system delivery process for a large and complex system.

1.1 New system delivery

Looking more closely at the life of such systems we can see that initially the system is sold to a customer according to a specification/contract, setting a functional base to what the system needs to be able to achieve during the complete lifetime of the system, e.g., control and protect of a transmission system or controlling one or several machines in a large industrial plant. Since selling the system is done in competition with other suppliers, concern need to be taken to create a complete system package not just focus on the system life-cycle, e.g., performance, cost, delivery time, features, and specification adherence, is also important. Only if the total ”score” is better than the competition the project will be an success and sold. Figure 1 displays a typical time line for building a transmission system. Starting with three to five year phase where the projects needs and possibilities are formalized, leading to a tender process. Once the tender process is over and an agreement is formed the process to build the system starts and continues with several different phases during another three to five years. Please note that the design of the used control and protection systems are designed and maintained in a parallel process and that this part is only the construction of the system part, e.g., combining the functionalities from different control and protection modules. Last phase includes the installation and commissioning before the system is taken into operation, including the burn in, trial and warranty periods. The system will now be in the owners (customers) use for the next 30-40 years. All through the process there is a constant interaction between customer and manufacturer, initially to set up the project in such a way that the most value is generated, during construction to make sure that correct requirements are met, and during operation for support and managing improvements.

To constantly strive for a better system solution pushes the manufacturer to use new technologies, and not only to play it safe and reuse old solutions. Part of the reason for this is the potentially higher evaluation scores of new system solutions with improved

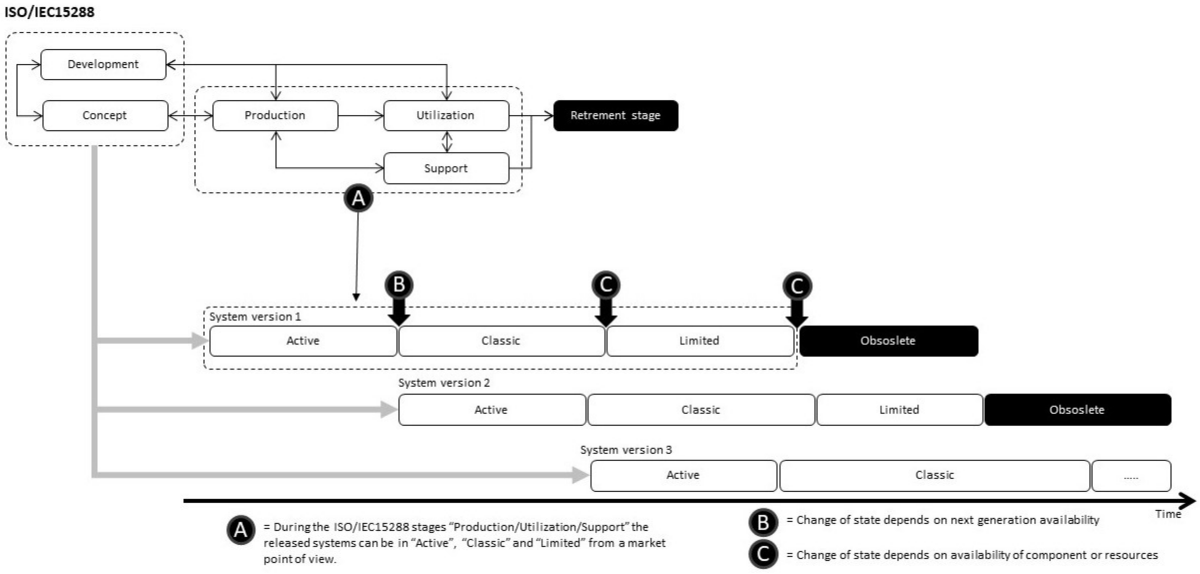

Figure 2: Top/left: ISO/IEC15288 life-cycle process stages. Bottom: example of life-cycle stages for three product releases with corresponding life-cycle stages.

performance. The following example will put this into the context of the automotive industry. Imagine that the car manufacturer Volvo was still selling the Volvo 240 as a brand new car, without any evolution since its introduction in the middle of the 1970ties. Although, now in competition with today's cars, equipped with the latest evolution of technologies, e.g., safety systems. The foundation of a good design can stay on the market for a long time, e.g., Volvo 240 was sold until 1993, when it is maintained and upgraded along the way, for example by adding anti-lock brakes, airbags, head rests, etc. At some point in time, several parts of the complete product, e.g., the vehicle platform, manufacturing tools, deign tools, etc., will be in a state where the cost of maintenance is to high with respect to the gain that the product is giving compared to, e.g., a new electrical vehicle platform. Any embedded system producer will be in the same situation, where the embedded systems' platform need not just be maintained, i.e., replacement of obsolete products, but rather would require a replacement with a completely new generation. These far reaching changes will depend not only of the pass of time but also, as mentioned, on market competition and rise of new system technologies. An example of such a technology change is the introduction of the power electronics IGBTs in power converters as a complement to Thyristors. This change in technology also led to the requirement of new levels of performance from the control and protection systems. Figure 3, shows an example with three different generations of a system that partly has inherit from the previously system generation while also adding new functionality, thus growing in size for each generation (A). At a given moment in time several system generations will be active on the market simultaneously (B), since the installations will be in operation for 30-40 years and the active live cycle of the systems are less than 10 years, before replaced by a new generation.

1.2 System in operation

Once the system is in operation on site the evolution can be divided into two different types. The first type is to maintain the system in its original form without changing its behavior, design, or specification. An example of such maintenance would be to replace obsolete components without changing the original specification and related requirements. Depending on the length of this phase several of the components that the system is based

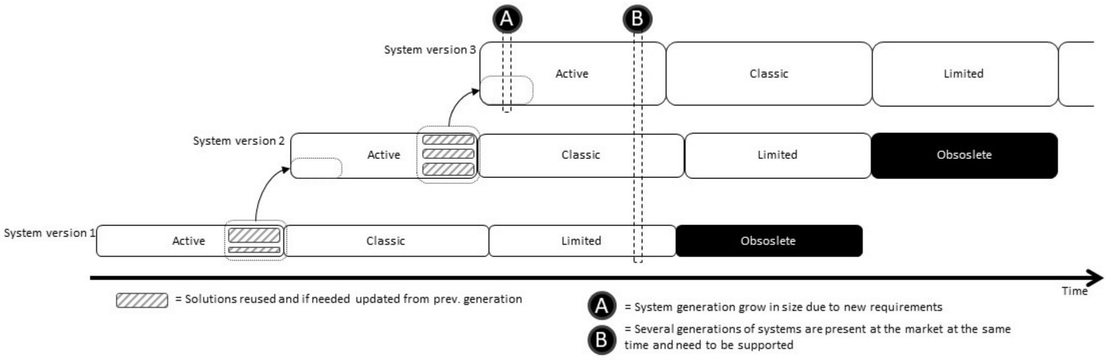

Figure 3: Simplified picture of the evolution of a system through generations, where parts of the system is migrated to next generation and new designs are added. At one specific time, B, several generations of the system will be used in different systems around the world.

on are in the risk of getting obsolete. This is true not only for hardware components, but also for software, e.g., Microsoft Windows has a life-cycle that relies on transitioning to new generations of the operating system and by that making the previously version obsolete. Not only access to components but also knowledge, can be lost in time, e.g., when staff skilled in design, test, or operation of the original system become retired. When working with this types of updates the intention is keep the original function, but the replacement part will in most cases not completely replicate the original, e.g., a new CPU will not run exactly the same binaries as the ten-year old CPU it is replacing. To isolate the changes the function on some system level has to be kept intact. The other way around, software obsolesces can be a trigger for changing hardware since e.g., older compilers are not able to run on later generations of operating systems. Once the function is updated it has to be verified in its original environment to make sure that all previously relevant functionality is intact, which for a larger system can be complicated. At some point in time during the system's life-cycle, it will not be possible, based on economical reasoning for the customer and manufacturer, to maintain the system as is. Instead a larger update is needed, i.e., a complete upgrade of the system, to the latest available technology and by that "resetting" the wheel of time to a situation where it is possible to maintain the system again.

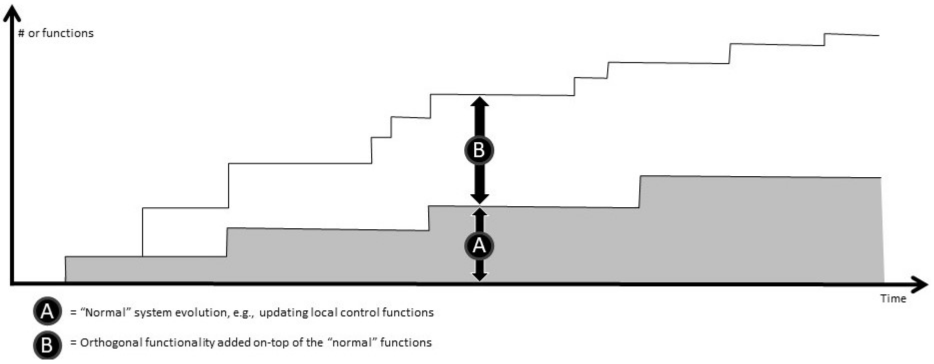

Figure 4: picture of normal vs. orthogonal evolution/functionality.

1.3 Evolution of system functionality

A second type of system evolution typically starts a few years after the system is delivered, in parallel to the maintenance stage, and is characterized by the evolution of the system functionality. The change of functionality could be in line with the original system design, e.g., adding or removing transmission lines connected to a substation that may require changes in, for example, already existing protection settings and adding new control and protection cubicles. The new systems are then interacting with the original system in the same way as the original functions did. This would result in a functional growth but based on solutions that were built into the system from the beginning and hence more or less foreseen at design time. New functional requests could also be more "orthogonal" to the existing system functionality, and therefor will require more disruptive changes, since not originally anticipated. This include functionality not giving any direct value to the original function, e.g., controlling the RPM of a water pump, but rather more indirect value, e.g., securing the system from cyber threats, providing feedback to a higher level system, increase safety levels to minimize risk of casualties. Industry 4.0 [12] is an example of an initiative where data from lower levels of the automation triangle is used to gain benefits on complete system level, e.g., to increase flexibility in manufacturing, revenue, new business opportunities or models. Within the power and transmission sector a similar initiative is Smart grids [32] with the intention to improve the energy usage in our grids by utilizing data from controllers in subsystems to use it to regulate higher level energy flows. In the car industry a similar development is seen when data is aggregated from many cars for use in higher level systems, e.g., assessing road conditions, and by that saving lives. When the value of making use of data from subsystems will be high, e.g., saving lives, the push to share information from the subsystems will increase, and it will be hard from a subsystem point of view not to follow this trend due to the gain/profit on other system levels. Since many, or most, of the subsystems are already existing infrastructure, e.g., transmission control and protection systems used in our grids today, cars that are already available on our roads, and control systems in existing industrial plants, most of needed data should already existing and you do not rebuild the complete system once the need to use the additional data are added, instead the new functionality has to work in parallel and orthogonal to the old.

1.4 Definition of orthogonal function

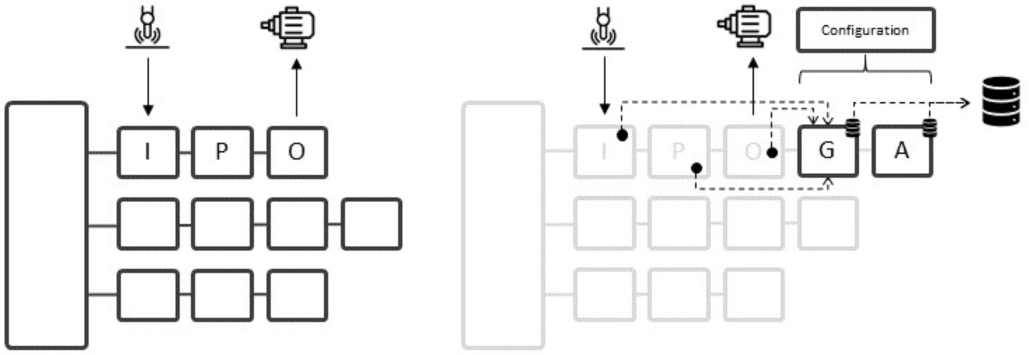

Figure 5: Left part of picture represents a three level scheduler with an input task (I), process task (P) and output task (O). In the right part also gathering of information (G), analytics (A) and configuration has been added as orthogonal functionalities that is not interfering with the original tasks (I,P and O) but rather using the information from the tasks.

An embedded system has typically one main task, e.g., to control and protect a physical asset. To be able to provide such functionality, a number of different functions have been added to the system, e.g., sensors, processing tasks, and outputs, generating a closed loop control. If additional functions are added to the embedded system that are not directly contributing to the main task, i.e., the closed loop control, then they are considered to be orthogonal to the system's main functionality. An example of such orthogonal functionality is cyber security that has no value for the closed loop control but is in several cases a prerequisite/must for the system to be allowed to deploy as a solution within a certain business segment, e.g., critical infrastructure. The orthogonal functionality will have a dependency with respect to the original functionality, e.g., in the case of cyber security such functionality can be required in order to secure the communication to/from the embedded device and by doing that it will also influence the communication solution/performance. Other examples of orthogonal functionalities could be data collection where the data is intended not the to be used directly in the closed loop control but rather on higher levels of the system or in other systems.

With an expected long life-cycle the requirements for orthogonal functions will vary over time and in most cases it is not possible to anticipate such functionality at the system's first design time, e.g., 10-15 years back. Instead these systems are required to be updated with orthogonal functionality continuously throughout the system's life-cycle.

1.5 Summary

In summary, a complex embedded system with an expected long life-cycle undergoes several different stages during its life-cycle and each of these cycles have their own challenges. From the initial stages including design of the embedded system, via system negotiation, system design, system commissioning, until the embedded system is taken into use, where the next phases starts with maintenance and the start of evolution of the systems' functionality. The evolution of the system can be divided into two different parts, one that expends the system with the same or similar functionality that already was existing in the embedded system. Or by introducing an orthogonal use-case.

In this thesis we have identified challenges connected to expanding an embedded system with new orthogonal requirements and we have evaluated technologies and architectures that could simplify the implementation going forward.

2 Challenges in industrial systems with an expected long-life considering emergent trends

2.1 Goal

The complexity of embedded systems is constantly increasing with the introduction of new technologies such as cloud computing and many-core processors, by adding new or modified requirements, e.g., related to Cyber Security, or through new use-cases, e.g., information gathering used for analytics by other systems. Some of these new use-cases is orthogonal to the original use-cases and there can be a requirement to add these to new systems or to already existing system. The overall goal of this thesis is to improve our knowledge how such ortogonal functionality in new use-cases can be introduced and maintained in an complex embedded system with an expected long life-cycle.

2.2 Challenges

In order to achieve the overall goal of this thesis we have identified four central research challenges that should be addressed. For each of these challenges we present one or several research contributions, each of which is presented in one or more peer-reviewed scientific publications.

The effort to building embedded systems includes a number of challenges, from development, maintenance, upgrade, to recycling. If the expected life time of the system spans as much as 30-40 years, the magnitude of these challenges will increase. During this long life time the system solution moves into an evolution stage, [9] where new systems are deployed and maintained during its life-cycle. Challenges will arise in, e.g., handling of hardware and software that are becoming obsolete, personnel that are changing their assignments within the company, that change company altogether, or eventually retire. Other challenges relate to the introduction of new functionality in the system, there may also be changes to regulatory frameworks, and new business models may be introduced.

- C0: identify a relevant subset of the challenges that arise when long life-cycles need to be considered. The findings are foremost based on organizational prerequisites, experience of the author, the supervisor team, and from fellow students at the industrial PhD school.

During the system life-cycle, from installation to the first expected larger upgrade, after 10 to 15 years, requirements and regulations will change. These changes are outside the original specification, and new functionality is typically asked for to be implemented in, or as an extension to, the already existing hardware and software systems. This include, e.g., changes in cyber security regulations or new requirements for data analytic that is orthogonal to the systems' original functionality.

- C1: Select one or more areas that include trends with orthogonal use-cases, motivated by business needs, and analyze the challenges that arise from introducing these use-cases to existing system designs and already installed systems.

- C2: Based on the challenges in C1, in what ways can new orthogonal functionality be introduced in an existing system?

Systems with an expected long life time will have the possibility to, and in some cases will be forced to, use several different generations of technologies during its complete life-cycle. An example of such an technology change is the introduction of multi-core CPUs, providing the system with more computational power and at the same time enabling true parallel execution of functions in one CPU. Legacy code either has to be ported and updated to allow for parallelism to take advantage of the new CPUs or the additional CPU cores can be used to implement new functionality, in some cases orthogonal to the original system, or consolidate functionality several CPUs to one multi-core CPU.

Time Sensitive Networking (TSN) has been identified by several businesses as a future technology standard for Ethernet based communication. Communication is and will also in the future be a critical part of embedded systems and a key enabler of future orthogonal solutions since in several cases the created information will be used outside the core functionality, e.g., the data may be used by big data analytics. We have used TSN as an example of a new technology and how it well it can be integrated into an existing system, the new technology need to support the life-cycle aspects of the system and if possible also replace older technologies or solutions to be able to reduce the number of technologies/solutions that the organization need to support.

- C3: How well does TSN today match the requirements of long life-cycles?
- C4: What are the challenges in integrating and using TSN in a legacy system?

3 Background and related work

3.1 Embedded systems

Embedded systems, defined as *"... a combination of computer hardware and software – and perhaps additional parts, either mechanical or electronic – designed to perform a dedicated function."*, [1], exist in several different installations around the world. Some really small, like a temperature sensor with a display, and other large complex systems, like cars, trains, airplanes, and energy systems. The complexity of the system depends on several factors, e.g., the number of use-cases, throughput, latency, parallelism, environment, area distribution, security, safety requirements, and the expected life-cycle that can range from one-time use, e.g., cargo tags to systems with a expected long-life time, e.g. up to 40-50 years for an energy transmission system.

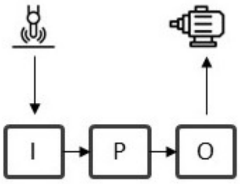

Figure 6: Generic embedded system consisting of input, processing and output.

In simplified view, all embedded systems follows the same model, 6: input, process, and output. The *input* could e.g., be sensor readings measuring the rotation speed of a car wheel and the amount of applied break pressure. The input is followed by*processing* the measured values to make a decision if the Anti lock Break System (ABS) should interact on one or more wheels. Finally the result from processing will result in an action via the *output* hydraulic break system. When the embedded systems starts to interact with its surrounding world, e.g., with the hydraulic break system, a Cyber-Physical System (CPS) is created. The CPS is then a higher coordinating system between the physical system and the computational elements of the embedded system.

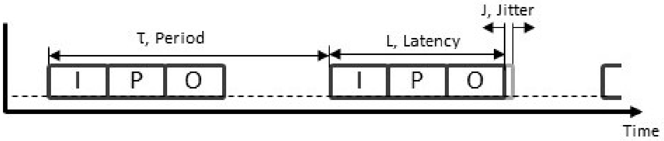

Figure 7: Scheduling of the three tasks, I,P and O with a period, latency and jitter.

Since we now have a system that need to react within a specific time, from input to output, we need to place real-time constraints on the functionality, e.g., task period, latency, and jitter, as displayed in Figure 7, and by that creating a real-time system, defined as *"Real-time systems are computing systems that must react within precise time constraints to events in the environment. As a consequence, the correct behavior of these systems depends not only on the value of the computation but also on the time at which the results are produced."* [27]. When building an maintaining real-time systems the real-time constraints need to be meet in all situations in order not to risk any malfunction and by that risk of damage on material assets, injuries, or loss of life.

The system in Figure 6 is simplified since the real-world system would not be independent to other systems in the car but rather be one part of a complete system in the car and by that interacting and creating dependencies to other systems, e.g., anti-skid system or damping control, as in Figure 8.

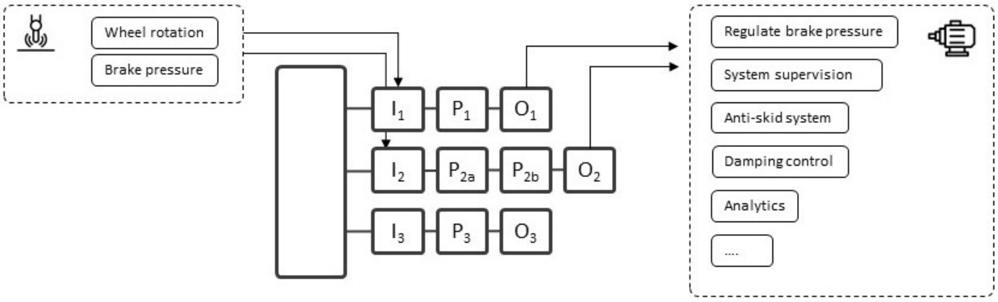

Figure 8: Simplified overview of an ABS system built around a scheduler.

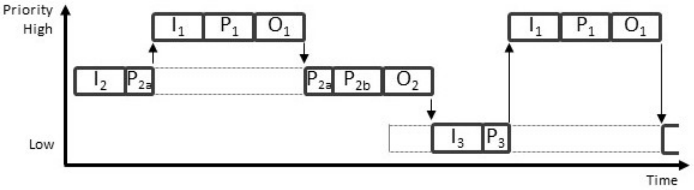

Figure 9: An example of an preemptive scheduling of the task from Figure 8 with three different priority levels.

The initial simplified embedded system has now become more complex and larger in size with interfaces to a number of different functions scheduled on the same CPU. In Figure 9 an example is presented to show a bit of the complexity when different tasks are scheduled with three different priority levels on the same CPU. Since only one task can run at a time, the tasks on level 2, e.g., P_{A2} are interrupted by the highest priority tasks, e.g., I_1, and by that an increased latency until data is sent to the outputs. Depending on how often this happens, depending on the interrupt frequency, the outputs, O_2, will experience jitter. More effort is needed in this case to make sure that all the real-time constraints are meet [25].

Another example of a system from the car industry is an Engine Management System (EMS), that in a fairly modern car that connects to more than 200 different types of sensors (inputs) and actuators (outputs) and consisting of half a million lines of code distributed over 2000 functional modules and 500 different source code files [5]. In a high end car you would find more than a hundred Electronic Control Units (ECUs) [2] [17] that are interacting with each other, creating a large and complex system. This complexity is making the system evolution more difficult when it comes to e.g. replacing a component, since it is not only the ABS unit that need to be retested but rather also all other dependent ECUs.

3.2 Availability, life-cycle and technology evolution

When discussing embedded systems in industrial applications they are usually part of a larger scope, i.e., a System of Systems (SoS) [10], where functional requirements, e.g., accuracy, availability, measurement range, communication protocol, and spare parts availability, are set by the next level in the system, e.g., the requirements for a merging unit in a substation environment is set by the protection function in the IED in combination with the transducer that is interfaced. The 24/7 availability of the system is now also relying on the availability of the merging unit, where different redundancy concepts can be used to improve the availability of a single component, e.g., dual systems, 2 out of 3, or 3/2 breaker scheme. Moreover, also the system life-cycle depend on the unit, i.e., the complete substation has an expected availability of 40-50 years and by that setting requirements on the components and system that handle it. Embedded system hardware components will in most cases not be available during 40 years so different strategies, [22]

[18] [13] [14] [28] , is needed to be able to replace obsolete parts of the system during the life-time. An example of an strategy from the HVDC and FACTS transmission and distribution business, agreed on by manufacturers and customers, is that the embedded control system, e.g., controllers and IO, should be upgraded approximately every 15:th year and the Human Machine Interface (HMI), servers, routers, and switches, every 8 years [4]. Handling obsolete hardware components is one reason to upgrade a system, another is obsolete software, e.g., outdated windows operating system versions or compilers. Different regulations may change during the system life-time, e.g., Restriction of Hazardous Substances Directive (RoHS)[24], safety requirements or machine directives.

The technology that we are using is also evolving and new solutions, with better performance and new possibilities, are available for implementation into future generations of our systems or as an update or replacement of a specific module that is in use today. An example of such a technology evolution is from the car industry that, with hundreds of ECUs per car, are moving to more centralized solutions, [6] [20] [21] by consolidating functionality to reduce the number of hardware units, using for example multi-core solutions. Still the initial real-time requirements need to be met,e.g. latency and jitter, and by that the complexity per unit increases. This in turn require effort into keeping different functional blocks independent [16] [15] to reduce development and verification efforts when blocks are updated. Similar technology steps, as with multi-core, can also been seen within other solutions, e.g., system-on-a-chip, where more that one type of functions is included in the same chip, e.g. CPUs, communication, and AI-engine. Other examples of new technologies that gives possibilities for new functionalities are Time Sensitive Networks (TSN), increased network bandwidth, cloud computing, and different analytic functions.

3.3 A new paradigm

The introduction of new technologies has also enabled an paradigm shift, i.e., a fundamental change in a basic concept, in almost all types of industries. An example is the "connected world" and/or using of data, from small sets of data to "big-data", with the aim to improve the system performance or create new functionalities in higher level systems. A similar set up as with the SoS described earlier in the car ECU example, when functionalities uses resources and capabilities from other subsystems to create a new system which offers extended functionality that differs from the sum of the subsystems, can be seen in ISO/IEC/IEEE 21839 [11]. Several larger initiatives are today pushing in this direction, e.g., Industry 4.0 [12] and Smart Grids [32], with intention to increase flexibility in manufacturing, reduce environmental impact, increase revenue, capture new business opportunities, or introduce new business models.

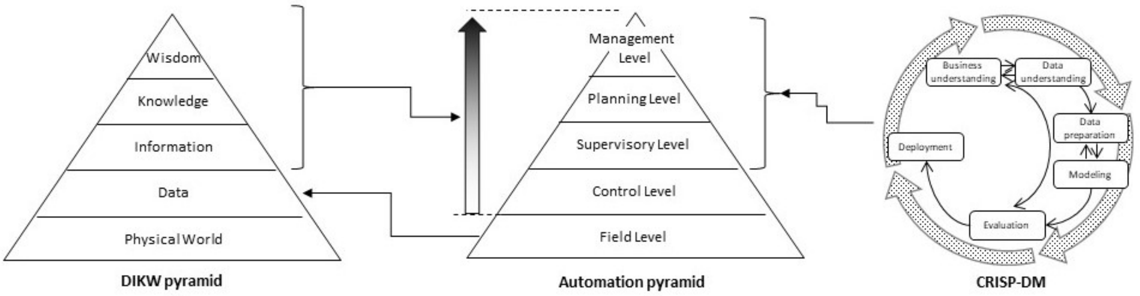

Figure 10: Left: DIKW pyramid, Middle: Automation Pyramid, Right: CRISP-DM process for data mining.

A key enabler in future systems will be to make "smart" decisions based on data, e.g., sensor data from the CPS systems. The relation between the wisdom needed to take the "smart" decisions and data can be described in a DIKW pyramid, Figure 10, (*D*ata, *I*nformation, *K*nowledge, and *W*isdom) where the bottom layer is the "real world" sampled by the sensors in the second level, generating "data". "Information" is generated by extracting "useful" information from the data, which then can be used for making decisions and taking actions. From this new information organizations can gain knowledge, next level, by organizing sets of information and finally create "wisdom" by appling the knowledge in the organization. The DIKW concept could for example be compared to the automation pyramid referenced in Industry 4.0 where the field level would contain the sensors and each level above that would improve the level of knowledge used to make decisions by combining data and refining the content through different data analytical methods. An example of a widely used data science framework is the *CR*oss *I*ndustry *S*tandard *P*rocess for *D*ata *M*ining (CRISP-DM) [31], including six phases (business understanding, data understanding, data preparation, modeling, evaluation and deployment), used in a iterating fashion to understand and improve the used data/information to create a better result.

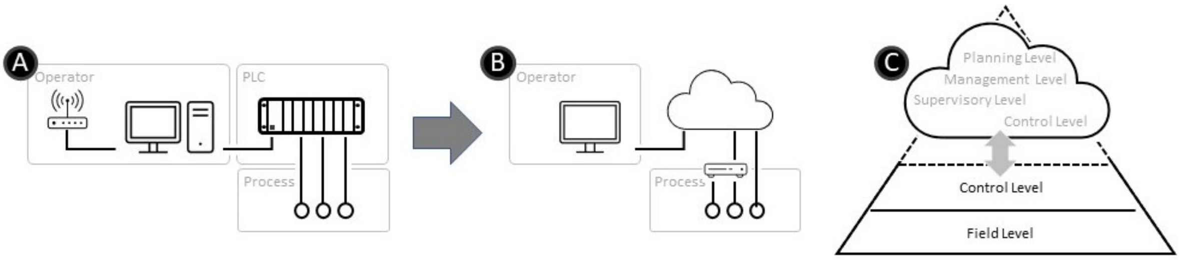

Figure 11: A: "normal" PLC installation at site, B: enhanced with cloud/real-time cloud, C: Example of automation pyramid with part of the functionality in an cloud instance.

One enabler for building solutions based on a extensive usage of data/information from a large set of sources is cloud computing. Cloud computing has emerged as a new computing paradigm, giving a flexible and dynamic infrastructure to handle compute and storage services on demand. A example of a use case, within Industry 4.0, is to enable working with smart objects, autonomous products, and decision making processes [30] Cloud technologies could in the future handle more and more of the tasks that are today located in the embedded systems layer, e.g., as a real-time cloud , and by that offload the embedded system and at the same time give more flexibility, Figure 11. Adapting the current cloud environments to be used in e.g., an industrial installation still has work left to be done, e.g., new approaches for vendor independence to ensure interoperability, and consideration of security issues [7].

Figure 12: Example of devices connected and supported by a Smart Grid.

The Smart Grid initiative, defined in the European Union as *"Smart grids are energy networks that can automatically monitor energy flows and adjust to changes in energy supply and demand accordingly. When coupled with smart metering systems, smart grids reach consumers and suppliers by providing information on real-time consumption"*, [29], i.e., a Smart Grid is a power system embedded with an information layer that allows improving the way resources are used, Figure 12, by using the measurement data available in the system, business data, and also external data [34]. Most of the operation data are available through already installed sensors and could be used to e.g., improve fault detection, transient stability analysis, electric device state estimation/health monitoring, and power quality monitoring [33].

3.4 Summary

Going forward, capturing the benefits of using data to enable new functionality will be required due to the large possible gains, e.g., environmental resource usage, social welfare, and business benefits. This will push us to enable the gathering of data from already deployed systems or systems that are under development and to be put in to service in the future. Enabling this data collection and distribution, in existing and future systems, will introduce several challenges, e.g., update and scheduling of embedded systems with new features running in parallel to legacy functionality, new cyber security threats, privacy protection, communication, data storage, and integration between experts (e.g., power system engineers, embedded specialists, data engineers, business analytic). The cost for meeting these challenges will be high but the gain will potentially be much larger.

4 Research methodology

In this thesis we have used the overall research process as defined in Section 4.1. Each research contribution has been generated using one or more of the research methods presented in Section 4.2. The mapping of scientific papers and corresponding research methods is presented in Table 2. Finally, the validity of the results presented in this thesis is presented in Section 4.3.

4.1 Research process

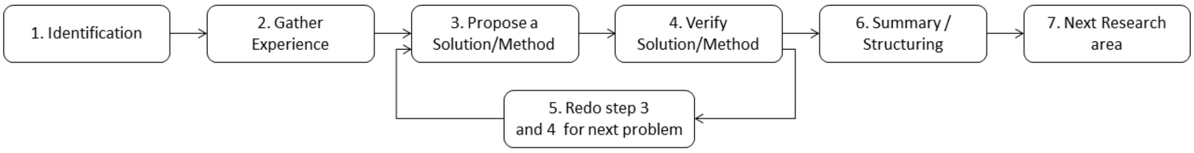

Figure 13: Research process described in 4 steps, where step 3 and 4 are repeated for each identified problem.

1. Identification of the research problem based on current trends in the real-time and embedded systems communities together with the defined research goal.
2. Gather experiences from industry and academia related to the described research goals by conducting workshops, discussions and study literature for related work.
3. Propose a solution/method to one of the problems that was defined in step 2.
4. Verification and validation of the research results by performing case-studies or other means of verification.
5. Redo step 3 to 4 for the next identified problem(s) or refinement of the current problem.

6. When all solutions/methodes has been verified the results need to be summarized and structured.
7. Move to next research area and redo the process.

4.2 Research methods

From [26] we have three initial questions connected to scientific and engineering research: *"What kind of questions are "interesting"?"*, secondly *" What kinds of results help to answer these questions, and what research methods can produce these results?"*. The last part is about the *"evidence that can demonstrate the validity of a result, and how are good results distinguished from bad ones?"*. Combined with the six typical phases that a technology goes through from basic research and concept to popularization, typically it takes 15 to 20 year, [23], Figure 14.

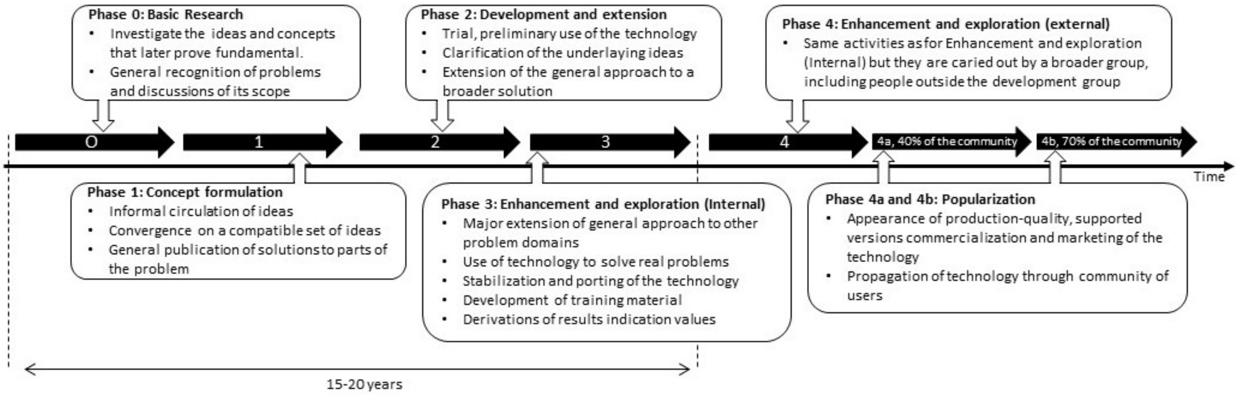

Figure 14: 6 stages of technology development, from concept to popularization.

Adding then a life-time of 30-40 years to the system gives long life-cycle where different types of scientific methods is needed depending on the stage or stages that the the current technology and system is in, since a system typically is a combination of different mature technologies.[3] propose three nested classes of results: *Findings* that are well-established scientific truths, *Observations* containing reports on actual phenomena, and *Rules of thumb* that is generalizations, signed by their authors and perhaps incompletely supported by data, but judged by usefulness. Observations and Rules-of-thumb provides valuable information and guidance in situations when findings are not available, or as a groundwork for an area where future research later on will give findings. [26] generally software engineering research is motivated by practical problems, e.g., implementation of new technology in legacy systems to allow for orthogonal functions, and key objectives are often quality, cost and timelines.

Table 1: Mapping of the used methods to the different papers.

	P_A	P_B	P_C	P_D	P_E	P_F
M_1	✓	-	-	-	-	✓
M_2	-	-	-	✓	-	-
M_3	-	✓	-	-	✓	-
M_4	-	-	✓	-	-	-

4.2.1 Method 1 - Exploratory study

For the exploratory studies we have divided the work into three steps: i) based on our experience we have formulated the current situation and the challenges that we meet, ii) validation of the current situation has been done by senior specialists with domain knowledge, and iii) through criteria that is important for the domain in which we have identified issues and possible research directions.

Regarding step i) we have used both primary and secondary research methods. The primary research has been conducted through workshops and in-depth discussions with a limited set of specialists. Secondary research methods have been conducted when it comes to finding related work in the existing literature.

Step ii) has been performed via reviews of reports and conclusions by relevant company experts and technical program committees of scientific conferences.

Step iii) has been performed via describing vital criteria for industry. Based on these criteria, challenges have been selected.

4.2.2 Method 2 - Formal verification

For the formal verification we have done in this research we have built models of the proposed systems and used the UPPAL model checker for verification of the models. The system that we want to verify has been modeled as networks of times automata. Represented by finite state machines that are extended with e.g., clocks and data types. The models are built up using a formal query language to specify properties that we want to check. Once the model is built we use it to check different system behaviors, e.g., can the synchronization mechanisms be blocked by any other functions in the system.

4.2.3 Method 3 - Relative comparison

Initially we have identified a challenge and proposed a number of solutions and need to compare them to each other. To keep the discussion on a general level, valid for more that one specific use-case, e.g., result is dependent too much on details in a specific hardware platform, the comparison is done in a more relevant manner on a higher level allowing for the reader to prioritize what aspects that is most important for their specific use-case and system. The conclusions has been presented by summarizing the results in a table or by presenting Pugh scores [19].

4.2.4 Method 4 - Standard readiness evaluation

As a first step we have identified a general problem, e.g., life-cycle, that need to be handled before use of a technology based on a new standard in an embedded system. From this we have derived a number of challenges and evaluated the standard according to them. Challenges including the completeness, e.g., are all standards finalized and adopted by large organizations and manufacturers, is there any openings in the standard that can lead to future problems, e.g., significant parts still left out of the standard, and aspects when it comes to implementation in an embedded system, e.g., configuration. At the end we summarized the result and allowed for the reader, dependent on the use-case, make a judgment on what aspects that is important in their filed of usage, e.g., the readiness of compliant to automotive standards, or availability of products on the open market.

4.3 Validity

A common focus for the research conducted in this thesis has been to keep the research on a generic level and for larger complex systems with an expected long life-cycle, with the goal of ensuring that the collected data can be used to find answers to the research questions. The challenges described in this thesis, even if they are mainly connected to systems with a long expected life-cycle, are in most cases also valid for other types of systems, e.g., consumer products as headsets, battery chargers etc. Keeping the question on a more generic level has the advantage of a broader scope that can be used to guide future research needed or to guide product decisions. If our resources would have been "unlimited" then of course a more detailed investigation in each sub area, e.g. effect of cache behavior for a specific ARM architecture on the challenge of adding analytic to a parallel CPU, would have given more exact answers, but on the other hand the architectures of today are changing rapidly, e.g., by introducing System on chips, and there are already to many CPU alternative to investigate all of them, e.g., ARM Axx/Rxx/Mxx, different versions of X86, RISC-V. The end user will have to make that investigation, since the choice of CPU architecture for a product is not only dependent on the cache behavior but rather also on, e.g., price, tool chain, design knowledge available in the organization, component availability and life-cycle. The last two are getting more and more important in the current situation in the world with the ongoing component shortage forcing companies to rather use, maybe not the most optimal design, but the one that allows to reach the product delivery goals. Time to market is critical when the income to the company are based on selling and paid on delivery.

The data collection has been made trough workshops and discussions involving the authors along with industry experts from different areas, e.g., transmission and distribution systems or automation systems. The subjects has been in focus for the industrial partner organization due to the current needs from the market, giving a push to support the discussions, and also an product development ongoing in parallel to the research project. The validity has by that been further improved by involving senior experts as reviewers and discussion partners. Several of the use-cases, e.g., Cyber Security, information gathering, are general and not in detail connected to specifically one type of industry and by that making the results valid in a broader scoop, but the investigations could have had a more product oriented focus, i.e., replace the complete product, if not focus would also have been on a long life-cycle, as needed by the involved industry partners. There is always a risk that the chosen focus, e.g., orthogonal functions, that is in focus of the current business, e.g., Smart Grid, Industry 4.0, vs other possible open questions within research community that to some other industrial users may have a higher value.

To keep the evaluations, when different implementation alternatives are possible, as independent as possible from specific hardware or software solutions we focused on using relative comparisons, e.g., Pugh score, vs. using more detailed investigations, e.g., based on timing measurements, that could have been done by investigating one or several different experiment setups. Experimental setups would give a more detailed view for the specific case or platform vs. the more relative approach, but also be limited to that specific platform. As a future step experimental setups could be used to explain the effect of some of the corner cases, e.g., internal communication performance in an SOC, or be used to emphasis/verify the high level results. Only relying on practical measurements without any high level reasoning would give a very narrow result if not a large and costly effort is put in the the work.

Similar approach was taken for the evaluation of the TSN standard vs. the subject of long life-cycle. Evaluation specific functionality in a standard, e.g., time synchronization, can be done in experimental setups, with the risk of being not generalized, instead we focused on general topics, e.g., if the standard is finalized an adopted, any openings that could change in future, based on our review of the standard from general life-cycle perspective and the knowledge the team has built up when using the standard in different research projects. There are several other options of emerging technologies, like TSN, coming to our future possible usage and the reason to choice TSN in our case is that network technologies will play an important role in future real time systems and it looks like TSN will be one of the key functions to enable the sharing of networks between different priority levels and also with accurate timing needed in a hard real-time system. The validity in the area is also increased by the high focus in the research group, at Mälardalens University, on the TSN standard.

For part of the technology validation we used formal models, giving us an independent way of verifying a solution, but at the same time we are also dependent on the validity of the model that we built for the verification. Here we can continue to work with improving the model and making it more useful for continued verification of other aspects in the same area. The formal verification approach was also a way to get around the ongoing pandemic that did not enable us, in a easy manner, to get access to the needed equipment.

5 Thesis contributions and included papers

5.1 Contributions

The scientific contribution of this thesis is composed of Paper A to F as follows.

Table 2: Mapping the thesis contributions to the challenges.

	P_A	P_B	P_C	P_D	P_E	P_F
C_1	✓	-	-	-	-	✓
C_2	-	✓	-	-	-	✓
C_3	-	-	✓	-	-	-
C_4	-	-	-	✓	✓	-

The contribution for the C_0 challenge was presented in the licentiate thesis.

In Paper A we introduces the concept of ”dual use” for a control system, i.e., not only controlling and/or protecting system but instead also handling orthogonal tasks, e.g., data gathering for higher levels of systems. We present three challenges related to i) information gathering, ii) life-cycle management and iii) data governance. For the challenges we propose directions for solutions that need to be evaluated already at design time.

In Paper B we define a reference embedded system, a single core system, with the intentions of comparing advantages and disadvantages when introducing ”dual usage”. With help of the reference system we compare three different alternatives solutions, a) a multi-core system where we are using a separate core for analytics, b) using a separate analytics CPU and c) analytics functionality located in a separate subsystem.

In Paper C we have evaluated the current status of the Time Sensitive Networking (TSN) standard with respect to how well it can be used in a complex embedded system with an expected long life-cycle. We identified challenges with in five different areas and proposed mitigation and opportunities.

In Paper D we evaluate and identifies challenges with integrating a legacy EtherCAT network into a TSN network. We propose a clock synchronization method based on TSN as a synchronization method for the EtherCAT network and verify the performance with formal verification.

In Paper E we evaluate the trade off between performance vs. latency and robustness for different communication solutions on a real-time Ethernet sensor network.

In Paper F we present cyber security challenges, and suggest solutions, with complex embedded systems within the area of power systems connected to long life cycles. Cyber security has been and will in the future be even more important aspect of embedded systems of all different kinds and we have in this paper presented challenges and standards connected to Power systems but most are generic and also valid for a generic use case.

5.2 Included papers

Our contributions are proposed in form of published papers. Currently, the solutions are represented by six papers, namely A, B, C, D, E and F. A - E are published at conferences and F is accepted at a conference.

5.2.1 Paper A

Title: Challenges in providing sustainable analytics of systems of systems with long lifetime.
Authors: Daniel Hallmans, Thomas Nolte, Kristian Sandström, and Stig Larsson.
Status: Published in SoSE 2021.
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.

Paper A contributions:

- We introduce the concept of ”Dual Usage” for a control system.
- We present three central challenges related to i) information gathering, ii) life-cycle management, and iii) data governance.
- We propose directions for solutions to these challenges that need to be evaluated already at design time.

My role: I was the main driver of the work under supervision of the co-authors. The initial work is based on my experiences from working in industry and being confronted by the need to implement a ”Dual purpose use” in an already existing control system.

5.2.2 Paper B

Title: Design considerations introducing analytics as a“dual use” in complex industrial embedded systems.
Authors: Daniel Hallmans, Thomas Nolte, Kristian Sandström, Stig Larsson, and Niclas Ericsson.
Status: Published in ETFA 2021.
Abstract: Embedded systems are today often self-sufficient with limited and predefined communication. However, this traditional view of embedded systems is changing through advancements in technologies such as, communication, cloud technologies, and advanced analytics including machine learning. These advancements have increased the benefits of building Systems of Systems (SoS) that can provide a functionality with unique capabilities that none of the included subsystems can accomplish separately. By this gain of

functionality the embedded system is evolving towards a "dual use" purpose. The use is dual in the sense that the system still needs to handle its original task, e.g., control and protect of an asset, and it must provide information for creating the SoS. Larger installations, e.g., industry plants, power systems and generation, have in most cases a long expected life-cycle, some up to 30-40 years without significant updates, compared to analytical functions that evolve and change much faster, i.e., requiring new types of data sets from the subsystems, not know at its first deployment. This difference in development cycles calls for new solutions supporting updates related to new requirements inherent in analytical functions. In this paper, within the context of "dual usage" of systems and subsystems, we analyze the impact on an embedded system, new or legacy, when it is required to provide analytic data with high quality. We compare a reference system, implementing all functions in one CPU core, to three other alternative solutions: a) a multi-core system where we are using a separate core for analytics, b) using a separate analytics CPU and c) analytics functionality located in a separate subsystem. Our conclusion is that the choice of analytics information collection method should to be based on intended usage, along with resulting complexity and cost of updates compared to hardware cost.

Paper B contributions:

- We define a reference Embedded system system with the intentions of "dual usage".
- We compare the reference system, which implements all functions in one CPU core, to three other alternative solutions: a) a multi-core system where we are using a separate core for analytics, b) using a separate analytics CPU and c) analytics functionality located in a separate subsystem.
- Generated an conclusion based on the intended usage.

My role: I was the main driver of the work under supervision of the co-authors. The initial work is based on my experiences working with similar questions industrial embedded systems.

5.2.3 Paper C

Title: Analysis of the TSN standards for utilization in long-life industrial distributed control systems.
Authors: Daniel Hallmans, Thomas Nolte, and Mohammad Ashjaei.
Status: Published in ETFA 2020.
Abstract: Large complex industrial Distributed Control Systems (DCS), e.g., power distribution systems, are expected to function for long time, up to 40 years. Therefore, besides having a long system verification phase for all subsystems, the design phase should consider various aspects when it comes to selection of which technologies to utilize when implementing such systems. In this paper, we study and investigate key challenges of using the Time Sensitive Networking (TSN) technology when it comes to design, maintenance and evolution of long life-span complex DCS. We also identify issues and challenges, and propose mitigation strategies for using the TSN technology in long-life system design. Our investigation and analysis shows that many of the TSN standards are in their evolution phase and may as a consequence be subject to different interpretations and implementations. Therefore, achieving a full capacity of using the TSN technology may not be possible, in particular when it comes to design of systems having an expected long life.

Paper C contributions:

- We evaluated the status of current TSN standards.
- Identified five challenges when using TSN in a system with long expected life time.
- Identified opportunities.

My role: I was the main driver of the work under supervision of the co-authors. Analysis part shared with Mohammad.

5.2.4 Paper D

Title: Clock synchronization in integrated TSN-EtherCAT networks.
Authors: Daniel Hallmans, Thomas Nolte, Mohammad Ashjaei, Daniel Bujosa Mateu, Alessandro V. Papadopoulos, and Julian Proenzaz.
Status: Published in ETFA 2020.
Abstract: Moving towards new technologies, such as Time Sensitive Networking (TSN), in industries should be gradual with a proper integration process instead of replacing the existing ones to make it beneficial in terms of cost and performance. Within this context, this paper identifies the challenges of integrating a legacy EtherCAT network, as a commonly used technology in the automation domain, into a TSN network. We show that clock synchronization plays an essential role when it comes to EtherCAT-TSN network integration with important requirements. We propose a clock synchronization mechanism based on the TSN standards to obtain a precise synchronization among EtherCAT nodes, resulting to an efficient data transmission. Based on a formal verification framework using UPPAAL tool we show that the integrated EtherCAT-TSN network with the proposed clock synchronization mechanism achieves at least 3 times higher synchronization precision compared to not using any synchronization.

Paper D contributions:

- We propose a clock synchronization mechanism based on the TSN standards to obtain a precise synchronization among EtherCAT nodes.
- formal verification with UPPAAL.

My role: Together with Daniel Bujosa Mateu, we were the main drivers of the work under supervision of the co-authors. Formal verification was done by Daniel Bujosa Mateu.

5.2.5 Paper E

Title: Consistent sensor values on a real-time Ethernet network.
Authors: Daniel Hallmans, Thomas Nolte, Kristian Sandström, and Stig Larsson
Status: Published in INDIN.
Abstract: In industrial control systems there is often a need for short latencies and/or consistent data gathering. In a system with limited resources it is a challenge to achieve the combination of short latencies and consistent data. In this paper we propose three different architectural solutions to this challenge, each having different trade-offs: one that gives a consistent set of data and also a short latency but with a higher resource usage, a second alternative that reduces resource needs but at the cost of an increased latency, and a third and final solution that reduces resource needs to a minimum however in doing so also increasing the latency. The results presented in this paper suggest that it is possible to get low latency and robustness at the cost of performance.

Paper E contributions:

- Evaluation of trade of between performance vs. low latency and robustness for a real-time Ethernet sensor network.

My role: I was the main driver of the work under supervision of the co-authors. The initial work is based on my experiences working with similar questions in industrial embedded systems.

5.2.6 Paper F

Title: Identified challenges and opportunities with cyber security standard compliance in combination with a long-expected lifetime.
Authors: Daniel Hallmans, Johan Malmström, and Jeremy Morgan.
Status: Accepted at Cigree 2022.

Abstract: Transmission and distribution systems, e.g., HVAC systems, HVDC or FACTS, has an expected operation lifetime of up to 40 years, – a time span that equals or is even longer than the average professional lifetime of an engineer – and during this time many things will change. During the 40-year lifetime the included embedded system will need to be maintained, updated and replaced several times. Cyber Security is one of the critical parts the builds up the complete systems and by that also need to follow its life-cycle.

To analyse the impact of changes over time, we assume a large transmission system to serve as an example, with a structure following the Cigré Technical brochure 603 put into service in 2015. The devices in the assumed system adhere to the then recently published standard for IED cybersecurity, IEEE 1686: 2013 – IEEE Standard for Intelligent Electronic Devices Cyber Security Capabilities.

In the decade that follows, the threat landscape will change as new threat vectors and new technology advancements appear. Standards will also change, and additionally new ones will be published to stay one step ahead of potential interference by cybercriminals. All future changes cannot be anticipated during the system's design phase, e.g., due to new use-cases, new generations of technologies, so the system needs to be adopted.

In this paper we describe a set of key cyber security challenges that we see today and going forward when maintaining an embedded system used in a transmission and distribution system during a long operation lifetime. We also give some suggestions going forward and a direction for future work.

Paper F contributions:

- Presenting challenges and opportunities with complex embedded systems within the area of power systems connected to long life cycles with focus on cyber security.
- Summary of relevant cyber security standards for transmission and distribution systems.

My role: One of the main drivers for the paper and my main contributions where around technology and orthogonal challenges and summary.

6 Conclusion and future work

6.1 Summary and conclusion

In this thesis we have investigated and proposed solutions for handling orthogonal functionalities in complex embedded systems with long expected life-cycles, up to 30-40 years for e.g., energy transmission systems or process industry installations. We have defined the orthogonal functionality as a functionality that does not directly contribute to the main task of the embedded system, e.g., a closed loop control that is used to control and protect a local installation, but still uses information or is deployed on the embedded system to achieve other goals, e.g., gathering of data for analytic or cyber security functionality. In systems with an expected long life-cycle the requirements for orthogonal functions will vary over time and in most cases it is not possible to anticipate or fully take height for such unknown functionality already at the system's first design time, e.g., 10-15 years earlier. Instead these systems need to be updated with new orthogonal functionality continuously during the system's life time.

Real time systems are located all around us and they are systems that need to react within precise time constraints connected to events that occurs in the physical environment around the system, i.e., not only depending on calculating a value but also on the time when the value is produced. Timing properties as task period, Latency and Jitter becomes important constraints. Based on the real time requirements the complexity increases when the system grows, e.g., more dependencies are created between different functions and subsystems and by that design and verification becomes more complex, and with increasing life-cycle, e.g., replacement of obsolete components, the effort to maintain and update becomes larger. A larger installation it can take 3-5 years just to agree on a specification, tender and negotiate the work before the engineering, procurement and construction phases start. Stages that also continues for several years, e.g., 3-5, until the installed and commissioning starts on a site. From the initial ideas of building the system already up to 10 years has already passed, when the systems becomes operational and its life-cycle of 30-40 years starts.

Nothing stays as is, so during the systems operational time new requirements will emerge and some of them is orthogonal to the initial functionality of the system, but still vital for the system owner, e.g., collection of run-time data used for analytics that at the end could lead to, e.g., reduced usage of environmental resources, improved social welfare, or business benefits. This will push the owner to enable the gathering of data from already deployed systems or systems that are under development. By doing this, several challenges will arise, e.g., update the scheduling of task in the embedded systems with new features in parallel to the legacy functionality, opening for new types of cyber security threats that need to be mitigated, privacy protection, new communication paths, new or larger data storage, needed synergies between experts (e.g., power system engineers, embedded specialists, data engineers, business analytic).

In this thesis we have introduced a concept of "dual use" i.e., orthogonal functionality and explored a set of challenges that arises when such functionalities are introduced in complex systems with long life-cycles. We have evaluated technologies and architectures that could simplify the implementation of future orthogonal functionalities. The initial challenge, C0, is connected to the identifying a relevant subset of challenges that an Embedded system with an expected long life-cycle is typically facing. Reason for investigating the challenge is to get an understanding of the "base load" that an organizations need to handle when it comes to systems with long life-cycles, e.g., component obsolescence, knowledge transfer in a organization [8].

On top of the "base load" we are now adding new challenges created by adding, or preparing for, orthogonal system functions, challenge C1, such as gathering data for analytics or adding cyber security solutions on different levels in new and also systems that are already deployed on the market systems and has been in use for several years.

In systems used to day, or planned, it will be hard to avoid the future functional growth and by that also there will be new challenges in, e.g., information gathering, life-cycle managements and data governance. Effort is needed already at design time to try to mitigate the future need, even if the need is not fully known, by e.g. be able to implement functionality that is not disturbing the core C&P functionality and allow it to have its own life-cycle. We can simplify the implementation of orthogonal functionality, c2, by making architectural during design time that gives future possibilities, e.g., having analytic on a separate core or having even on a separate hardware with connectivity to the source. Each proposed solution also have negative effect when it comes to e.g. cost, effect on C&P, possibility to update already available systems, making it a trade off between different parameters. There is not one ”silver bullet” since every use-case will have its separate answer, dependent on embedded system design, organizational priorities, time, cost, etc.

New technologies solutions has always played a important role in building efficient embedded systems, that e.g. can allow for orthogonal functional growth, and continue to be so also in the future. When an emerging technology is entering the market, early in the Hype curve as a technology trigger, an important question is to understand if this is a technology that will be available for the future or or if it will not mature to a stable technology that can be used during a product complete life-cycle. We see that Time Sensitive Network (TSN) is a technology that could enable a lot of benefits for the future products, e.g. network prioritization to allow consolidating of network traffic, but at the same time we can point at several challenges that are not yet handled, e.g. standard is not fully finalized for all areas. Building large complex systems also usually means that a new technology need to co exist with several of the legacy technologies already available in the system, e.g., EtherCAT over TSN. When it comes to EtherCAT and TSN we are able to show that TSN clock synchronization could be used within an EtherCAT network and by that integrating it in to the TSN network.

We can conclude that the gain in implementing orthogonal functionalities, e.g., gathering of analytics data, are in some cases so large, e.g., environmental resource usage, social welfare, business benefits, and is today supported by several large initiatives as Smart-Grids and Industry 4.0 that we need to implement it into new and also excising systems. Enabling the functionality will contain several challenges, e.g., update and scheduling of embedded systems with new features in parallel to legacy functionality, new cyber security threats, privacy protection, communication, data storage, synergies between experts (e.g., power system engineers, embedded specialists, data engineers, business analytic) will be high but the gain will typically be much larger. Strategies to handle the updates is needed already at design time even if all future requirements are not known.

6.2 Future work

Since we can conclude that requirements and technology will change going forward, a future important research subject will be how to create functionality that are independent to each other and also from the used technology, e.g., C&P functionality side by side with cyber security or data gathering, that share same resources, access to same information, sharing same communication interface, but at the same time are independent in such a way that they can be designed, tested and updated separately. If it would be possible several of the issues of maintaining a system over a long time period or adding new functions to an existing system would much easier, e.g., replacing obsolete hardware with a new generation, adding new functions several years later on. Some of the new technologies that are entering the market today, e.g., systems on a chip, many core solutions, are in some cases not making this easier since then more functions are sharing the same hardware and by that will interfere with each other, for example in the case of many core where typically the memory bus is common for all cores, and the hardware set up in a system of a chip solution will probably not look the same as in your legacy system and by that give new timing behavior between modules. Since new technologies will not be the same as the technologies that we have today, a ”re-produce-ability” can not strictly be based on replication same type of hardware behavior but rather setting constrains on the timing parameters, e.g., period, latency, jitter, for a specific task.

Since we concluded that new technologies will be important we will also continue to monitoring evolving technologies and their possible impact on new solutions from a system performance view and also from a system life-cycle aspect. Here we also need to continue to identify important parameters that can indicate for how long a specific technology will be available.

7 Planning

7.1 Time plan

The time plan for the remaining activities is to have the PhD proposal in November 2022 and then to complete the PhD during end of Q2 or early Q3 2023.

7.2 Thesis outline

The format of the licentiate thesis will be a collection of papers. The following sections are planned to be included in the thesis:

Part I

1. Introduction and motivation
2. Research methods and goals
3. Background and related work
4. Included papers and thesis contributions
5. Conclusions and future work

Part II

1. Paper A
2. Paper B
3. Paper C
4. Paper D
5. Paper E
6. Paper F

7.3 Progress

7.3.1 Publications

- Paper A: Published
- Paper B: Published
- Paper C: Published
- Paper D: Published
- Paper E: Published
- Paper F: Accepted

7.3.2 Courses

The required amount of credits towards a PhD degree is 75. As shown in Table 3, all has been completed.

Table 3: Overview of courses towards PhD degree.

Course name	ECTS	Status
Research Planning	4.5	Completed
Parallel computing, theory, hardware, software - special focus on multi-core	6.0	Completed
Real-time systems II	4.5	Completed
Real time communication	6	Completed
Research methods in natural science and engineering	7.5	Completed
Professional ethics	7.5	Completed
Empirical research methods	7.5	Completed
Systems Thinking and its Application in Embedded Systems	7.5	Completed
Industrial Systems Cloud Computing	7.5	Completed
Wise workshop	1.5	Completed
Technology Transfer	4.5	Completed
AI	3	Completed
TSN for Industry	7.5	Completed
Models and Methods to Manage Complex Systems	7.5	Completed
Total	75	

References

[1] M. Barr and A. Massa. *Programming embedded systems: with C and GNU development tools.* " O'Reilly Media, Inc.", 2006.

[2] R. Berger. Roland berger study explores need for consolidation in vehicle electronic architectures. `https://www.greencarcongress.com/2015/07/20150729-berger.html`, 2015 (accessed August 10, 2022).

[3] F. P. Brooks. Grasping reality through illusion—interactive graphics serving science. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1–11, 1988.

[4] W. B. Cigre. Cyber asset management for hvdc and facts, 2021.

[5] D. Claraz, S. Kuntz, U. Margull, M. Niemetz, and G. Wirrer. Deterministic execution sequence in component based multi-contributor powertrain control systems. In *Embedded Real Time Software and Systems (ERTS2012)*, 2012.

[6] M. Di Natale and A. L. Sangiovanni-Vincentelli. Moving from federated to integrated architectures in automotive: The role of standards, methods and tools. *Proceedings of the IEEE*, 98(4):603–620, 2010.

[7] O. Givehchi, H. Trsek, and J. Jasperneite. Cloud computing for industrial automation systems—a comprehensive overview. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2013.

[8] D. Hallmans. Evolutionary aspects of complex embedded systems with long life cycles, 2018.

[9] 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*, pages 384–389. IEEE, 2014.

[10] I. ISO, IEC. Iso/iec/ieee international standard – systems and software engineering – system of systems (sos) considerations in life cycle stages of a system. 2019.

[11] I. ISO, IEC. Iso/iec/ieeee international standard – systems and software engineering – system of systems (sos) considerations in life cycle stages of a system. 2019.

[12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014.

[13] X. Meng, B. Thörnberg, and L. Olsson. Component obsolescence management model for long life cycle embedded system. In *2012 IEEE AUTOTESTCON Proceedings*, pages 19–24. IEEE, 2012.

[14] A. Meyer, L. Pretorius, and J. Pretorius. A model to manage electronic component obsolescence for complex or long life systems. In *2004 IEEE International Engineering Management Conference (IEEE Cat. No. 04CH37574)*, volume 3, pages 1303–1309. IEEE, 2004.

[15] F. Nemati, J. Kraft, and T. Nolte. Towards migrating legacy real-time systems to multi-core platforms. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 717–720. IEEE, 2008.

[16] F. Nemati, J. Kraft, and T. Nolte. A framework for real-time systems migration to multi-cores, 2009.

[17] A. B. E. S. E. T. NXP. Future advances in body electronics. `https://www.nxp.com/docs/en/white-paper/BODYDELECTRWP.pdf`, 2013 (accessed August 10, 2022).

[18] M. G. Pecht and D. Das. Electronic part life cycle. *IEEE Transactions on Components and Packaging Technologies*, 23(1):190–192, 2000.

[19] S. Pugh. The systems engineering tool box, 2009.

[20] D. Reinhardt and M. Kucera. Domain controlled architecture. In *Proc. Third International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2013)*, 2013.

[21] D. Reinhardt and G. Morgan. An embedded hypervisor for safety-relevant automotive e/e-systems. In *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, pages 189–198. IEEE, 2014.

[22] F. J. Romero Rojo, R. Roy, and E. Shehab. Obsolescence management for long-life contracts: state of the art and future trends. *The International Journal of Advanced Manufacturing Technology*, 49(9):1235–1250, 2010.

[23] J. Samuel, T. Redwine, and W. Riddle. Software technology maturation. In *Proceedings of the 8th international conference on Software engineering, London, England*, 1985.

[24] A. Sepúlveda, M. Schluep, F. G. Renaud, M. Streicher, R. Kuehr, C. Hagelüken, and A. C. Gerecke. A review of the environmental fate and effects of hazardous substances released from electrical and electronic equipments during recycling: Examples from china and india. *Environmental impact assessment review*, 30(1):28–41, 2010.

[25] L. Sha, T. Abdelzaher, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, A. K. Mok, et al. Real time scheduling theory: A historical perspective. *Real-time systems*, 28(2):101–155, 2004.

[26] M. Shaw. What makes good research in software engineering? *International Journal on Software Tools for Technology Transfer*, 4(1):1–7, 2002.

[27] J. A. Stankovic and e. K. Ramamritham. Tutorial: Hard realtime systems, 1987.

[28] J. Torresen and T. A. Lovland. Parts obsolescence challenges for the electronics industry. In *2007 IEEE Design and Diagnostics of Electronic Circuits and Systems*, pages 1–4. IEEE, 2007.

[29] E. Union. Smart grids and smart meters enable better management of energy networks and more efficient consumption. `https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters_en`, 2015 (accessed September 8, 2022).

[30] W. Wahlster. From industry 1.0 to industry 4.0: Towards the 4th industrial revolution. In *Forum Business meets Research*, 2012.

[31] 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*, volume 1, pages 29–39. Manchester, 2000.

[32] X. Yu and Y. Xue. Smart grids: A cyber–physical systems perspective. *Proceedings of the IEEE*, 104(5):1058–1070, 2016.

[33] Y. Zhang, T. Huang, and E. F. Bompard. Big data analytics in smart grids: a review. *Energy informatics*, 1(1):1–24, 2018.

[34] T. Zhao, Y. Zhang, and D. Zhang. Application technology of big data in smart distribution grid and its prospect analysis. *Power System Technology*, 38(12):3305–3312, 2014.

A Completed activities towards third cycle outcome

Table 4 shows an overview of the remaining third cycle outcomes, as stated in the Individual Study Plan (ISP).

Table 4: Overview of Third Cycle Outcomes towards PhD degree.

Remaining Third Cycle Outcomes - PhD	**Completed**
1 - Knowledge and understanding	**Yes**
1.2a Broad and systematic knowledge and understanding of the research area	**Yes**
1.2b Deep and specialist knowledge in a defined part of the research area	**Yes**
1.3 Deep knowledge of research methods in general, and of research methods in the specific research area	**Yes**
2. Competences and skills	**Yes**
2.4 Ability to conduct scientific analysis and critical assessment	**Yes**
2.5a Critically and independently identify and formulate research questions	**Yes**
2.5b, 2.6 Significantly contribute to the development of knowledge	**No**
2.7 Orally and in writing present and discuss research results with the national and international scientific community, and the society in general	**Yes**
2.8 Identify need of new knowledge, contribute to the development of society, and support learning of others	**Yes**
3. Judgment and approach	**Yes**
3.4 Show intellectual independence and scholarly integrity and ability to make ethical judgments	**No**
3.5 Demonstrate deep understanding of science's role and use in society, including its possibilities and limitations, and responsibility of its use	**Yes**

1. Knowledge and Understanding	**Completed**
For the doctoral degree, the doctoral student must: *1.2 Demonstrate broad knowledge and systematic understanding of the research as well as depth and current specialist knowledge in a defined part of the research, and 1.3 Show familiarity with scientific methodology in general and the specific research area in particular.*	
1.2a Broad and systematic knowledge and understanding of the research area	**Yes**

1. Knowledge and Understanding	**Completed**

I have taken the following courses:
- Number of courses, within the research area, are listen in Table 3

I have attended the following webinars, seminars, licentiate and doctoral proposals and presentations:

- Lic. presentation by Pablo Gutiérrez Peon
- Doctoral Presentation by Mikael Åslund
- ITS-Easy yearly meetings and study trips.
- 15 April 2019, Licentiate Thesis
- 2020 Nicklas Ericson Lic

I have attended the following schools, workshops and conferences:

- 2015 IEEE 13th International Conference on Industrial Informatics (INDIN)- COMPSAC 14
- IEESD 2014 is the 6th IEEE International Workshop on Industrial Experience in Embedded Systems Design
- 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA 13)
- 12th IEEE World Conference on Factory Communication Systems COMMUNICATION in AUTOMATION (WFCS 2016)
- ECSA, the 9th European Conference on Software Architecture
- Long-term Industrial Collaboration on Software Engineering (WISE2014)
- 2020 IEEE 25th Conference on Emerging Technologies & Factory Automation (ETFA 20)
- 2021 IEEE 26th Conference on Emerging Technologies & Factory Automation (ETFA 21)
- 2021 SOSE 16th Annual System of Systems Engineering Conference (SOSE 21)
- 2022 CIGRE Conference for Power Systems Expertise (CIGRE 22)
- All ITS-Easy Trips to other universities/Industries.

1.2b Deep and specialist knowledge in a defined part of the research area	**Yes**

1. Knowledge and Understanding	**Completed**

I have conducted a state of the art study: no

I have contributed with research results via the following publications:

- papers listed in section Section 5.2

I have participated in the following workshops and conferences:

- ETFA 2012
- ETFA 2013
- COMPSAC 2013
- IEESD 2013
- ETFA 2014
- COMPSAC 2014
- IEESD 2014
- INDIN 2015
- WFCS 2016
- ETFA 2020
- SoSE 2021
- ETFA 2021
- Cigree 2022

1.3 Deep knowledge of research methods in general, and of research methods in the specific research area	**Yes**

I have taken the following research methods courses and/or training:

- Research methods in natural science and engineering 7,5 hp
- Empirical research methods (VT 2014 7.5p)

I have reviewed the following scientific articles:

- Review of papers has been included in several of the courses and also for some of the ETFA conferences that we participated in.

2. Competences and skills	**Completed**
For the doctoral degree, the doctoral student must: *2.4 Demonstrate the ability to scientific analysis and synthesis and in independent, critical examination and assessment of new and complex phenomena, issues and situations.* *2.5 Demonstrate the ability to critically, creatively and with scientific accuracy identify and formulate issues and to plan and use appropriate methods to conduct research and other advanced tasks within given time frames and to review and evaluate such work.* *2.6 Demonstrate an ability, through its own research significantly contribute to the development of knowledge.* *2.7 Demonstrate ability in both national and international contexts orally and in writing with authority to present and discuss research and research results in dialogue with the scientific community and society in general.* *2.8 Show ability to identify needs for further knowledge and show what is required for both research and education and in other professional contexts contribute to the development of society and support the learning of others.*	
2.4 Ability to conduct scientific analysis and critical assessment	**Yes**

I have written and published the papers listed in 1.2b above.

I have reviewed the scientific articles listed in 1.3 above.

I have set up and supervised the following theses:

- I have supervised several different master thesis and summer workers during the past 20 years, e.g., Alfred Krappman (Cache handling x86), Fredrik Öman (Emulation of C167 hardware), Torbjörn Nilsson (Optical instrument transformer, FPGA)

I have written the following funding applications:

- Supported the authors in several differnet projects, e.g. EMC2, IMPRINT, DIGEST, Fiesta, COLAND

I have reviewed funding applications of others: no

- Supported the authors in several differnet projects, e.g. EMC2, IMPRINT, DIGEST, Fiesta, COLAND

2.5a Critically and independently identify and formulate research questions	**No**

I have written and published the papers listed in 1.2 above.

I have written the phd thesis proposal: yes

I have written the phd thesis: no

I have contributed to the funding applications listed in 2.1 above.

2.5b, 2.6 Significantly contribute to the development of knowledge	**Yes**

2. Competences and skills	**Completed**

2.5b, 2.6 Execute research I have assisted teaching and/or lectured in the following courses:
I have supervised the following theses:

- I have supervised several different master thesis during the past 20 years, e.g., Alfred Krappman (work on Cache handling), Fredrik Öman (Emulation of C167 CPU hardware), Torbjörn Nilsson (FPGA replication)

I have developed the following course material:

- I have created and conducted several internal and external courses for ABB/Hitachi, e.g. Introduction to the MACH control and protection system for HVDC and GPQS.

I have assisted teaching and/or lectured in the following courses:

- -

2.7 Orally and in writing present and discuss research results with the national and international scientific community, and the society in general	**Yes**

I have written and published the papers listed in 1.2 above.

I have attended, with own contributions, the following seminars, workshops and conferences:

- Written in 1.2 above.

I have participated in the following co-production research/projects:

- my own project in ITS-EASY with ABB/Hitachi and MDH/MDU
- EMC2, http:www.artemis-emc2.eu, Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments
- IMPRINT
- DIGEST
- Fiesta

I have disseminated research in society via the following activities:

- Presentation of ABB/Hitachi and related research to student

2.8 Identify need of new knowledge, contribute to the development of society, and support learning of others	**Yes**

I have supervised the theses listed in 2.4 above.

I have participated in the co-production research/projects listed in 2.7 above.

I have participated in the seminars listed in 2.7 above.

I have participated in the teaching listed in 2.6 above.

I have participated in written the funding applications listed in 2.4 above.

<table>
<tr><th>3. Judgement and approach</th><th>Completed</th></tr>
<tr><td>For the doctoral degree, the doctoral student must:
<i>3.4 Viewing intellectual independence and scholarly integrity and an ability to make ethical judgments.</i>
<i>3.5 View deeper insight into the possibilities and limitations, its role in society and the responsibility for its use.</i></td><td></td></tr>
<tr><td>3.4 Show intellectual independence and scholarly integrity and ability to make ethical judgements</td><td>No</td></tr>
<tr><td colspan="2">I have written and published the papers listed in 1.2 above.
I have written the lic proposal: yes
I have written the lic thesis: yes
I have written the phd proposal: yes
I have written the phd thesis: no
I am the research planning course and plan to take the research methodology as listed in Table 3</td></tr>
<tr><td>3.5 Demonstrate deep understanding of science’s role and use in society, including its possibilities and limitations, and responsibility of its use</td><td>Yes</td></tr>
<tr><td colspan="2">I have participated in the seminars listed in 2.7 above.
I have participated in the co-production research/projects listed in 2.7 above.</td></tr>
</table>