2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)

Automatic Network Configuration for Real-time, Distributed Industrial Automation Systems

Ben Schneider
fortiss GmbH
Munich, Germany
schneider@fortiss.org

***Abstract*—The current trend towards the Industrial Internet of Things raises new requirements regarding the connectivity of distributed automation systems. Time-sensitive Networking is a new IEEE standard fulfilling these requirements by providing determinism and real-time for convergent networks. The networks in this domain are currently configured manually, but the rising number of network nodes, more frequent reconfigurations and the combination of different scheduling mechanisms will make a manual network configuration infeasible.**

My proposed Ph.D. thesis strives to automate the network configuration for real-time, distributed industrial automation systems. This can be accomplished by deducing the relevant network requirements from extended domain-specific models (e.g., IEC 61499) and synthesizing network schedules according to the deduced information that can be transformed into constraints and solved by constraint-based solvers like Satisfiability Modulo Theorem solvers.

***Index Terms*—Automatic Network Configuration, Real-time Networks, TSN, Convergent Networks, Distributed Industrial Automation Systems**

I. Motivation

The current trend towards the Industrial Internet of Things (IIoT) is going to change the domain of automation systems from centralized control to distributed processes with high demands towards connectivity of the different devices. This trend results in a higher modularity of production plants which is a key enabler for easy *reconfiguration* (Software and Hardware) of automation systems [24]. Technologies like IEC 61499, the domain specific modelling language for distributed industrial automation systems, and the new IEEE Time-Sensitive Networking (TSN) standard provide solutions for future-proof industrial automation systems.

A. Modelling of Distributed, Industrial Automation Systems

IEC 61499 defines different models for the development of distributed industrial automation systems [1] [39]. Figure 1 shows IEC 61499's Application, System and Distribution Model.

The *Application Model* contains Function Blocks (FBs) which encapsulate system functionality (e.g., arithmetic operations, IO access, network access, ...). The FB's interface consists of event and data interface elements used to interact with an FB. A sequence of events (event chain) connecting different FBs and their related data form an IEC 61499 compliant application.

IEC 61499's *System Model* provides information about the physical setup of the target system, consisting of devices (the Programmable Logic Controllers (PLCs)), network segments representing the connection of PLCs via the specified network devices.

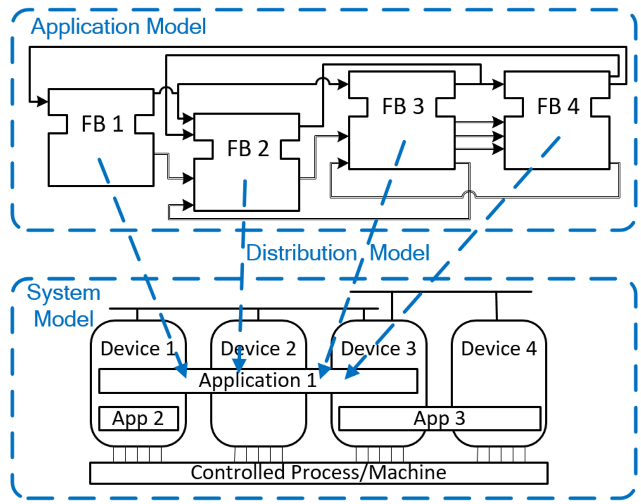

Fig. 1. Models of IEC 61499 [30]

Finally, the *Distribution Model* is responsible for mapping FBs to the devices. Communication Service Interface Function Blocks (e.g., publish/subscribe or client/server communication) are added manually to the distributed application in order to make FBs which belong to the same application, but are distributed on different devices, able to communicate with each other. A complete description of all models of IEC 61499 can be found in the book by Zoitl and Lewis [39].

The usual workflow of an engineer, who is developing a distributed system according to IEC 61499, is to start with modelling the systems functionality (Application Model), continued with the physical setup of the plant (System Model), followed by mapping the application to the devices (Distribution Model). In the end, the steps for establishing a communication channel between FBs, which run on different devices, have to be performed manually by:

1) choosing the communication pattern (publish/subscribe, or client/server)
2) choosing the fieldbus/protocol to be used (e.g., OPC UA, EtherNet/IP, UDP, or TCP)
3) configuring protocol specific parameters on the forward-

978-1-7281-5125-0/19/$31.00 ©2019 IEEE
DOI 10.1109/MODELS-C.2019.00096

ing devices (e.g., traffic paths, or schedules)

4) configuring protocol specific parameters on the end devices (e.g., IP address, or port)

This final network configuration procedure is repeated for each and every connection and is tedious and error-prone [26] [36], but was feasible, despite the high effort, because current distributed systems are rarely reconfigured and usually run for decades after the initial manual configuration. Requirements of future automation system's networks change drastically towards reconfiguration and connectivity. The IEEE is currently working on a new family of standards summarized in IEEE802.1 Time-Sensitive Networking, that can cope with the new connectivity requirements.

B. Time-Sensitive Networking adds Additional Features to Standard Ethernet

The configuration of networks in today's automation systems is a very complex task, because each of the big market players has its own proprietary fieldbus protocol which is operated and configured in a different manner but is generally used in similar use cases and applications, like high performance motion control. Overall there are over 10 Ethernet-based fieldbus protocols in the domain of DIAS and over 20 which are not Ethernet-based. These fieldbuses are not *interoperable* because of different base communication technologies that are used, for example EtherCAT using a modified Ethernet frame versus Powerlink which uses the standard compliant Ethernet frame [25]. However, some of these fieldbuses can be connected via expensive gateways that the manufacturers provide in order to convert frames from one fieldbus to another, which also causes a performance penalty in most cases.

The fieldbuses do not only differ during operation, but also during configuration. Each manufacturer has its own configuration tool, which needs a tremendous amount of time and requires detailed and fieldbus-specific knowledge.

Apart from these proprietary fieldbus protocols developed by different companies, the IEEE started in 2012 to work on the vendor independent Time-sensitive Networking (TSN) standards. TSN can simplify the hardware composition and enhance the interoperability of the connected devices by providing amongst others the following features:

- **Convergence:** Transmission of network packets with different requirements over the same network cable (i.e., different traffic classes)
- **Interoperability:** Devices of different vendors can communicate via a common network protocol (standard Ethernet)
- **Determinism:** Guaranteed transmission of network packets
- **Real-time:** Guaranteed transmission of network packets within a defined time (i.e., maximum latency)
- **Availability:** A fail-operational state for example through redundant network links or mechanism for redundant master clocks for time synchronization

As TSN is an extension of the Ethernet standard (ISO OSI layer 2), these new features can be used by standard Ethernet-based protocols. Thus, TSN provides the infrastructure for existing and future Ethernet-based protocols to be real-time capable and deterministic in domains like industrial automation, automotive, and others.

Different TSN Standards. The convergent time-sensitive network is achieved by a set of schedulers which are standardized by the IEEE 802.1 TSN Task Group in order to meet the requirements of the different use cases. Examples for real-time schedulers are IEEE 802.1Qbv (Time Aware Shaper) which is following a Time-Division-Multiple-Access (TDMA) approach [3] and targets for example cyclic, high performance motion control traffic, IEEE 802.1Qav (Credit Based Shaper) targeting high bandwidth streams like video streams for quality management or safety critical processes [2] IEEE 802.1Qcr (Asynchronous Traffic Shaping - not published yet, currently under standardization) that targets event-based sensor data with dynamically changing bandwidth requirements [6] or IEEE 802.1Qch (Cyclic Queueing and Forwarding) which delays the traffic for a max. amount of time at each hop and targets for example process or energy automation where the huge size of the machinery can just not benefit from very low latency on the network [23] and therefore sustain a certain level of latency and jitter [5].

All these use cases and their related schedulers can in the future be realized by the Ethernet-based TSN communication, that combines network traffic with different requirements on the same cable (i.e., the convergent network). This convergent network allows different Ethernet based protocols to coexist side by side and use the same TSN infrastructure.

The number of different TSN schedulers is expected to grow caused by new industrial use cases.

Impact on the Industrial Automation Domain. The convergent time-sensitive network has a high impact on the interoperability of the network, by enabling the transmission of different kind of traffic via the same network infrastructure. The combination of OPC UA and TSN further extends the interoperability from a technical to a syntactical and semantic level through standardized OPC UA Companion Secifications. OPC UA over TSN is considered as the first vendor-independent, real-time capable communication solution for the factory of the future [12] [37]. OPC UA is a middleware for distributed systems in the automation industry, which includes communication mechanisms, information models and built-in security amongst other things. The importance of OPC UA over TSN as an interoperable networking standard is also demonstrated by the TSN Shaper Group (industrial consortium for validating TSN standards for the automation domain, consisting of all the big players in Europe's automation domain) who officially joined the OPC Foundation in November 2018 [13] and formed the new Field Level Communication Task Group. This is the first step towards a vendor independent "fieldbus" for the industrial automation domain, which is developed by all major players of the domain together.

II. Problem Summary and Hypothesis

In order to summarize one could say, that the convergent, time-sensitive network can reduce cabling effort, simplify the hardware composition of a production plant, and fulfil the increasing interoperability requirements of distributed industrial automation systems (DIAS). The *problem* of adopting TSN in the industry is the high configuration complexity [23] [37] [36] [22] [14] [13].

Therefore I summarize the problem to be solved by my Ph.D. thesis as follows:

The task of network configuration for time-sensitive networks is manually unsolvable due to the rise of complexity, which is caused by convergent traffic, the growing number of networking nodes, and reconfiguration at runtime.

The infeasibility of a manual network configuration immediately results in the question, how network configuration can be automated. My approach is based on real-time network modelling in the industrial automation domain and the deduction of relevant network parameters from the models, which serve as input for the constraint-based schedule synthesis. The toolchain to be developed for automatic network configuration will reduce the engineering effort of distributed industrial automation systems. Therefore, the *hypothesis* of my Ph.D. thesis is summarized as follows:

The configuration for convergent Time-Sensitive Networks can be automated using network modelling and schedule synthesis in the domain of distributed industrial automation systems.

III. Research Description and State of the Art

The main problem tackled in my thesis is complexity. This complexity is particularly caused by new requirements motivated by the automation domain which are the *distribution of control logic* (keyword Industrial IoT), the *availability* in terms of fail-safe operation of the plant and the demand for frequent *reconfiguration* of production plants [24] [39]. This domain-specific requirements trigger the need for novel networking concepts to be developed and adopted. The *convergent network* is for example needed in order to cope with the distributed application, because it is not feasible to create physically separated networks for the different parts of a distributed control application [37]. The *availability of the network* is another key requirement as the availability of the production plant relies on a working network. Last but not least, an industrial network needs to fulfill *deterministic and real-time requirements*. The use cases I want to consider are bound to the boundaries of a factory and are on the one hand the controller-to-controller communication, where a distributed application needs to communicate in real-time on a horizontal level (e.g., a distributed closed loop controller), and on the other hand communication where controllers transmit data into the local cloud, which is in the logical boundaries of the factory [37].

For tackling this complexity and striving towards the goal of automatic network configuration for DIAS, I identified the research questions described in the following sections.

A. Real-time Network Modelling in the Domain of Distributed Industrial Automation Systems

For an appropriate model of a DIAS, I chose models according to the standard IEC 61499 [1]. This domain specific modelling language fulfills the aforementioned requirements. However, it does not provide network modelling in a satisfactory way [30] [20] [36]. Therefore, the fundamental question addressed in this part of the thesis is:
How can real-time networks in the domain of distributed industrial automation systems be modeled?

1) ***RQ 1.1*** *What are the network modelling capabilities of IEC 61499 and which parameters, configuration steps, device types and device features are missing to configure convergent TSN networks?:* Network modelling for distributed industrial automation systems is a continuously evolving field of research. Froschauer et al. [20] and Wenger et al. [36] already showed that the current network modelling capabilities of IEC 61499 are not accurate enough to perform (real-time) network modelling. However, they do not state which parts of IEC 61499 are missing for network modelling. This gap is partly solved in my previous publication [30].

This research question is a requirements analysis for configuration information based on IEC61499 that is on the one hand needed as input for synthesising convergent TSN schedules (e.g., network device information, routes, latency and jitter requirements) and on the other hand for storing the results of the solver (e.g., sending and receiving times of network packets, the priority of streams or the duration of transmission time windows of a schedule) in modelling elements of IEC61499, for example by generating communication Function Blocks similar to what Lednicki et al. [26] started.

2) ***RQ 1.2*** *How can we enhance the models of IEC 61499 in order to fulfill real-time network modelling?:* After the requirements analysis performed in RQ1.1, the next step is to enhance the models of IEC61499 for real-time networking according to TSN by either showing prototypical implementations of additional models that could be added to the IEC61499 standard or by an interface to another modelling language like Automation Modelling Language (AML) [10], Architecture Analysis and Design Language (AADL) [19], UML Modeling and Analysis of Real-time and Embedded systems (MARTE) [9] or others.

The FRONTICS project [21] [36] [20] [29] aimed at an automatic network configuration by modeling software and hardware of a system and relate them via parametrized communication relationships in AADL. This work can be extended in the following way: The existence of TSN simplifies the physical structure of a network and therefore changes the communication setup. I want to specifically extend their work by taking real-time network and TSN scheduling parameters into consideration.

B. Constraint-based Schedule Synthesis for Time-sensitive, Convergent Networks in the Domain of Distributed Industrial Automation Systems

Time-triggered scheduling is a bin-packing problem and therefore known to be NP-complete [33]. The complexity of this design space exploration (DSE) problem can be handled by general purpose constraint-based solvers like Satisfiability Modulo Theorem (SMT). Steiner et al. [32] used SMT solvers for schedule synthesis of time-triggered networks in 2010. The usage of SMT solvers is since then adopted for solving scheduling problems for TTEthernet [32] [15] and later for the Time-Aware Shaper [33] [17] [16] [18] and for the Peristaltic Shaper [31] (the last two are both TSN schedulers). In the following years, the field of TSN schedule synthesis using general purpose tools was extended towards combined task and network level scheduling using Mixed Integer Programming (MIP) [38] [15]. Voss et al. evaluated a combined deployment and schedule synthesis for mixed-criticality applications [35].

However, these methodologies are (to the best of my knowledge) not used in the domain of DIAS, where networks (including real-time schedules) are still configured manually and domain specific adjustments are needed for the already existing constraints. The optimization of the schedulers execution time (scalability in DSE) is also an active field of research [33] [34]. Therefore, the fundamental question addressed in this part of my thesis is summarized to:
How can we adopt existing methodologies for constraint-based schedule synthesis for Distributed Industrial Automation Systems?

1) ***RQ 2.1*** *How can we extend the existing scheduling constraints with domain-specific constraints?:* This research question focuses on the domain specific communication requirements summarized in the beginning of this chapter. The indicators according to these performance and availability requirements are bounded latency, bandwidth and jitter with a specific cycle and update time using the appropriate mechanism for time synchronisation and fulfilling a given recovery time [37]. How these domain-specific requirements can be transformed into constraints, on top of the already existing scheduling constraints like Steiner et al. [33] Craciunas et al. [16] or Farzaneh et al. [18], for a network schedule synthesis tool is to be answered here.

2) ***RQ 2.2*** *How can we combine different TSN scheduling mechanisms and perform network schedule synthesis?:* TSN provides the convergent network, where traffic with different requirements can be transmitted on a single network cable instead of having multiple, physically independent network setups. The TSN testbed Industrial Internet Consortium (IIC)[1] recently released a whitepaper defining the different traffic types of industrial automation traffic [11]. Their future work is to allocate the already existing TSN schedulers to the defined traffic classes and create additional schedulers for traffic classes that have no suitable scheduler yet. The scheduling constraints for most of these schedulers have been discussed in various publications for example for the Time-Aware Scheduler in [17] [33] [18], or the Peristaltic Shaper in [31]. My goal in this research question is to analyse these already existing scheduling constraints and to combine in order to achieve a convergent network. Specifically interesting w.r.t. to DIAS will be, for example cascading different schedulers e.g., using the mechanisms of the Peristaltic Shaper inside a transmission time slot that was reserved according to the TAS.

3) ***RQ 2.3*** *How can the execution time of the constraint based solver be optimized for schedule synthesis in DIAS?:* The goal of this research question is to optimize the execution time of the constraint solver (which grows exponentially) for the domain-specific TSN scheduling constraints. I plan to tackle this problem, by dividing the problem into smaller fragments, calculating a solution for the fragments and combining them afterwards (divide-and-conquer). A reasonable starting point for such investigations is to consider different routing strategies and their impact on the execution time of the scheduler [33] [27]. These routing strategies include isolation of flows, middlebox traversal (meaning that a stream has to be routed through a specific networking device) or network engineering for example with load balancing [34].

IV. Solution

The TSN standard IEEE 802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements [4] mainly standardizes how TSN networks shall be configured and therefore specifies three types of configuration architectures: a fully centralized, a fully distributed and a hybrid architecture, that combines the centralized with the distributed approach. As the fully distributed approach currently lacks features for configuring certain relevant TSN standards (e.g., the Time-Aware Shaper [3]) [4] and the hybrid approach is currently not relevant in the industry [13], my thesis will focus on centralized model, which is shown in Figure 2. The implementation in the different levels of the approach

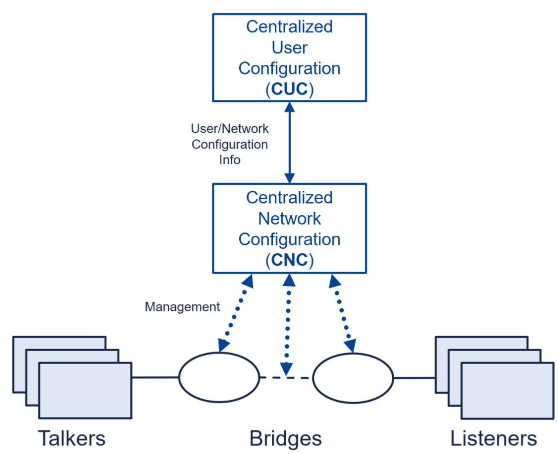

Fig. 2. Centralized configuration approach according to IEEE 802.1Qcc [4]

is described in the following:

- **Centralized User Configuration (CUC):** derive the network requirements from IEC61499 models by a prototypical implementation in the Eclipse 4diac framework.

[1] https://www.iiconsortium.org/

- **User/Network Configuration Information:** connect the requirements gathered in the CUC to the Centralized Network Configuration (CNC) engine. As Z3 (an SMT solver) has a Java API, it could directly be integrated as a plugin into the 4diac IDE (Java-based). The second and more reusable solution would be, to use Automation ML (AML) models as data exchange format between 4diac, the CNC and/or other tools.
- **Centralized Network Configuration (CNC):** automatically choose the suitable TSN scheduler used for the transmission of traffic with the modeled requirements. The next step is to compute the network configuration, which would include a valid Time-Aware Shaper (IEEE 802.1Qbv) configuration, using the SMT solver Z3.
- **Management Interface:** transform the scheduling results into YANG models and generate the appropriate communication FBs using 4diac. YANG models [7] provide network modelling structures, which can be sent via the network management protocol NETCONF [8] to forwarding devices in order to configure them.
- **Fieldlevel:** deploy the network configuration (YANG for forwarding devices and 4diac communication FBs for end devices) received from the CNC.

V. Contribution Summary

The first part of the expected contribution is a sophisticated real-time network model enhancing the standard IEC 61499 that can cope with the requirements of convergent TSN networks for example for storing configuration parameters in the communication FBs. These extensions are needed in order to gather the input parameters for the network schedule synthesis and to store the results of the solvers. The real-time network modelling results may influence the extension of the IEC 61499 standard.

The second part of the contribution is the extension of constraint-based network schedule synthesis with domain-specific requirements from distributed industrial automation systems. Therefore, I want to combine network streams with different requirements from the domain on the same network, and schedule them according to the TSN standard which is most appropriate for the respective stream. Additionally, I strive to optimize the execution time of SMT solvers for the TSN scheduling problem by a divide and conquer approach.

VI. Case Studies / Evaluation

The toolchain, described in the last section, shall be evaluated on the SMS-Demonstrator, that is used to evaluate concepts in the BaSys4.0 research projects. Current plans for BaSys4.2 also involve the extension of this demonstrator in the upcoming 3 years of the project. The demonstrator is a simulated process, producing thin aluminium pieces and foil out of huge coils. This process is controlled via distributed controllers currently connected by a proprietary and high performance network infrastructure that shall be exchanged by a TSN network infrastructure. The performance of the generated network schedules for this use case will be tested via network experiments, that confirm the required characteristics of latency, jitter and availability.

As the toolchain to be developed aims at automating the configuration process of TSN networks for distributed automation systems, the evaluation will also include usability tests of the tools and the maintainability of the schedules, generated by the developed tools.

VII. Assumptions and Scope

My thesis focuses on requirements coming from the domain of DIAS with strong focus on models of IEC 61499. However, the constraints for TSN network schedule synthesis are independent of the domain and can be reused, but have to be combined with additional constraints reflecting the network mechanism relevant for other domains.

The network modelling will be according to the standards of IEEE 802.1 TSN (switched Ethernet). Other protocols/fieldbuses are not considered in this work and might result in slightly different configuration requirements. This means that the link layer (ISO OSI layer 2) is the main layer considered in my thesis and requirements or solutions which can not be fulfilled by this layer are not in the scope of my thesis. However, these requirements like safety and security are crucial for the successful operation of a DIAS and the developed concepts and toolchain shall be extensible towards these additional requirements. Furthermore, I assume the correct transmission of signals on ISO OSI layer 1 and do not take this layer into consideration.

The communication out of the boundaries of the factory are not in the scope of my thesis, as they involve other challenges e.g., the convergence of real-time traffic from different companies transmitted on the infrastructure of a third level network provider.

VIII. Research Plan

Some preliminary work for my Ph.D. project has already been done. In the previous publication [30], I identified the present and missing elements for modelling TSN networks in an IEC61499-compliant way. Our most recent publication at the Formal Methods in Computer-Aided Design (FMCAD) conference 2019 proposes an open source tool *TSNSched*[2], that allows the scheduling of multicast streams with the Time-Aware Shaper using the Z3 SMT solver [28]. The 4diac runtime environment (forte) is currently extended by function blocks that support the time synchronization of the IEC61499-compliant application to network times. Additionally a communication layer was implemented that reduces the jitter of forte's communication architecture to be applicable in high performance TSN use cases.

Overall, I plan to finish the domain-specific, real-time network modelling (RQ1.1 and RQ1.2) by mid 2020 and the schedule synthesis (RQ2.1, RQ2.2 and RQ2.3) by March 2021. The final thesis will be finished in the beginning of 2022.

[2]https://github.com/ACassimiro/TSNsched

References

[1] IEC 61499 part 1: Architecture.

[2] IEEE 802.1Qav - Forwarding and Queuing Enhancements for Time-Sensitive Streams. Available online at http://www.ieee802.org/1/pages/802.1av.html last accessed on October 30th 2018.

[3] IEEE 802.1Qbv - Enhancements for Scheduled Traffic. Available online at http://www.ieee802.org/1/pages/802.1bv.html last accessed on October 30th 2018.

[4] IEEE 802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. Available online at https://1.ieee802.org/tsn/802-1qcc/ last accessed on October 30th 2018.

[5] IEEE 802.1Qch Cyclic Queuing and Forwarding . Available online at https://1.ieee802.org/tsn/802-1qch/ last accessed on October 30th 2018.

[6] IEEE 802.1Qcr - Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping. Available online at https://1.ieee802.org/tsn/802-1qcr/ last accessed on October 30th 2018.

[7] IEEE 802.1Qcw YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing. Available online at https://1.ieee802.org/tsn/802-1qcw/ last accessed on March 15th 2019.

[8] RFC6241 - Network Configuration Protocol (NETCONF). Available online at https://tools.ietf.org/html/rfc6241 last accessed on March 15th 2019.

[9] A UML profile for MARTE: Modeling and analysis of real-time embedded systems, beta 2. Available online at https://www.omg.org/omgmarte/ last accessed on January 08th 2019.

[10] Whitepaper Automation ML - Communication. Available online at https://www.automationml.org/o.red.c/publications.html last accessed on January 08th 2019.

[11] A. Ademaj, D. Puffer, D. Bruckner, G. Ditzel, L. Leurs, M. Stanica, P. Didier, R. Hummen, R. Blair, and T. Enzinger. Iic results white paper: Time sensitive networks for flexible manufacturing testbed - description of converged traffic types, April 2018.

[12] D. Bruckner, R. Blair, M. P. Stanica, A. Ademaj, W. Skeffington, D. Kutscher, S. Schriegel, R. Wilmes, K. Wachswender, L. Leurs und M. Seewald, R. Hummen, E. C. Liu, and S. Ravikumar. OPC UA TSN A new Solution for Industrial Communication. 2018.

[13] D. Bruckner, M. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter. An introduction to OPC UA TSN for industrial communication systems. *Proceedings of the IEEE*, pages 1–11, 2019.

[14] S. S. Craciunas, R. Serna Oliver, and W. Steiner. Demo abstract: Slate XNS–an online management tool for deterministic TSN networks. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 103–104, April 2018.

[15] Silviu S. Craciunas and Ramon Serna Oliver. Combined task- and network-level scheduling for distributed time-triggered systems. *Real-Time Systems*, 52(2):161–200, Mar 2016.

[16] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems, RTNS 2016, Brest, France, October 19-21, 2016*, 2016.

[17] Silviu S. Craciunas, Ramon Serna Oliver, and Wilfried Steiner. Formal scheduling constraints for time-sensitive networks. *CoRR*, abs/1712.02246, 2017.

[18] M. H. Farzaneh, S. Kugele, and A. Knoll. A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Sept 2017.

[19] Peter H. Feiler and David P. Gluch. *Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis Design Language*. Addison-Wesley Professional, 2014.

[20] R. Froschauer, F. Auinger, A. Schimmel, and A. Zoitl. Engineering of communication links with AADL in IEC61499 automation and control systems. In *2009 7th IEEE International Conference on Industrial Informatics*, pages 582–587, June 2009.

[21] Roman Froschauer, Alois Zoitl, and Martijn Rooker. Frontics - a communication framework for networked automation and control systems*. *IFAC Proceedings Volumes*, 2010. 10th IFAC Workshop on Intelligent Manufacturing Systems.

[22] M. Gutirrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat. Self-configuration of IEEE 802.1 TSN networks. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, Sept 2017.

[23] Oliver Kleineberg and Axel Schneider. *Time-Sensitive Networking - for dummies*. John Wiley Sons, Inc., 2018.

[24] Yoram Koren. *The global manufacturing revolution: product-process-business integration and reconfigurable systems*, volume 80. John Wiley & Sons, 2010.

[25] L. Lachello, P. Wratil, A. Meindl, S. Schnegger, B. Karunakaran, H. Song, and S. Potier. Industrial ethernet facts, 2012.

[26] L. Lednicki and J. Carlson. A framework for generation of inter-node communication in component-based distributed embedded systems. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8, Sept 2014.

[27] N. G. Nayak, F. Drr, and K. Rothermel. Incremental flow scheduling and routing in time-sensitive software-defined networks. *IEEE Transactions on Industrial Informatics*, May 2018.

[28] Aellison Santos, Ben Schneider, and Vivek Nigam. TSNsched: Automated schedule generation for time-sensitive networking. *Formal Methods in Computer-Aided Design 2019*, 2019.

[29] A. Schimmel, A. Zoitl, R. Froschauer, M. Rooker, and G. Ebenhofer. Model-driven communication routing in industrial automation and control systems. In *2010 8th IEEE International Conference on Industrial Informatics*, pages 896–901, July 2010.

[30] Ben Schneider, Sebastian Voss, Monika Wenger, and Alois Zoitl. Using IEC61499 models for automatic network configuration of distributed automation systems. Feb 2018.

[31] J. Specht and S. Samii. Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet. In *2017 IEEE Real-Time Systems Symposium (RTSS)*, pages 178–187, Dec 2017.

[32] W. Steiner. An evaluation of smt-based schedule synthesis for time-triggered multi-hop networks. In *2010 31st IEEE Real-Time Systems Symposium*, pages 375–384, Nov 2010.

[33] W. Steiner, S. S. Craciunas, and R. S. Oliver. Traffic planning for time-sensitive communication. *IEEE Communications Standards Magazine*, 2(2):42–47, JUNE 2018.

[34] Kausik Subramanian, Loris D'Antoni, and Aditya Akella. Genesis: Synthesizing forwarding tables in multi-tenant networks. *SIGPLAN Not.*, January 2017.

[35] S. Voss and B. Schätz. Deployment and scheduling synthesis for mixed-critical shared-memory applications. In *2013 20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, pages 100–109, April 2013.

[36] M. Wenger, A. Zoitl, R. Froschauer, M. Rooker, G. Ebenhofer, and T. Strasser. Model-driven engineering of networked industrial automation systems. In *2010 8th IEEE International Conference on Industrial Informatics*, pages 902–907, July 2010.

[37] M. Wollschläger, T. Debes, J. Kalhoff, J. Wickinger, H. Dietz, G. Feldmeier, J. Michels, H. Scholing, and M. Billmann. Kommunikation im industrie 4.0 umfeld, April 2018.

[38] L. Zhang, D. Goswami, R. Schneider, and S. Chakraborty. Task- and network-level schedule co-synthesis of ethernet-based time-triggered systems. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 119–124, Jan 2014.

[39] A. Zoitl and R. Lewis. *Modelling Control Systems Using IEC 61499*. The Institution of Engineering and Technology, 2014.