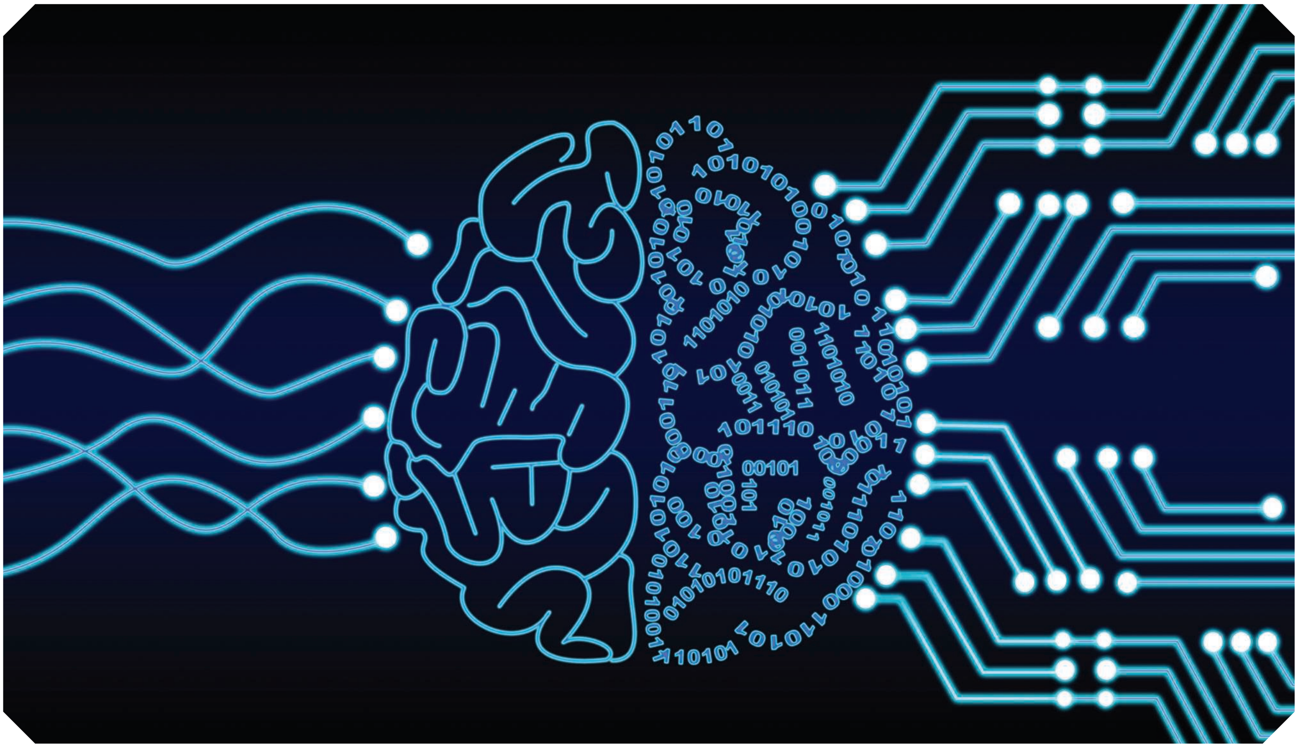
Smart Transducers in Distributed and Model-Driven Control Applications

Empowering Seamless Internet of Things Integration



©ISTOCKPHOTO.COM/FASBENTEK

HANS-PETER BERNHARD,
ALOIS ZOITL, and
ANDREAS SPRINGER

Digital Object Identifier 10.1109/MIE.2019.2936626
Date of current version: 23 December 2019

Smart transducers based on ISO/IEC/IEEE 21450 provide standardized access to sensors and actuators. Hence, no centralized database is needed for the calibration and interpretation of the data collected by transducers. The smart transducer stores and maintains its individual transducer electronic data sheet (TEDS). Whenever transducers are replaced in an automation system, the correct description, calibration, and data stream interpretation come along with the newly connected transducer. The domain-specific modeling language defined by IEC 61499 improves the software development process for automation systems. However, it lacks a clear transducer interaction model.

In this article, we combine these two major ideas of industrial automation to reap the benefit of easy-to-use smart factory computation methods. We show how modeling means are adapted to use the TEDS description of any smart transducer. Such models communicate with the transducer at the time of execution. When transducers or models are changed, automatic validation allows a short commissioning phase.

Background

In recent years, we have seen a major change in production and machine automation systems. Volatile global markets, shorter product lifecycles, and higher product variants require the quick and easy adaptation of production systems and their components. A core element to achieve the required adaptability is the increase in software size and, therefore, software complexity in production and machine automation systems. New means and methods are necessary to manage this complexity and reduce the expanding software development effort.

Model-driven software engineering has shown a great potential to be a means of reducing software engineering effort in different domains [1], [2]. Especially for industrial automation and control systems, a domain-specific modeling language was defined in standard IEC 61499 [3]. IEC 61499 focuses on a strong encapsulation of software components and hardware-independent software modeling to increase software quality and reusability [4], [5]. Recent research has shown that the models defined in IEC 61499 have the potential to greatly reduce the development effort compared to existing industrial software engineering approaches [6]. However, a main hindrance to leveraging these advantages in industry is that control applications require access to physical inputs and outputs. This requirement introduces additional and partly hidden dependencies on the hardware and, therefore, reduces the reusability and testability of control software components [7]. Furthermore, any input–output (IO) subsystem configuration (for example, fieldbuses, sensor, and actuator characteristics) are not treated by IEC 61499 at all [8].

Here, the ISO/IEC/IEEE 21450 standard for smart transducers supports the elimination of hidden hardware dependencies and introduces a well-documentable flexibility. The TEDS, as part of the ISO/IEC/IEEE 21450 smart transducer standard family, allows machine-readable data sheets to be stored on the transducer devices. The data sheets are always available to describe the sensor and actuator characteristics [9]. Furthermore, a unified access to the smart transducer is defined. This allows for the definition of generic requirements in IEC 61499 control applications and, during the setup phase, the matching of the IO requirements of the application with the available transducers.

In this article, we investigate the requirements and processes from a control application perspective and show how ISO/IEC/IEEE 21450 can support their fulfillment. In addition, the IEEE 2700-2017 [10] describes a methodology for defining sensor performance parameters with the goal of reducing system integration efforts. It defines a minimum set of performance parameters with the required units, conditions, and distributions for different types of sensors. These definitions can be included as an add-on to the TEDS description of ISO/IEC/IEEE 21450.

Finally, there is another direct link to be mentioned. The International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO) adopted the IEEE standards 1451.0 as ISO/IEC/IEEE 21450 and 1451.1 as ISO/IEC/IEEE 21451-1.

Smart Transducers ISO/IEC/IEEE 21450 Standards Family

A smart transducer is a sensor or actuator combined with a nonvolatile memory and a processing device that can communicate with other devices. Smart sensors send their measurement data not only as bitstreams but also provide metadata to allow a physically correct interpretation of the measurement data for any other device, such as a controller. The ISO/IEC/IEEE 21450 family of standards introduces a standardized description of metadata that is stored directly on the smart transducer, in the form of the so-called TEDS. These metadata are aligned with the raw measurement values and fully describe the sensor measurements. The transducer provides that the TEDS can be read in the same way as the data. Therefore, all devices that are capable of reading the data can also read their interpretation. Even if the sensor is new to the system, no external instance has to coordinate description and data.

An architectural overview of the standardized communication for smart transducers is given in Figure 1. It covers the relation between the standard and the physical, or logical, entities. Via the transducer interface module (TIM) and the network-capable application processor (NCAP) (ISO/IEC/IEEE 21451-1), the standardized network connectivity with Discovery Services, Transducer Access Services, TEDS Access Services, Event Notification Services, and Transducer Management Services is realized. These services are available via an application programming interface (API) or via standardized protocols: TCP/IP, HTTP, XMPP, SNMP, MQTT, and MQTT-SN (ISO/IEC/IEEE 21451-1-x) [11], [12].

TEDS

TEDS are memory blocks stored in a standardized format in nonvolatile memory within a TIM. The TEDS data blocks are mainly binary and organized as a byte array, where each parameter follows the type, length, and value definition. In general, TEDS data are stationary, and either the sensor manufacturer or an authorized user defines the TEDS information and writes it on the TIM. Several exceptions based on specific use cases for time variant TEDS exist, but this is outside the scope of this article. While further TEDS are optional, all TIMs require at least the following four TEDS.

1) The *Meta TEDS* sets basic communication parameters, such as time-out values to support the communication with TIM. Parameters, such as the number of transducer channels, describe the complexity of the TIM and how many individually measured variables are provided.
2) The *transducer channel TEDS* provides an interface to all information concerning the transducer channel to enable its correct operation. The details for the parameters can be found in ISO/IEC/IEEE 21450-2010 [13]. One of the parameters describes whether the channel is a sensor, an actuator, or an event sensor. Furthermore, a smart transducer features calibrated

values; therefore, each transducer channel is linked to calibration data. The parameter in the transducer channel TEDS is mandatory but not the calibration TEDS. In the transducer channel TEDS, the defined unit fields are mandatory. Therefore, each parameter to be measured or controlled has an International System of Units (SI) unit and can be processed according to the TEDS information. Furthermore, the TEDS also defines the operating range of the transducer channel, the characteristics of the digital IO, the operational mode of the unit, and the timing information.

3) The *transducer name TEDS* provides a place to store a name via which the system or the end user identifies this transducer.
4) The *physical layer TEDS* (*PHY TEDS*) relies on the physical communication media that are used to connect the TIM to the NCAP and is left open in this standard. The detailed information on those communication links is defined in [14].

Although this standard covers the sensor data and their representation, it completely lacks all security aspects. Further extensions of the TEDS standard will have to deal with signing the data on the sensor and certifying the calibration for a certain period of validity. In addition, other aspects of sensor communication, such as lossless compression [15], must also be included in the standard to meet energy restrictions for low power sensor nodes [16].

NCAP

An NCAP covers the link between the TIM and other communication partners, such as network communications, TIM communications, data conversion functions, and application functions. It also provides access to the TIM circuitry to switch functionality by changing transducer channels available on the TIM. The NCAP usually contains a controller and the interface to a network that may support other nodes. An NCAP could be an 8-b microprocessor for a simple proprietary control network, or a 32-b microprocessor for an Ethernet-based control network. Once a TEDS is read, the NCAP knows the data rate of the TIM, how many channels a TIM provides, and the data format of each TIM's transducer. It also knows the physical units of the sample values and how the samples are mapped to SI units.

An NCAP can be a gateway-like device connected to the arbitrary network and also to more than one TIM device. This allows for a reduction of

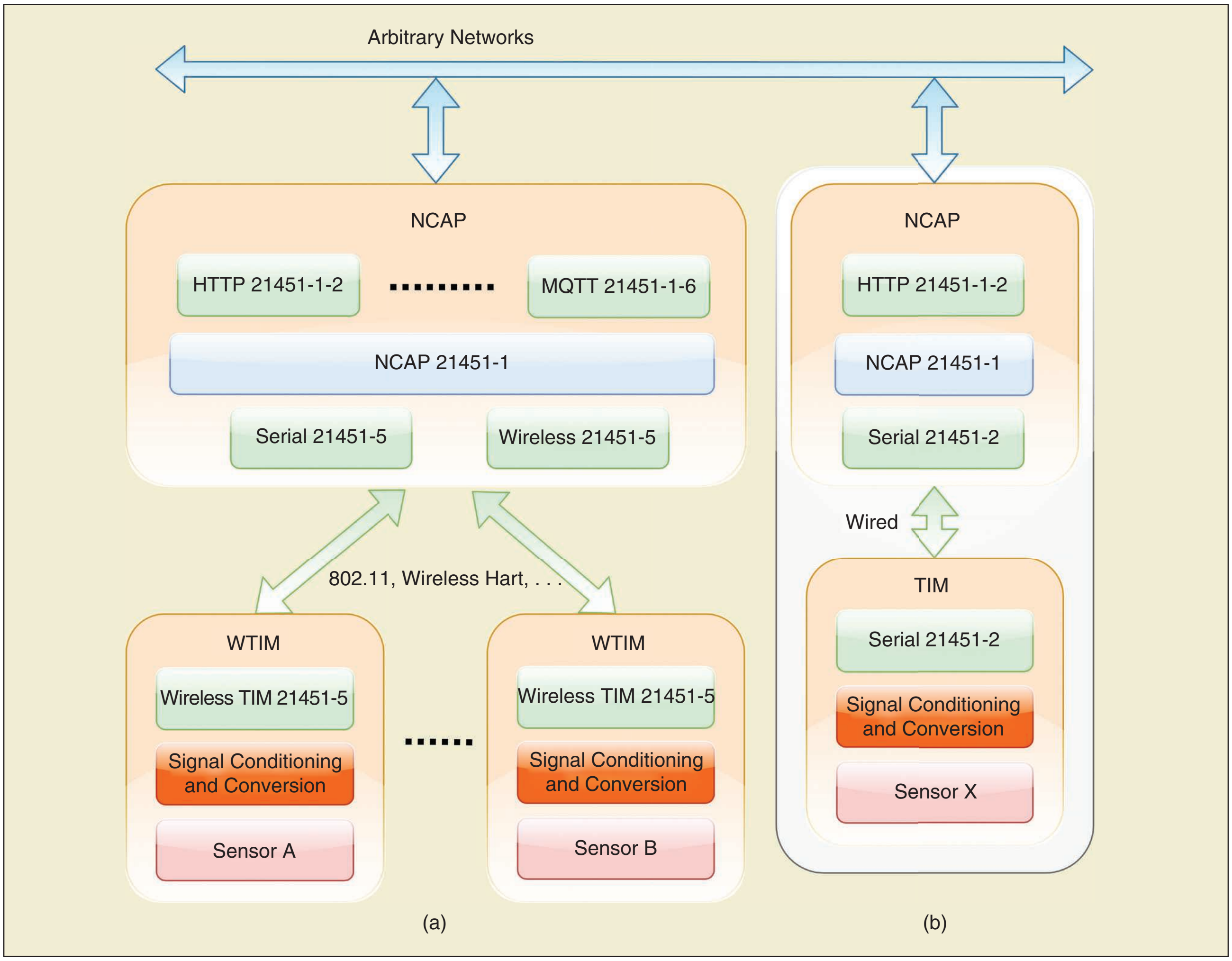


FIGURE 1 – The smart transducer communication architecture illustrating (a) the distributed sensors using wireless communication defined in ISO/IEC/IEEE 21451-5 and (b) a sensor module using wired communication defined in ISO/IEC/IEEE 21451-2.

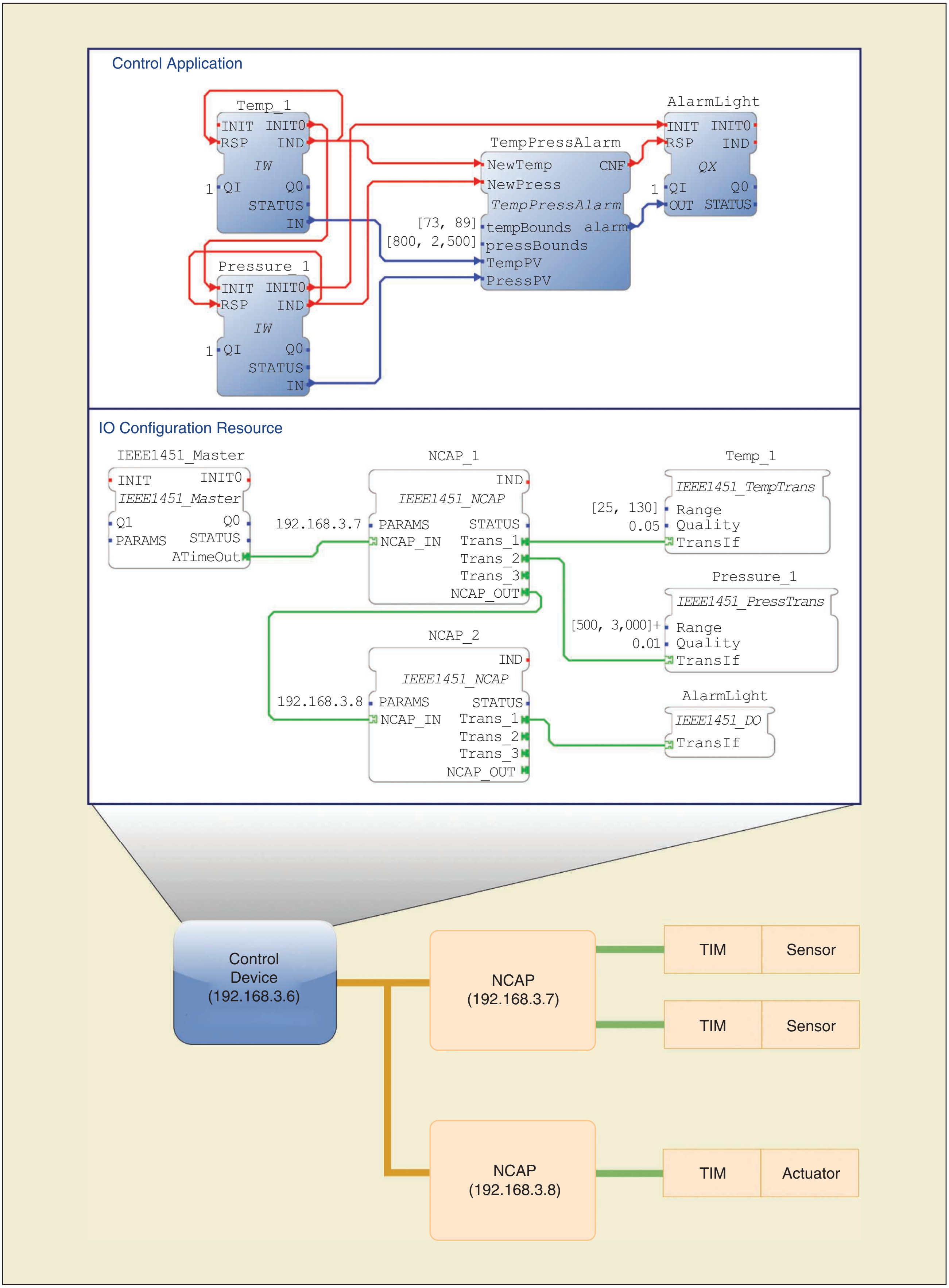


FIGURE 2 – An illustration of an example using the proposed NCAP implementation integrated in Eclipse 4diac. The details of the function block descriptions are given in [3]. PARAMS: parameters.

the network's complexity. Additionally, if the machinery or system to be controlled has a wireless TIM (WTIM), the coverage of the wireless network can be substantially extended by using more than one NCAP. To support the coverage, this NCAP-WTIM assignment that provides the best communication quality is used. This can be done without rearranging the machinery or installing cables.

An NCAP-TIM combination can be implemented as one hardware unit that provides simple access due to the reduced assignment effort when sensors are mounted. Compared to a TIM, the NCAP provides a higher variety of communication protocols, such as an Ethernet PHY layer or others, which serve as a managed network extension to the TIM.

Smart Transducers Applied to IEC 61499

Considering the typical application development and deployment lifecycle, we investigated how the ISO/IEC/IEEE 21450 standard, and specifically, TEDS can reduce the development effort and increase application quality. In IEC 61499, the starting point is the application design with a generic IO specification. This is followed by the configuration of the control hardware, including communication systems, as well as the IO subsystem. Finally, the application is deployed and executed on the control devices. During deployment and application startup, a validation step checks whether the real world meets the application requirements. In the case of validation issues, error management needs to be applied.

Application Design and the IO Perspective

Separating control application and control devices is a core concept in IEC 61499 [3], similar to other model-driven architecture approaches. It implies that the control application is first modeled independently of the controller hardware. However, as the control application requires sensors and actuators for correct operation, the interaction with them needs to be defined at this modeling stage. In related work, it has been shown that IO interaction specification can be generalized and made independent of the physical IO interfaces [17]. The generalized concept foresees generic function blocks (FBs) to define the requirements for certain sensor inputs and actuator outputs as part of the control application. The top part of Figure 2 shows a control application example for a simple temperature- and pressure-monitoring application. Two sensor values will be received by two generic FBs of type IW. The FB TempPressAlarm processes the sensor updates and generates the control output. If a certain condition is detected, an indicator light (FB QX) is turned on.

The next step in the IEC 61499 modeling methodology is to define the available control devices and how they are connected. However, IEC 61499 considers only devices that are capable of executing IEC 61499 models. IO interaction, as well as the configuration of IO subsystems, is mostly neglected in the standard. This drawback was overcome by introducing dedicated IO configuration resources in the devices [17]. In these resources, the available IOs are modeled again with dedicated FB networks, where the FBs represent the various IO devices as well as, for example, fieldbus interfaces. Adapter connections are used to define the order of the devices and the association to different fieldbus segments (the middle part of Figure 2). The FBs used can then hold parameters specific to the IO subsystem (for example, scan times and bus addresses). With unique names, a link to the generic IO FBs in the application is established. During device startup phase, this information is used to check whether the specified IO devices are available.

It is important to note that the control application is an abstract view of the control design. The IO configuration is mapped to a versatile model within the control device. To clarify this, there is no other link between the IO configuration and the application other than the identification of the sensor and actuator devices. It does not matter whether the devices are connected to a fieldbus, M-Bus, or any other hardware design. Moreover, ISO/IEC/IEEE 21451 allows smart sensors and actuators to be mapped from the hardware (lower part of Figure 2) to the IO configuration (middle part of Figure 2). Details are given in the next section.

Connect Application and Smart Sensor with NCAP and TEDS

The generalized IO configuration concept [17] was a major step toward removing IO dependence from IEC 61499 control applications. Therein, the IO configuration resources are defined on a very low level (such as single inputs and outputs, physical setup of the IO bus system, and raw sensor values). Here, we proceed to the next step and introduce the ISO/IEC/IEEE 21450 transducers as generic IO FBs that cover defining the requirements and their physical link.

Also, the sensor value mapping and interpretation is part of the standard [18]–[20]. The implementation of an application that considers real-world sensors must be designed and tailored for the specific use case with certain sensor requirements. Preferably, the application uses measurements based on calibrated samples given in SI units or normalized values. This allows the simple integration in control applications and reduces scaling or rescaling errors. In addition, timing parameters defined for the TIM enable the alignment to the requirements of the IO FB and the control application. The TEDS description allows a check on whether the used sensor fulfills the requirements given by the control engineer during design and run-time.

Here, we follow the idea that the FB for an IO device is an implementation of an NCAP or an interface to an NCAP (Figure 2). The diagram shows the great variability of the smart transducer concept. The NCAP FBs connect the required sensors and actuators and manage the communication. Each smart transducer has its own description stored on the transducer hardware, and if connected, the NCAP can simply read the information of the TEDS and store it for further processing in the digital representation of the sensor or actuator. Hence, this establishes an automatic link between the ISO/IEC/IEEE 21450 FB and the TEDS located and maintained in the transducer hardware.

As in the example of Figure 2, various sensors, wired or wireless, can be attached to one NCAP, as long as they are using the same communication protocol and PHY layer. Additionally, if there are numerous different sensors with individual communication interfaces, the NCAP devices can be used to manage the bridging to the control device. Roughly speaking, for each PHY layer and protocol combination, an individual NCAP must be used. Furthermore, if the communication range of the sensors, wired or wireless, is limited, NCAP devices can be used to bridge and extend the communication range.

TEDS As Digital Twin in Control Applications

Any ISO/IEC/IEEE 21450 device attached to an IEC 61499 application can provide its metadata. Hence, whether the application is supported by the connected physical device can be evaluated. This is concluded from a digital twin interpretation of the TEDS. The digital twin representation of a virtual TEDS is also described in IEEE 1451.4 [13]. On transducer hardware, the TEDS is linking the digital twin with the physical reality [21], greatly reducing the system configuration effort and also the value processing effort for control engineers.

Initially, we focus on the first startup of an IEC 61499 device. In a simple setup, all available NCAP devices are connected via any network to a controller. The NCAPs themselves are bound to their environment and list the attached FBs representing the TIMs. This design process can be supported by a development environment, such as 4diac IDE from the Eclipse 4diac open source project. Moreover, the details of the sensor, actuator, or event sensor can be collected from the TEDS of the physical device if the environment supports a discovery procedure of IEC 61499 devices. The control engineer benefits from knowing which IO devices are present. The automatic digital twin eases the design process because initial mapping, remapping, and the simulation are supported with the real hardware information available.

Knowing the design of the IO configuration, we can validate the mapping by an automated check to align the device and the IO throughout its lifecycle. If this mapping is invalidated by any (hard- or software) reconfiguration, automatic or semiautomatic remapping must be performed. This closes the loop from the physical device to the design process of the IO described in the previous section.

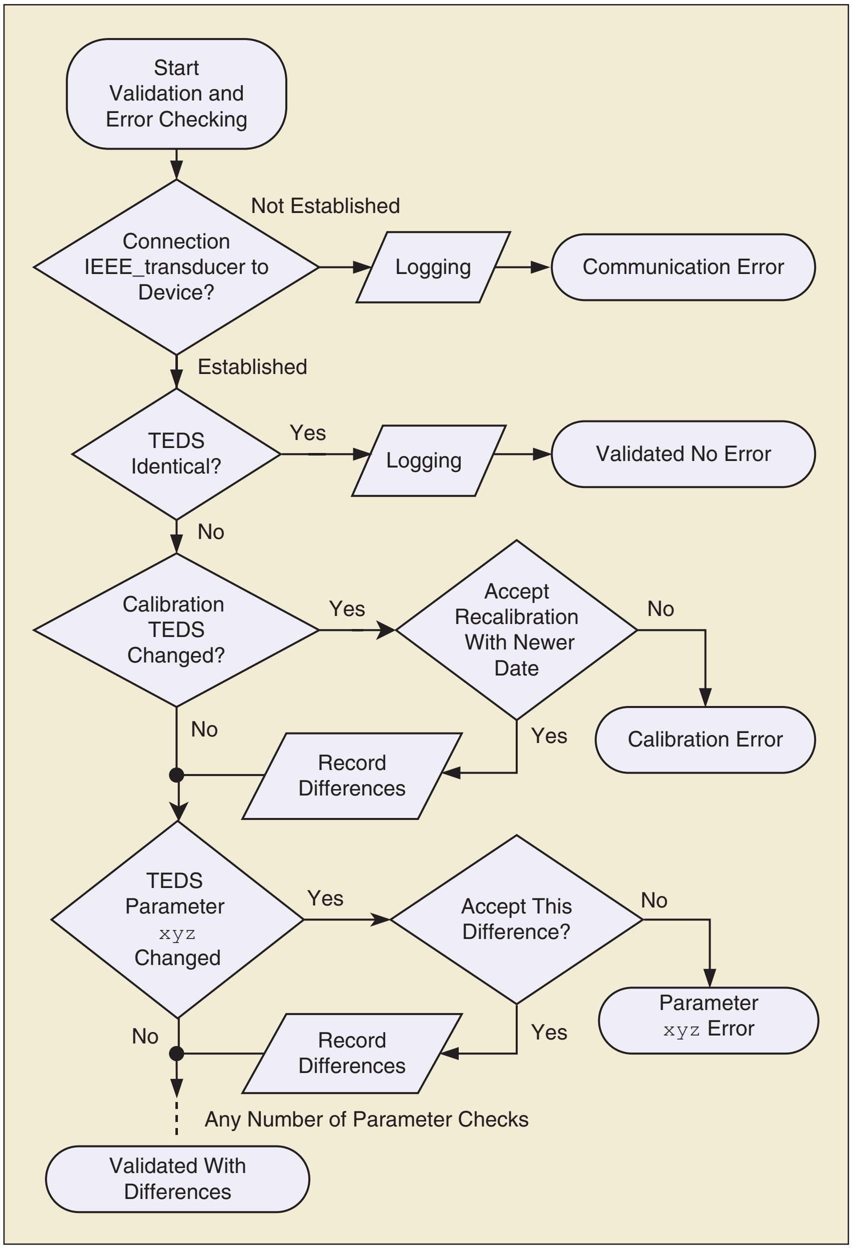


FIGURE 3 – A validation check for the correct transducer connection.

Validation, Documentation, and Error Management

The advantage of utilizing TEDS is that during the startup of the automation system, automatic validation and documentation of the required IO capabilities to the available IOs are performed (as described in the last section). In Figure 3, we show a validation check algorithm that can be executed for one or more transducers. How strictly validation errors have to be handled depends on the application. The TEDS allows for a very accurate description of the sensors, which can be used to verify whether it fulfills the requirements given by the application. Even so, a full logging can be performed that allows changes and the unchanged state to be recorded for a full documentation of the attached transducers whenever the control application has been executing.

This poses the question but also gives the possibility of how to handle such validation issues. A great advantage of the presented approach where the required IO capabilities are specified with FBs is that, during startup, these FBs can inform the application via events and data about IO validation issues. These events can now be used by application-specific error handlers modeled again in FBs to perform error handling tasks [22].

The simplest error-handling mechanisms would be to stop the application execution and inform the user. More complex mechanisms can perform regular revalidation for a certain number of times (for example, five times over a period of 5 s). Even degraded performance scenarios are envisaged, such as utilizing other sensors to compensate for a missing one. However, this compensation provides a less accurate sensor value that will result in a degraded operation, but instead of a full stop, the process can be continued.

This allows for a very fine-grained error management and an automated IO device logging. It could even be done on a subsystem basis. If the application permits, parts of the application could be operated fully, while others would be in error or in degraded operation mode.

Implementations of the NCAP

There are three different methods to utilize the NCAP concept in an IEC 61499 device. This concept allows the TIM to be connected very flexibly and handles the TEDS for the convenience of an application programmer. First, we consider a control and data bus of the IEC 61499 device, which is incompatible with the NCAP network interfaces [see, for instance, Figure 4(a)]. In this case, a bridging hardware must be used to communicate with the external NCAP. The bridging hardware does not have any special functions. It handles only the standardized communication to and from a TIM via an external NCAP.

The software representation is an NCAP FB in the IO model of the application. This FB has a list of all connected TIMs. The NCAP FB retrieves the TEDS from the connected TIMs and provides validation against the requirements of the control application. With each restart of the system or hardware change, the IO configuration and the real-world IO can be validated.

In Figure 4(b), the network interface of the NCAP is connected directly to the system bus of the control device. Also, in this case, the NCAP FB provides the handling of the TEDS to guarantee the correspondence between the modeled and real IO devices.

Finally, the most complex software requirement is a pure virtual NCAP [Figure 4(c)]. In this case, the TIMs are directly connected to the control device. To enable the full NCAP functionality in the control device, a virtual NCAP must implement the services described in the ISO/IEC/IEEE 21451-1 standard [23]. In principle, it would be possible to remove the virtual NCAP and access TIMs directly, but from a systematic perspective, it is important to stick to the standard-conform design pattern and use a virtual NCAP.

Conclusion

In this article, we introduced a promising link between distributed model-driven control software engineering and transducer-centric data sheet management by TEDS, namely ISO/IEC/IEEE 21450. In complex control applications, the configuration of sensors and actuators (such as transducers) is usually done manually. The required tasks include the assignment of the transducers to the entities in the control application, calibration, and documentation of the transducers used. These tasks are prone to errors not only because the transducers may become accidentally mixed up but also due to the delay between changing and adding the appropriate update of the documentation. Furthermore, the control application interacts with transducers on a very low level. Therefore, transducer changes may also affect the control application.

The presented approach shows how high-level requirements for transducers can be specified at the application level. Based on these requirements, an automatic mapping between available transducers and application requirements can be performed during the control system startup. Furthermore, TEDS forces the use of SI units, which unifies the transducer value processing, further decoupling control applications from the specific transducers used.

While ISO/IEC/IEEE 21451-1 defines a unified network-based access to transducers, it lacks the support for the upcoming Time-Sensitive Networking (TSN) standard maintained by IEEE 802.1 working group. This raises some open questions for future work within smart transducer standardization. Although there are some open issues, the combination of IEC 61499 and ISO/IEC/IEEE 21450 with the

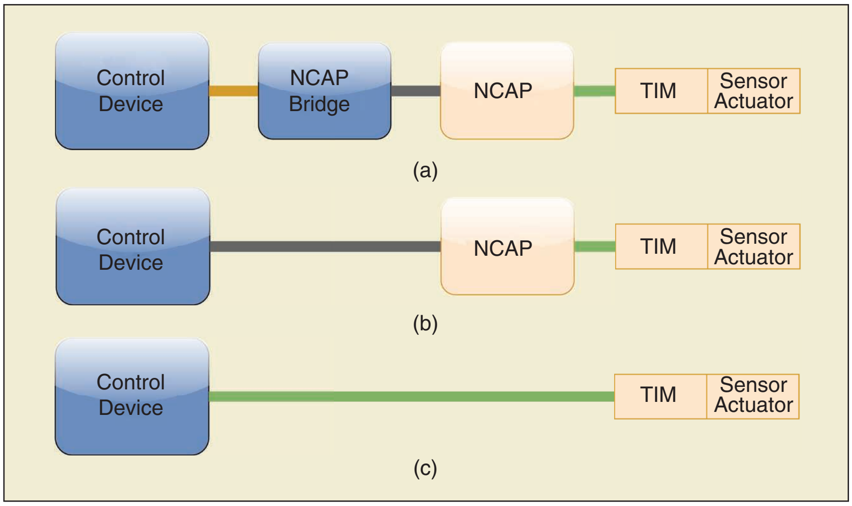


FIGURE 4 – The NCAP implementations for IEC 61499 devices. (a) An NCAP device connected via hardware interface to the control device. (b) A native NCAP is connected to the controller if the NCAP network interface matches controller. (c) A TIM is directly connected to the controller and the NCAP becomes virtual.

suggested transducer discovery process in a development environment, such as the Eclipse 4diac, allows significant progress in systematic application engineering for control systems.

The established link between the two standards is an excellent standardization concept for the emerging Internet of Things. However, two aspects might need to be addressed to become accepted at a large scale. First, the popularity of the ISO/IEC/IEEE 21450 and IEC 61499 needs to be improved. Second, for the rather complex standards, an easy-to-use framework has to be designed.

Biographies

Hans-Peter Bernhard (h.p.bernhard@ieee.org) earned his M.Sc. degree in electrical engineering and his Ph.D. degree from TU Wien, Vienna, Austria, in 1991 and 1998, respectively. He is head of the Wireless Communication Networks Research Unit at Silicon Austria Labs and senior scientist at the Institute of Communications and RF Systems at Johannes Kepler University, Linz, Austria. His research focuses on real-time wireless communication systems, factory communication in harsh environments, security and safety mechanisms for wireless communication, and energy-efficient signal processing for communication as well as energy-efficient smart transducers for the Industrial Internet of Things. He is an active member of the IEEE 1451 technical committee of the Standard for a Smart Transducer Interface for Sensors and Actuators.

Alois Zoitl (alois.zoit@jku.at) earned his M.Sc. degree in electrical engineering and his Ph.D. degree from TU Wien, Vienna, Austria, in 2002 and 2007, respectively. He is a professor in cyberphysical systems for engineering and production at the Johannes Kepler University, Linz, Austria. His research focuses on adaptive production systems, distributed control architectures, and dynamic reconfiguration of control applications as well as software development and software quality assurance methods for industrial automation. He is a founding member of the open source initiatives Eclipse 4diac, providing a complete IEC 61499 solution, and OpENer. Since 2009, he has been an active member of the IEC SC65B/WG15 for the distributed automation standard IEC 61499. He was named convenor of the group in May 2015.

Andreas Springer (andreas.springer@jku.at) earned his M.Sc. degree in electrical engineering in 1991 from TU Wien, Vienna, Austria, and his Ph.D. degree in 1996 from the Johannes Kepler University, Linz, Austria. He is a professor for communications engineering, head of the Institute for Communications Engineering and RF-Systems at Johannes Kepler University, Linz, Austria, and a research area coordinator in the Austrian K2 Center for Symbiotic Mechatronics. Since 2017, he has been co-leader of the Christian Doppler Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications. His research interests include wireless communication systems, industrial wireless sensor networks, architectures and algorithms for multiband/multimode transceivers, and molecular communications. He is participating in Secure Connected Trustable Things (SCOTT), a European project on trust in industrial wireless sensor networks with 57 partners.

References

[1] C.-H. Yang, V. Vyatkin, and C. Pang, "Model-driven development of control software for distributed automation: A survey and an approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 292–305, 2014.

[2] B. Vogel-Heuser and F. Ocker, "Maintainability and evolvability of control software in machine and plant manufacturing: An industrial survey," *Control Eng. Practice*, vol. 80, pp. 157–173, Nov. 2018.

[3] A. Zoitl and V. Vyatkin, "IEC 61499 Architecture for distributed automation: The 'glass half full' view," *IEEE Ind. Electron. Mag.*, vol. 3, no. 4, pp. 7–23, 2009.

[4] V. Vyatkin, "IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 768–781, 2011.

[5] W. Dai and V. Vyatkin, "A component-based design pattern for improving reusability of automation programs," in *Proc. 39th Annual Conf. IEEE Industrial Electronics Society (IECON)*, Nov. 2013, pp. 4328–4333.

[6] P. Gsellmann, M. Melik-Merkumians, and G. Schitter, "Comparison of code measures of IEC 61131-3 and 61499 standards for typical automation applications," in *Proc. IEEE 23rd Int. Conf. Emerging Technologies and Factory Automation*, 2018, pp. 1047–1050.

[7] M. Wenger, M. Melik-Merkumians, I. Hegny, R. Hametner, and A. Zoitl, "Utilizing IEC 61499 in an MDA control application development approach," in *Proc. IEEE Conf. Automation, Science and Engineering*, Trieste, Italy, Aug. 24–27, 2011, pp. 495–500.

[8] R. Abrishambaf, M. Hashemipour, and M. Bal, "Structural modeling of industrial wireless sensor and actuator networks for reconfigurable mechatronic systems," *Int. J. Adv. Manuf. Technol.*, vol. 64, nos. 5–8, pp. 793–811, 2013. doi: 10.1007/s00170-012-4070-y.

[9] A. Kumar, V. Srivastava, M. K. Singh, and G. P. Hancke, "Current status of the IEEE 1451 standard-based sensor applications," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2505–2513, May 2015.

[10] *IEEE Standard for Sensor Performance Parameter Definitions*, IEEE Standard 2700-2017 (Revision of IEEE Standard 2700-2014), Jan. 2018.

[11] E. F. Silva, B. J. Dembogurski, A. B. Vieira, and F. H. C. Ferreira, "IEEE P21451-1-7: Providing more efficient network services over MQTT-SN," in *Proc. IEEE Sensors Applications Symp. (SAS)*, Mar. 2019, pp. 1–5.

[12] R. Seng, K. B. Lee, and E. Y. Song, "An implementation of a wireless sensor network based on IEEE 1451.0 and 1451.5–6LoWPAN standards," in *Proc. IEEE Int. Instrumentation and Measurement Technology Conf.*, May 2011, pp. 1–6.

[13] *ISO/IEC/IEEE Information Technology—Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*, ISO/IEC/IEEE 21450:2010(E), May 2010.

[14] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE Standard 1451.5-2007, Oct. 2007.

[15] T. Mitterer, H. P. Bernhard, and H. Zangl, "Power efficient data compression and transmission in low power wireless sensors for ISO/IEC/IEEE 21450," in *Proc. IEEE Int. Conf. Industrial Cyber-Physical Systems (ICPS)*, Taipei, Taiwan, May 2019, pp. 1–6.

[16] H.-P. Bernhard, A. Springer, A. Berger, and P. Priller, "Life cycle of wireless sensor nodes in industrial environments," in *Proc. 13th IEEE Int. Workshop Factory Communication Systems*, Trondheim, Norway, May 2017, pp. 1–9.

[17] W. Eisenmenger, J. Meßmer, M. Wenger, and A. Zoitl, "Increasing control application reusability through generic device configuration model," in *Proc. 22nd IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, Sept. 2017, pp. 1–8.

[18] Z. Liu, G. Monte, and V. Huang, "ISO/IEC/IEEE P21451-001 standard for signal treatment of sensory data," in *Proc. IEEE 25th Int. Symp. Industrial Electronics (ISIE)*, June 2016, pp. 766–771.

[19] *IEEE Recommended Practice for Signal Treatment Applied to Smart Transducers*, IEEE Standard 21451-001-2017, Dec. 2017.

[20] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators–Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE Standard 1451.4-2004, Dec. 2004.

[21] E. Y. Song, M. Burns, A. Pandey, and T. Roth, "IEEE 1451 smart sensor digital twin federation for IoT/CPS Research," in *Proc. IEEE Sensors Applications Symp. (SAS)*, Mar. 2019, pp. 1–6.

[22] M. Steinegger, A. Zoitl, M. Fein, and G. Schitter, "Design patterns for separating fault handling from control code in discrete manufacturing systems," in *Proc. 39th Annu. Conf. IEEE Industrial Electronics Society (IECON)*, 2013, pp. 4368–4373.

[23] *ISO/IEC/IEEE Information Technology—Smart Transducer Interface for Sensors and Actuators – Part 1: Network Capable Application Processor (NCAP) Information Model*, ISO/IEC/IEEE 21451-1:2010(E), May 2010.