Alexander Schlaefer, Sibylle Schupp, Andre Stollenwerk (Hrsg.)
2nd Workshop on Fail Safety in Medical Cyber-Physical Systems, Wien 2016

Sensor Supervision and Control Value Limitations in Networked Intensive Care

Jan Kühn[1] André Stollenwerk[1] Christian Brendle[2] Thorsten Janisch[3] Marian Walter[2]
Rolf Rossaint[4] Steffen Leonhardt[2] Stefan Kowalewski[1] Rüdger Kopp[3]

Abstract: This paper presents an example of use for an embedded software architecture in an automated intensive care setup. The communication structure allows to handle sensor faults and control value limitations. Decentralized sensor and actuator supervision is realized by a network of embedded systems, which can be adapted to the requirements of different medical setups. It is implemented in a setup of medical devices for autonomous veno-venous extracorporeal lung assist.

Keywords: MCPS, software architecture, sensor fault detection

1 Introduction

There is a trend of clinical setups towards systems which provide patient-dependent assistance and automation of controllable processes. This development is also influenced by achievements in the fields of networked *cyber physical sytems* (CPS) and distributed systems within other safety critical industries, like the automation industry and automotive industry. Nevertheless, improvements from other technological fields are often not applied in medicine, due to several reasons. Whereas proving sufficient security is the main issue in e-Health, assuring safety is the main problem for further automation of treatments in intensive care. However, the development process suffers from weak or missing regulatory guidance and legal certainty.

The development of software for medical devices and applications in Germany is regulated by the law for medical products, which is the national realization of the European directive 93/42/EWG [MDD07]. Furthermore, it is guided by the nationally adopted standards DIN EN 60601 [DI07] and DIN EN 62304 [DIN09]. Especially, the interconnection of medical devices as necessary basis for further automation lacks of a detailed description.

[1] RWTH Aachen University, Informatik 11 - Embedded Software, Anschrift, 52056 Aachen, kuehn@embedded.rwth-aachen.de
[2] RWTH Aachen University, Philips Chair for Medical Information Technology, 52056 Aachen
[3] RWTH Aachen University Clinic, Department of Intensive Care and Intermediate Care, 52056 Aachen
[4] RWTH Aachen University Clinic, Clinic for Anesthesiology, 52056 Aachen

1.1 Clinical Automation

Several approaches for software architectures handling the interconnection of medical devices have been proposed without leading to a widely applied standard in the industry with the desired level of detail. But a lot of work on this field is in progress. Promising examples towards successful standardization are the IEEE 11073 standards [Ka15],the OR.NET related projects [KL14] [Mi15] and the MDPnP project [Ha12] [PAG14]. Not many implemented examples were extensively tested in safety critical treatments with a high level of automation. Therefore experiences with applications under realistic conditions are highly valuable.

Successful standards regarding the software for distributed systems is rather long established in other industrial fields. These have been developed respecting common demands for safety. With the upcoming automation of other major industries and the resulting need for networked distributed systems appropriate reference architectures were developed. There are several examples for ongoing standardization processes regarding this topic like a published reference architecture for the Internet of Things [Ha15]. The AUTOSAR (AUTomotive Open System ARchitecture) reference architecture was developed by the automotive industry and is widely used. Functional safety of the distributed systems is a central goal of it [Fü09].

The architecture presented in this paper is comparable to AUTOSAR regarding the structure of the communication management. It has been demonstrated to be robust and easily manageable for over four years of application in many in vitro and in vivo experiments for closed-loop control in intensive care [St11]. It is focused on Inter-ICU (Intensive Care Unit) equipment communication for therapy automation. The direct interaction with the clinical network is not desired and therefore not addressed here, due to arising security issues. The network is realized with the use of cost-efficient embedded hardware, designed for the interaction with a wide range of medical devices.

2 An Embedded System for Clinical Automation

2.1 Clinical Setup

A highly automated clinical setup for the treatment of severe *acute respiratory distress syndrome* (ARDS), a life threatening disease, is in focus of our research. The setup is shown in Fig. 1. The setup includes an automated veno-venous *extracorporeal lung assist* (ECLA) and a mechanical ventilation combined by a global control strategy [Br15]. Usually, the therapy follows a treatment strategy based on an established protocol. In contrast to this, the automation allows the realization of a desired gas transfer rate with respect to other goals like minimization of blood trauma. Several sensors are required to allow the calculation of the patients' state.

The network is based on a self-developed microcontroller node which provides different interfaces and allows handling the medical device as a distributed system in our network.

2.2 Networked System of Medical Devices

The setup consists of several monitoring devices and three actuators, a gas blender (prototypical), a bloodpump (Medos DP2) with a controller (prototypical) and a mechanical ventilator (Siemens Servo 300). Most of the devices provide different in- or outputs (IOs), usually analog or digital ports ranging from UART over USB to Ethernet. For the sake of simplicity we use the available analog and serial ports to connect the devices via the self-developed microcontroller nodes (ASMOs) over CAN bus. A defective node of the network can be exchanged easily by switching it of and using a backup node without communication interrupt. A central controller application runs on a dSpace MicroAutobox, a real time capable platform for rapid prototyping, which is also connected to the CAN bus. A defective controller can also be exchanged. Nodes responsible for actuators assure a safe state.

The ASMO nodes (Fig. 1) are based upon 32bit Atmel ARM SAM7 microcontrollers and a set of onboard peripherals like CAN bus, UART, AD/DA converters and several digital ports, which allow the use of situation based hardware components [St11]. All relevant sensor and actuator data is provided via CAN bus as integer value with synchronized time stamp. Data update intervals are usually based on the data rate of the medical device.

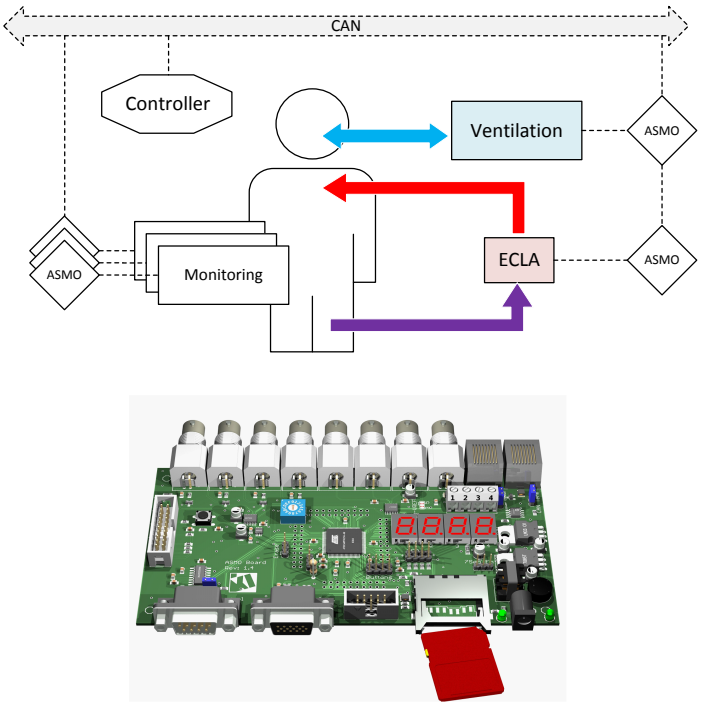

Fig. 1: Automated ARDS therapy setup and ASMO node hardware design [St13]

2.3 Software Architecture

The software architecture in use is based on the open source operating system ChibiOS/RT. This is a real time operating system for embedded systems with focus on a compact kernel size and fast execution times. The structure used here is presented in Fig. 2. Each layer interacts with the layers above and below. Other possible interactions between the layers via dedicated APIs are not depicted in detail. The lower part, which is responsible for the hardware interaction, is the *basic software layer* (BSW). On the left side is the *real time operating system* (RTOS) ChibiOS, which provides the structure for application management like real-time capable thread scheduling and interrupt handling. It is accompanied by a *hardware abstraction layer* (HAL), where several *low level drivers* (LLD) were added for explicit hardware driving on the microcontroller nodes. Obviously, these two layers must interact closely with each other, which is explicitly shown in this case. The high level drivers mainly consist of the *communication layer* (COM) with the CAN bus driver and the *in-/output layer* (IO). The IO level has drivers for analog and digital signals like UART and SPI used to get data from sensors and actuators. Overlaying the communication layer is the *data provisioning layer* (DPL). It coordinates the data exchange between applications and the COM layer. The COM layer can be extended by other communication drivers, like drivers for communication via Ethernet or FlexRay bus. A part of the DPL is the *wrapping layer* (WL), which allows the fast integration of model based software development. In this case it is realized for code of models generated by the Simulink Coder (The MathWorks, Inc.). The interface for DPL and WL interaction is automatically generated. Dedicated Simulink blocks are provided to implement the data access on the level of the model. The DPL and the WPL are described in detail in [St11]. Also part of the DPL is the *safety layer* (SL), which was tested as example on the application level for the mechanical ventilator. The integration in the DPL will be described below. Above it is the *application layer* (AL) with the *medical device applications* (MDAs) and the *model based applications* (MBAs). They realize the interconnection of the different devices by fulfilling controller and supervisor tasks for medical actuators or providing the data of medical sensors over CAN. An example is given in [St15]. Complex control or supervision tasks are usually done by MBAs, which use the WL introduced above. Advantages are a reduced implementation effort for complex tasks like adaptive control algorithms, and the possibility of testing, analyzing and verifying on model level. An example is given in [St14]. Since all data provided or required by an application are automatically managed by the DPL, the applications can be distributed arbitrarily on the ASMO nodes. This is only restricted by limited hardware resources, like memory or peripheral device access. A feature of ChibiOS is its static kernel implementation. This allows to realize a static implementation by following appropriate coding guidelines. The advantage is the higher applicability and therefore more precise results of static analysis techniques like a stack size calculation. Since the architecture was focused on cost-efficient embedded solutions and a static implementation, it has the according characteristic. The fast and small implementation might be less extensive in features and less flexible than other approaches. The advantages and disadvantages of such a solution are well known: it is not feasible to update software parts online, but simple to exchange a device by an updated one. The compact code size reduces the effort for testing and the overall reliability of developed code.

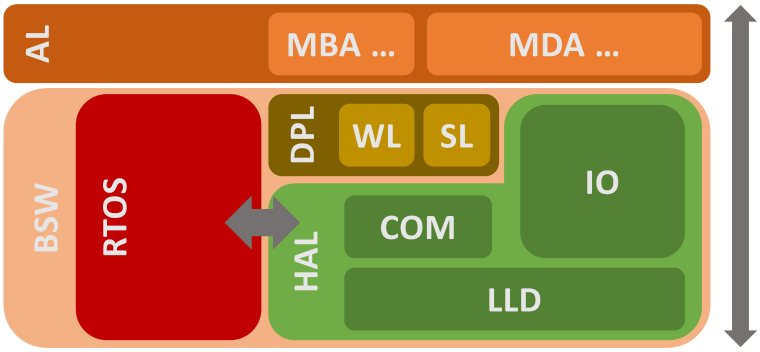

Fig. 2: Software architecture in ECLA-VENT

3 Safety Issues

Networking medical cyber-physical systems allows to work with complex control strategies or determine the patients state more exactly by personalized physiological models. But higher levels of interconnection and automation come with a tradeoff of increasing complexity and new dependencies. The additional functionality goes along with new possible faults due to new dependencies like it was discussed in [Kü15]. Techniques to communicate and handle such faults are discussed in the following sections.

3.1 Sensor Faults

In intensive care setups most devices are still not connected. Medical staff has to interpret the sensor data and supervise the sensor states manually. The data is used to make adjustments of the therapy parameters to improve the outcome. The medical staff can be assisted by sensor fusion to give additional data from soft sensors or to increase the precision of sensor supervision like in [Me11]. There have been numerous works with the focus on sensor fault detection [CP12]. Soft sensors can be simply implemented as applications in the architecture (Fig. 2). An example is the calculation of the so called stress index of a mechanical ventilated lung, where the calculation is based upon pressure and airflow measurements [Fe15]. In our case, it is not part of the output of the mechanical ventilator itself but can be calculated because it is an indicator how invasive the ventilation is. In contrast to sensor fusion, sensor validation is always a crucial part of safety critical systems. As a result the sensor has to notify the depending applications of its state. As this can be a rather complex task, it could also be achieved by a separate application. The safety layer can provide a sensor state signal, which is element of $\{safe; uncertain; fault\}$. Dependent applications can react on a sensor fault. If a sensor is not validated, its state is *uncertain*. The reaction of a controller depends on the implementation. Without valid sensor data, fallback strategies can be implemented, like freezing control values in the present state or a safe default state, which usually has to be predefined fault situation dependent.

3.2 Actuator Faults

The sensor data is the basis for the automation of a system. In a distributed system like the presented intensive care setup, the control signals are limited by physiological, technical or safety constraints. The data flow principle is shown in Fig. 3. Control values and limitations can be sent via CAN or directly provided by the DPL. The safety layer allows only control values in the given limitations. The control value limitations can change due to problems like the further restriction of an actuator output due to malfunction like it was presented in [St15]. An additional task of the safety layer is the reaction to the unvalidated control value, like the nearest possible value and communicating the validation result to all dependent applications. The correct operation in form of the realization of the desired control values is a supervision task and has to be implemented on the application layer. A general solution, for handling control value limitations on the controller level, cannot be given, because it strongly depends on the nature of the controller. A PID controller implementation can be adapted by adjusting the anti-windup, but in a more complex or even nonlinear controller the handling is far more demanding or even not possible.

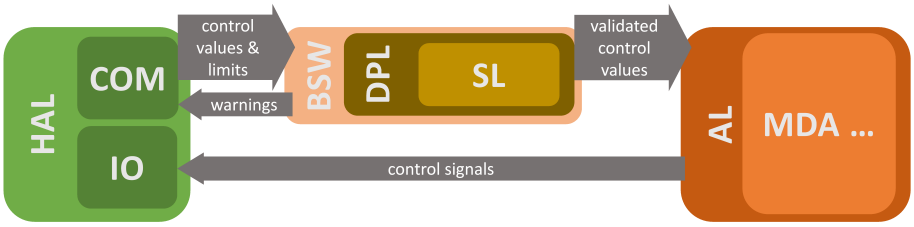

Fig. 3: Safety layer in ECLA-VENT

4 Conclusion and Outlook

A safety layer concept regarding a software architecture for distributed embedded systems in an intensive care application was presented and discussed. It is focused on embedded systems that allow intelligent sensors and actuator communication. It provides the corresponding positive an negative characteristics. The implementation can be analyzed more easily for correct implementation, but is also less flexible compared to other approaches. It provides sensor states as supervision and fault detection results. Further, it is responsible to ensure the correct operation of actuators by control value validation and limitation. The validation results have to be communicated to allow handling the limitation. Additionally, it must be possible to adapt the limits to allow a reaction on certain events. Open research topics are developer assistance by templates for different sensor fault detection methods and controller adaption to different fault states. Furthermore, techniques for handling unstable controller behavior should be implemented, for example by choosing a situation dependent stable state.

Acknowledgements
This work was supported by the German Research Foundation DFG (DFG - Grant PAK 138/2). The authors gratefully acknowledge this allowance.

References

[Br15] Brendle, C.; Hackmack, K.-F.; Kühn, J.; Wardeh, M. N.; Kopp, R.; Rossaint, R.; Stollenwerk, A.; Kowalewski, S.; Misgeld, B.; Leonhardt, S.; Walter, M.: In silico evaluation of gas transfer estimation during extracorporeal membrane oxygenation. In: 9th IFAC Symposium on Biological and Medical Systems. 2015. To appear.

[CP12] Chen, J.; Patton, R. J.: Robust model-based fault diagnosis for dynamic systems, volume 3. Springer Science & Business Media, 2012.

[DI07] DIN, EN: , 60601-1 (VDE 0750-1): 2007-07 Medizinische elektrische Geräte–Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale, 2007.

[DIN09] DIN EN 62304. Berichtigung 1: Medizingeräte-Software - Software-Lebenszyklus-Prozesse. Mai 2009., 2009.

[Fe15] Ferrando, C.; Suarez-Sipmann, F.; Gutierrez, A.; Tusman, G.; Carbonell, J.; Garcia, M.; Piqueras, L.; Compan, D.; Flores, S.; Soro, M. et al.: Adjusting tidal volume to stress index in an open lung condition optimizes ventilation and prevents overdistension in an experimental model of lung injury and reduced chest wall compliance. Critical Care, 19(1):9, 2015.

[Fü09] Fürst, S.; Mössinger, J.; Bunzel, S.; Weber, T.; Kirschke-Biller, F.; Heitkämper, P.; Kinkelin, G.; Nishikawa, K.; Lange, K.: AUTOSAR–A Worldwide Standard is on the Road. In: 14th International VDI Congress Electronic Systems for Vehicles, Baden-Baden. volume 62, 2009.

[Ha12] Hatcliff, J.; King, A.; Lee, I.; Macdonald, A.; Fernando, A.; Robkin, M.; Vasserman, E.; Weininger, S.; Goldman, J. M.: Rationale and architecture principles for medical application platforms. In: Cyber-Physical Systems (ICCPS), 2012 IEEE/ACM Third International Conference on. IEEE, pp. 3–12, 2012.

[Ha15] Haller, S.; Ho, E.; Jardak, C.; Olivereau, A.; Serbanati, A.; Thoma, M.; Walewski, J.: , Internet of Things-Architecture IoT-A Deliverable D1. 3–Updated reference model for IoT v1. 5, http://www.meet-iot.eu/deliverables-IOTA/D1_3.pdf, 01.12.2015, 2015.

[Ka15] Kasparick, M.; Schlichting, S.; Golatowski, F.; Timmermann, D.: New IEEE 11073 standards for interoperable, networked point-of-care Medical Devices. In: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE. IEEE, pp. 1721–1724, 2015.

[KL14] Kühn, F.; Leucker, M.: OR. NET: Safe Interconnection of Medical Devices. In: Foundations of Health Information Engineering and Systems, pp. 188–198. Springer, 2014.

[Kü15] Kühn, J.; Schoonbrood, P.; Stollenwerk, A.; Walter, M.; Brendle, C.; Wardeh, M. N.; Rossaint, R.; Leonhardt, S.; Kowalewski, S.; Kopp, R.: Safety Conflict Analysis in Medical Cyber-Physical Systems using an SMT-Solver. In: Fail Safety in Medical Cyber-Physical Systems. volume 1337. CEUR Workshop Proceedings, pp. 19–23, 2015.

[MDD07] MDD, Medical Devices Directive, Council Directive 93/42/EEC of 14 June 1993 concerning medical devices, OJ No L 169/1 of 1993-07-12, changed according to directive 2007/47/EC of 25 September 2007, 2007.

[Me11] Medjahed, H.; Istrate, D.; Boudy, J.; Baldinger, J.; Dorizzi, B.: A pervasive multi-sensor data fusion for smart home healthcare monitoring. In: Fuzzy Systems (FUZZ), 2011 IEEE International Conference on. IEEE, pp. 1466–1473, 2011.

[Mi15] Mildner, A.; Janß, A.; Dell´Anna-Pudlik, J.; Merz, P.; Leucker, M.; Radermacher, K.: Development of Device-and Service-Profiles for a Safe and Secure Interconnection of Medical Devices in the Integrated Open OR. In: Risk Assessment and Risk-Driven Testing, pp. 65–74. Springer, 2015.

[PAG14] Plourde, J.; Arney, D.; Goldman, J. M.: OpenICE: An open, interoperable platform for medical cyber-physical systems. In: Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on. IEEE, pp. 221–221, 2014.

[St11] Stollenwerk, A.; Göbe, F.; Walter, M.; Kopp, R.; Arens, J.; Kowalewski, S.: Smart Data Provisioning for Model-Based Generated Code in an Intensive Care Application. In: High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS/MDPnP 2011), Chicago, USA. Upenn, pp. 0–8, 2011.

[St13] Stollenwerk, A.: Ein modellbasiertes Sicherheitskonzept für die extrakorporale Lungenunterstützung. Dissertation, Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen, Juli 2013. AIB-2013-7.

[St14] Stollenwerk, A.; Kühn, J.; Brendle, C.; Walter, M.; Arens, J.; Wardeh, M. N.; Kowalewski, S.; Kopp, R.: Model-based supervision of a blood pump. In: 19th World Congress of the International Federation of Automatic Control. pp. 6593–6598, 2014.

[St15] Stollenwerk, A.; Kühn, J.; Walter, M.; Brendle, C.; Wardeh, M. N.; Rossaint, R.; Leonhardt, S.; Kowalewski, S.; Kopp, R.: Software-based Prediction of Cannula Occlusion during Extracorporeal Blood Circulation through Networked Medical Data. In: Fail Safety in Medical Cyber-Physical Systems. volume 1337. CEUR Workshop Proceedings, pp. 1–6, 2015.