Mechanism for Safe Remote Activation of Networked Surgical and PoC Devices using Dynamic Assignable Controls*

Martin Kasparick[1], Max Rockstroh[2], Stefan Schlichting[3], Frank Golatowski[1], and Dirk Timmermann[1]

Abstract— **The number of devices within an operating room (OR) increases continuously as well as the complexity of the complete system. One key enabler to handle the complexity is an interoperable and vendor independent system of networked medical devices. To build up such an interoperable system we use the proposed IEEE 11073 SDC standards (IEEE P11073-10207, -20701, -20702) for networked point-of-care (PoC) and surgical devices. One of the major problems within the OR is that typically every device has its own control unit. This leads to unsatisfying situations like a high number of foot switches that causes operating errors or the problem that the physician cannot reach the control unit of the device where parameters have to be changed or an activation should be triggered. Dynamically assignable controls will solve these problems. This paper describes mechanisms that allow a safe remote activation of safety critical device functionalities based on a potentially unsafe off-the-shelf network with problems like connection loss and jitter. The proposed systems is based on a periodic reactivation of the device functionality and the additional use safety related information that is included into the activate operation command. The main advantage is that all described mechanisms make use of the self-description capability provided by the IEEE 11073 SDC. This enables a real interoperability and plug-and-play functionality because both the medical device and the control client do not need any a priori knowledge about each other.**

I. Introduction

The device ensembles within the operating room (OR) become more and more complex. This can only be handled with an interoperable and vendor independent interconnection between the medical devices. An emerging aspect is the dynamical interconnection between controls (like foot switch, handhold switch, touch based control units) and medical devices. The association can be changed during the operation according to the surgical workflow. For example this shall reduce the number of foot switches within complex operations which will increase patient's safety.[1]

The three proposed standards that are grouped together under the name IEEE 11073 SDC as part of the IEEE 11073 family of standards realize such an interoperable interconnection of medical devices within OR and clinic: IEEE P11073-10207, -20701, and -20702. The interconnection will be realized based on standard networks, e.g. Ethernet or WiFi. This means that there are the well-known problems like latency, jitter, packet loss, etc. well as the possibility of a total connection loss. Especially for a remote activation of the functionality of a surgical device like tuning on and off a surgical shaver or high frequency (HF) device this is very safety critical. In this paper we present mechanisms for a safe interconnection between controls and medical devices for a remote activation of the device functionality. The mechanisms are based on the principle of periodic retriggering of the activation under the usage of safety related information that is included into the activate operation command. (The activate operation command is the IEEE 11073 SDC command that triggers a defined activate operation at the device.)

For the delimitation of use cases the proposed system can be used, we present a classification of device functionalities to distinguish devices concerning their safe device functionality states. Other safety critical aspects like providing the current configuration of the dynamically assignable controls for the physician and the way of configuring the assignment are out of scope of this paper.

II. State of the Art

A. IEEE 11073 SDC Interoperability Standard Proposals

The three proposed IEEE 11073 SDC (System and Device Connectivity) standards define mechanisms for a dynamic, interoperable, and vendor independent interconnection of networked medical devices for the OR and clinic. The system is based on the principles of a service-oriented architecture (SOA). IEEE 11073 SDC consists of IEEE 11073-10207 (Domain Information & Service Model for Service-Oriented Point-of-Care Medical Device Communication), IEEE 11073-20702 (Medical Devices Communication Profile for Web Services), and IEEE 11073-20701 (Service-Oriented Medical Device Exchange Architecture & Protocol Binding).

We will describe some basic aspects of these standard proposals which are necessary to understand the mechanisms described in this paper. A detailed description can be found in [2] and [3]. IEEE 11073-10207 defines the domain information and service model and is derived from the classical IEEE 11073-10201. The Medical Device Information Base (MDIB) stores the device capability description and the device state. The device description is modeled as a tree hierarchy. The leaves are called metrics and represent the measurements, parameters, and settings of the devices. The remote control capabilities are defined within the Service

*This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the OR.NET project.

[1]Author is with the Institute of Applied Microelectronics and Computer Engineering, University of Rostock, 18119 Rostock, Germany firstname.lastname@uni-rostock.de

[2]Author is with the Innovation Center Computer Assisted Surgery (ICCAS), University of Leipzig, 04103 Leipzig, Germany firstname.lastname@medizin.uni-leipzig.de

[3]Author is with Center of Competence System Solutions - System Architecture of Drägerwerk AG & Co. KGaA, 23558 Lübeck, Germany firstname.lastname@draeger.com

Control Object (SCO) where several operations can be defined. E.g. set operations to modify a metric value or activate operations that trigger device functions of arbitrary complexity. Examples for activate operations are simple in-/decrease of parameters or the activation of device functionalities like turning on and off the rotation of a shaver or the power emission of a HF-device. The corresponding command for triggering is called activate operation command. The defined operations are available via services and can be used by other network participants that implement client functionalities.

IEEE 11073-20702 Medical DPWS (MDPWS) defines extensions and restrictions due to medical safety issues of the Devices Profile for Web Services (DPWS) which is an implementation of the SOA paradigm for embedded and resource constrained devices. The key extension for this paper is the so called safety context. It allows the device to define and advertise the requirement that the client has to include additional safety relevant contextual information into the header of an operation command. E.g. the device can require that the client has to include the state version into the safety context. If this information is not included or the value does not match to the expected value the device will reject the operation command. Note that the requirement is advertised by the device at runtime. So it is not necessary that this information is previously available for the client.

B. *Safe Remote Activation*

As already mentioned the mechanisms described in this paper aim at communication based in standard commercial off-the-shelf network infrastructure. Today's solutions for a safe remote activation are based on specialized fieldbuses like CAN, Profinet, or EtherNET/IP. These solutions lack of flexibility (e.g. no plug-and-play capability) and scalability. Additionally fieldbuses use typically specialized hardware. The usage of fieldbuses leads to a double wiring within the OR that should be avoided as far as possible. Systems for a safe remote activation based on standard network infrastructure especially using the IEEE 11073 SDC have not been described yet. Nevertheless the basic principles are established for safety related communication. For example the IEC 61784-3 [4] that deals with functional safety communication and the test principles for safety relevant communication GS-ET-26 [5] describe mechanisms like incremented message IDs or periodic resend of messages and commands.

III. Safe Remote Activation for Networked Surgical and PoC Devices

A. *Safe Device Functionality States*

To develop a system for safe remote activation it is necessary to define what the characteristic of a safe state of the medical device functionality is. In our context activation means that for example the motor of a shaver or a pump gets started or power is emitted at the instrument of an ultrasonic or radio-frequency surgical device. If the use case of the device functionality is to provide a destructive effect the safe state is to turn off the activation if an error occurs. According to the safety requirements defined in the IEC 61800-5-2:2007 [6] the safety functions “safe stop 1 or 2” or “safe torque off” will be initiated to stop the activation. In contrast there are devices like a lung ventilator where the activation has to continue. The third class is built by devices where it is necessary to start and stop the activation at any time, for example the coagulation function of a HF-device. Thus these three classes can be distinguished:

- **Class 1**: Safe State: Off
- **Class 2**: Safe State: On
- **Class 3**: On and Off are reachable states at any time

From the point of view of a remote activation class 1 and 2 are equivalent because a failure has to be detectable but the reaction to the failure is part of the device logic and not part of the remote activation mechanism. For clarity and simplification we concentrate on device functionalities of class 1 in this paper but the mechanism is analog valid for class 2. Note that the described mechanism cannot be transferred directly to devices of class 3 where the reliability of the network has to be guaranteed.

B. *Requirements*

This paper describes a concept for safe remote activation of device functionalities of class 1 and 2. The primary focus is on connections where the activation of the device ends after a certain period of time if an error occurs. This includes a complete connection loss between the device and the control as well as problems that occur in case of high latency or jitter. Jitter is defined as the variation of packet delay. In the case of a high jitter the transmission time of packets is very diverging, e.g. caused by network bottlenecks. Thus, it can occur that the retrigger activation commands send by a pressed control do not arrive at the device that should be activated within the defined time limit. In this case the device will stop its activation in order to pass into the safe state. The physician will recognize this behavior and will typically release the control. For example many surgical cutting devices, like bone knifes, stop their activation in case of overload states that can be caused by too much pressure exerted to the device by the physician. In this case the foot switch has to be released before the device can be reactivated again. After an indefinite period of time the packages containing the activation commands from the control will arrive at the device. If this happens the device must not start its activation because it can happen that at this point of time no physician presses the control. Thus, it is required that a safe system is able to identify activation commands that arrive too late at the device and has to be able to distinguish from a new activation that can possibly arrive very short after a former activation ended. So these requirements can be derived for the presented mechanisms:

1) Stop activation in case of connection loss between control a device after a defined activation duration.
2) Do not start the device activation multiple times caused by one activation sequence if the network jitter is higher than the activation duration of the device.

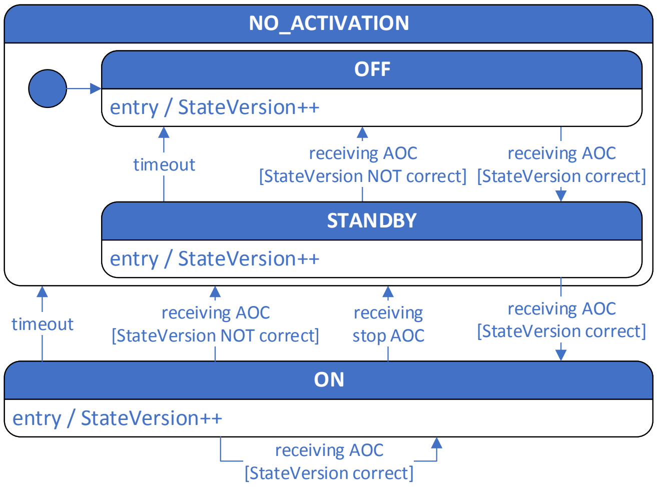
NO_ACTIVATION
OFF
entry / StateVersion++
timeout
receiving AOC [StateVersion NOT correct]
receiving AOC [StateVersion correct]
STANDBY
entry / StateVersion++
timeout
receiving AOC [StateVersion NOT correct]
receiving stop AOC
receiving AOC [StateVersion correct]
ON
entry / StateVersion++
receiving AOC [StateVersion correct]

Fig. 1. Statechart modeling the activation of the device functionality (Shortcut: AOC = Activate Operation Command)

C. Mechanisms to Ensure Safe Remote Activation for Networked Surgical and PoC Devices

Requirement 1 can be solved by a simple periodic reactivation of the device activation. The device describes the required period within which the reactivation has to be done by the control. This will be done within the description of the activate operation that is part of the device description. The *ActivateOperationDescriptor* has two parameters for this purpose: The attribute *ActivationDuration* that defines how long the activation will take place and the attribute *Retriggerable* that has to be true to allow a continuous device activation if an activate command arrives before the activation duration expired.

The second requirement cannot be fulfilled using the simple periodic reactivation mechanism. It has to be ensured that the device can distinguish between two situations: 1) interruption of the periodic reactivation caused by network jitter, 2) release and renewed actuation of the control. This is achieved by modeling the activation of the device functionality as a state chart with three states: *OFF*, *STANDBY*, and *ON* where the first two states are sub states of the parent state *NO_ACTIVATION*. The state chart is illustrated in Fig. 1. The representation of these states can be implemented as an enumeration string metric of the medical device. The states can also be mapped to the activation state of the device component supplying the functionality. This metric or the device component is used as the operation target of the activate operation. The state of this target includes a state version counter that is incremented whenever the state changes. This also includes an increment of the state version every time the activate operation is retriggered even if the value does not change and stays "on". By using the MDPWS safety context the device can require that the client has to send the target state version number within every activate operation command.

The activation process is illustrated in Fig. 2. When the first activate command has been received and processed by the device the state changes from *OFF* to *STANDBY*. The activation of the device functionality does not start at this moment. Analogous the state of the target element is set to the value "standby". This leads to an increment of the target state version from n to $n + 1$. The control client gets the new value and the new state version by eventing mechanisms. For the next activation operation command the client includes the state version $n+1$ into the safety context. If this activation operation command (which is the second command of this sequence) arrives at the device within the defined activation duration time the state *ON* is entered and the physical activation of the device is started, e.g. the surgical shaver starts operating.

The implementation of the state *STANDBY* is necessary to ensure the jitter safe behavior for the beginning of the device functionality activation. Let us assume the following situation: The control is pressed but the first activate operation command has a high latency. This could lead to the situation that the physician has yet released the control when the device receives the command and the device functionality is triggered. To avoid this unsafe situation we introduced the state *STANDBY*. The device functionality is not triggered in this state. By introducing this intermediate state the device is able to measure whether the activate operation command that actually triggers the activation of the device functionality arrived within the defined activation duration.

The further activate operation commands retrigger the activation of the device if they arrive at the device within the defined activation duration period. Every time the activate operation command is processed the target state version is incremented from n to $n+1$. So the next activate command has to include the state version $n+1$ within its safety context.

The second case is that a retrigger activate operation command including the state version n within the safety context arrives at the device after the activation duration has timed out (see middle part of Fig. 2). This means that the device functionality activation state changes from *ON* to *OFF*. This transition leads to an increment of the state version from n to $n + 1$. The retrigger activate operation command that arrived too late due to network latency contains the state version n within its safety context. This causes a mismatch between received state version n and actual state version $n+1$. The control client is informed about the rejection of its activate operation command by an *OperationInvokedReport* containing the *InvocationState* "failed" and a corresponding error message. Additionally it is recommended to implement a technical alert to propagate the timeout to other network participants that might be interested in the information like display- or logging-units.

If a control sends a new activate operation command after the activation has timed out the activate operation command contains the correct state version $n + 1$ within its safety context. So a new sequence starts (see lower part of Fig. 2). The system enters the state *STANDBY* and the activation of the device functionality is triggered by the next activate operation command. The described mechanism allows to distinguish between an activate operation command

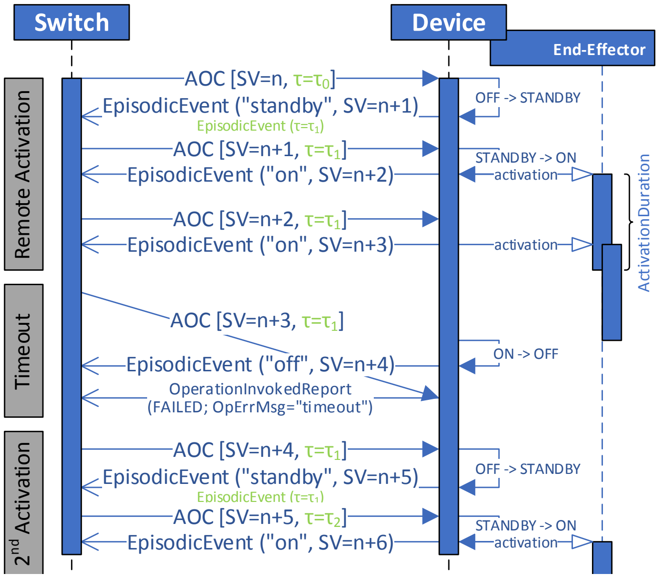

Fig. 2. Sequence diagram of the remote activation process incl. a timeout and a 2nd activation. The light green parts indicate the extension described in Sec. III-D. (Shortcuts: AOC = Activate Operation Command; SV = StateVersion; τ = Random Token)

that arrives after the activation duration has timed out and a regular new activate operation command that also arrives very close to the timeout.

If the control has been released it can send a stop activate operation command to stop the activation of the device functionality before the activation duration has timed out. This may reduce the time between the control has been released and the end of the activation. The device functionality activation times out if this command has a high latency.

There are three main advantages of the described mechanism. The first advantage is that all aspects of the mechanism are covered by the self-description mechanisms defined within IEEE 11073 SDC. This means that no a priori knowledge is necessary to implement an interoperable, vendor-independent and dynamical interconnection between control clients and devices with remote activation capabilities. All necessary information and requirements are included in the device capability description. Thus, it is not necessary to have specific profiles or documents that define this interconnection to ensure the safety requirements. The second advantage is that the usage of the state version of the activate operation target does not produce any description or communication overhead. The state version is an mandatory attribute and the control client has to subscribe to the change events of the target state due to medical safety issues. Thirdly no time synchronization is needed between client and device.

D. Extension Against Malicious State Version Calculation

Using the knowledge about the increment of the state version by one a malicious control client does not have to wait for the asynchronous event containing the new state version. The new state version can be calculated. This can compromise the jitter safety at the beginning of the remote activation. (The further process is not affected.) The control client sends the first activate operation command including the current state version n in its safety context element. Then the control client sends the second activate operation command immediately after the first command but holding the blocking period (see Sec. III-E) that contains the state version $n+1$, as it can be calculated easily. If both commands arrive at the device with a high latency the device will trigger its functionality activation although this could lead to an unsafe situation.

This can be solved by introducing a token metric that contains a random number. A new random number will be generated when the target state enters the *STANDBY* state. For the safety context of the activate operation command it will be defined that the current value of the token metric has to be included into the message header. As the control client cannot calculate the random token this extension will fix the possibility of an unsafe situation. The description and communication overhead is low because the token will only be generated if the *STANDBY* state is entered.

E. Considering Activate Operation Command Flooding

Furthermore there is another way a malicious control client to compromise the mechanism. After the initialization the control client can still calculate the required state version without waiting for the asynchronous response. Thus, the control client can flood the device with a high number of activate operation commands in a very short time. If the device is not fast enough to process the activate operation commands they will be buffered. When the control is released and stops sending new activate commands there are still commands within the buffers of the device. These commands will be processed and the device functionality will be activated longer than it is intended by the physician using the control.

This can be solved if the device implements a blocking period in which the incoming activate operation commands for this target are not processed and dropped. As a too long blocking period can lead to a drop of valid commands we empirically determined 25 % of the activation duration as suitable. This ensures that flooding with activate operation commands cannot lead to a safety critical misbehavior.

F. Multi-Control-Scenario and Security Consideration

In real OR environments there are probably multiple controls that can be associated with one and the same activation of one device. Thus, it can be necessary to be sure that the activation cannot be triggered by more than one control at the same time. So the device has to decide whether it accepts or rejects activate operation commands based on the information from which control this command has been sent. To identify the client the optional header field *wsa:From* could be used. The disadvantages are that this field is not mandatory and it can be manipulated. For a trustable authentication it is recommended to use a HTTPS connection. As MDPWS forces the usage of x.509.v3 certificates the pure authentication can be extended by roles.

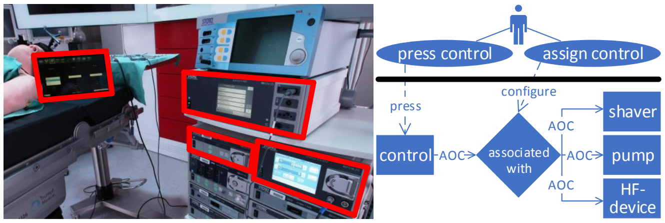

Fig. 3. Left: Demonstrator for safe remote activation (at ICCAS, Leipzig): Control client implementation on a tablet computer; devices with activation functionalities: shaver, pump, HF-device (involved components highlighted). Right: Use Case Diagram (“assign control” is out of scope of this paper) and Activity Diagram for this use case (AOC = Activate Operation Command).

For example the chief physician will have more rights and a higher priority than the surgical nurse. These roles can be defined using x.509.v3 extensions. They can be used to define privilege hierarchies. This allows the decisions whether the control is generally allowed to trigger the activation and which control has the higher priority in the case of multiple triggering requests. Note that it is recommended to use the keep-alive connection in order to minimize the overhead of establishing the secure connection. It might be useful if the device requires also a secure for other interactions like reading the device description. This ensures that the secure connection is built at a point of time where performance is not as relevant as it is for the remote activation.

IV. Proof of Concept

To validate the mechanisms described in this paper we built up a real world demonstrator with real medical devices. To demonstrate the safe remote activation we use three different devices: a surgical shaver, a surgical pump, and a HF-device. The control client is represented by a touch based control application that represents the functionality of a foot switch. This control client can be associated to one of the three devices to activate their device functionalities. The left part of Fig. 3 shows this set-up as part of a complete OR environment that has been built up within the OR.NET project [7]. The right part of Fig. 3 gives a schematic overview of the shown use case.

We built up a system that allows a continuous activation of the devices triggered by the control client based on the described mechanisms. The realization is based on standard Ethernet using off-the-shelf hardware. For the implementation we use the IEEE 11073 SDC reference implementations *openSDC* in Java for the control client and *OSCLib* in C++ for the medical devices. The system has been realized with an activation duration of 250 ms, without any latency optimization of neither communication stack, application logic, nor network infrastructure. For devices with a high mechanical inertia like a surgical pump this duration is quite acceptable. For the activation of surgical shaver and HF-device the performance of the systems has to be increased in the future.

V. Conclusion and Future Work

In this paper we presented mechanisms for safe remote activation of safety critical medical device functionalities. The described system meets the requirements arising from an unsafe standard network using off-the-shelf hardware. It is realized by the usage of the new IEEE 11073 SDC standard proposals that enable an interoperable and vendor independent interconnection of medical devices. The safe remote activation is built up on the principle of a cyclic reactivation of the device functionality and the additional embedment of safety related information into the remote activate operation command. The main advantage of the described system is that is enables a real interoperability and plug-and-play functionality between controls (like foot switches, handhold switches, and touchscreens) and devices because all described mechanisms make use of the self-description capability provided by IEEE 11073 SDC. Thus, both the medical device and the control client do not need any a priori knowledge about each other. Additionally there is not much communication overhead and no time synchronization between client and device is necessary.

The described system is suitable for a remote activation of devices functionalities with a safe state that is either “on” or “off” if anything gets wrong. In the proposed classification of device functionalities these are the classes 1 and 2. The concept has been validated within a real world demonstrator. The control client can be dynamically associated with a surgical pump, surgical shaver, or HF-device to trigger the remote activation of the device functionality in a safe way.

In the future the performance of the system has to be increased to enable lower activation duration time periods. This means that the processing and propagation time has to be decreased in order to realize a lower period of time between two activate operation commands. Furthermore class 3 devices have to be addressed, where the state “on” and “off” have to be reachable at any time. One aspect to realize this is to increase the deterministic behavior of the network. Additionally mechanisms for a user-friendly configuration of the dynamically assignable controls have to be developed and investigated. This also includes strategies to display the current assignment and configuration to the actors within the OR considering the requirements of a surgical environment.

References

[1] A. von Saucken, S. Donner, and M. Kraft, “Ergonomic problems originating in the use of high-frequency and ultrasonic medical devices,” *Biomedical Engineering/Biomedizinische Technik*, vol. 57, no. SI-1 Track-J, pp. 951–954, 2012.

[2] M. Kasparick, S. Schlichting, F. Golatowski, and D. Timmermann, “Medical DPWS: New IEEE 11073 standard for safe and interoperable medical device communication,” in *Standards for Communications and Networking (CSCN), 2015 IEEE Conference on*, Oct 2015, pp. 212–217.

[3] ——, “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*, Aug 2015, pp. 1721–1724.

[4] IEC 61784-3, “Industrial communication networks Profiles Part 3: Functional safety fieldbuses General rules and profile definitions.”

[5] DGUV Test: Prüf- und Zertifizierungsstelle Elektrotechnik, “GS-ET-26 Grundsätze für die Prüfung und Zertifizierung von ”Bussystemen für die Übertragung sicherheitsbezogener Nachrichten” Stand 2014-03,” 2014.

[6] IEC 61800-5-2:2007, “Adjustable speed electrical power drive systems - Part 5-2: Safety requirements - Functional.”

[7] “OR.NET - Flagship project funded by the German Federal Ministry of Education and Research,” www.ornet.org (retrieved May 20th, 2016).