ENSURING MEDICAL DEVICE INTEROPERABILITY WITH LEGACY SYSTEMS

SDC
SERVICE-ORIENTED DEVICE CONNECTIVITY

AURIGA®

> *The lack of medical devices that are interoperable from the device's interface leads to high hurdles for product and service innovations so that manufacturers shy away from the development of the products or services and so the benefits of the potential innovation never manifest in the market.*
>
> **Dr. Stefan Schlichting,**
> Subgroup chair for IEEE 11073 SDC standards and former member of the board of OR.NET e.V.

Today, every healthcare facility setting is different and typically not a mono-branded environment, but utilizing the best of bread approach. Open standards proliferation has dramatically facilitated interoperability and data exchange. By leveraging the IEEE 11073 Service-oriented Device Connectivity (SDC) standards, point-of-care (PoC) medical device manufacturers use integration and mapping techniques to add value to their products.

This whitepaper discusses challenges occurring while converting proprietary protocols into IEEE 11073 SDC and shares Auriga's solution development experience in ensuring medical device interoperability with hospitals' legacy systems.

SDC allows for bidirectional or device-to-device communication enabling highly acute environments to operate in a safe and secure mode. The standards address the most common device-related challenges healthcare facilities face today, among which are device control, data visualization, automation, data analytics and alarms fatigue.

A 12-day alarm system analysis at Johns Hopkins indicated there were an average 350 alerts per bed per day. In one intensive care unit, the average was 771 alerts per bed per day. Yet some 85%–90% of these alerts are false or nuisance alarms, indicating conditions that don't require clinical intervention.

Failure to respond to an alarm can cause patient harm and may potentially be life threatening. The United States Food and Drug Administration (FDA) reported over 500 alarm-related patient deaths during a five-year period, and many believe that this report significantly underestimates the magnitude of the problem.

IEEE 11073 SDC AT A GLANCE

SDC was envisioned and developed by OR.NET, a non-profit organization uniting industrial specialists, clinicians and researchers.

OR.NET e.V.

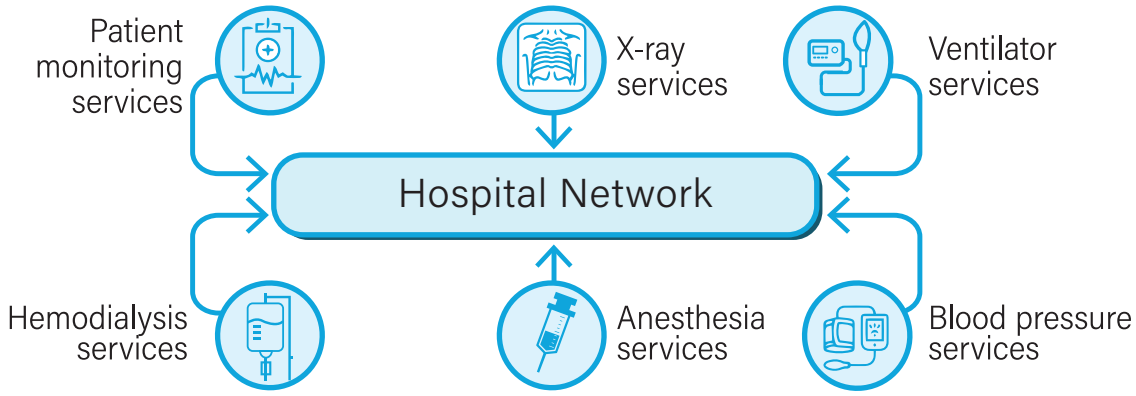
IEEE 11073 SDC is a part of the established ISO/IEEE 11073 family of device interoperability and complements the Healthcare Interprise IT standards landscape that typically uses HL7 v2, HL7 FHIR or DICOM.

The core SDC standards are based on the paradigm of a service-oriented architecture (SOA) and consist of the transport standard called Medical Devices Communication Profile for Web Services (IEEE 11073-20702), the Domain Information and Service Model (IEEE 11073-10207), and the Exchange Architecture and Protocol Binding (IEEE 11073-20701).

SDC specifies the communication protocol for PoC equipment to ensure patient safety by enabling reliable data exchange between medical devices within an open IP-based system. While SDC - like every emerging technology - promises new opportunities for medical device manufacturers and hospitals, the conversion of legacy systems to SDC protocol can be challenging and tricky. We summarized our experience below with confidence that it could be applied to most PoC medical devices on the market.

We are sure that it will help you investigate the challenges and mitigate the potential risks beforehand.

SERVICE ORIENTED MEDICAL DEVICE ARCHITECTURE



EIGHT CHALLENGES TO KEEP ABREAST OF

1. SETTING LEGACY DEVICE ATTRIBUTES IN MDIB

MDIB is a structured collection of any data objects that are provided by any PoC medical device including descriptive and state information.

Creation of SDC Medical Data Information Base (MDIB) for medical device is very device-specific. For this reason, it is no wonder it is one of the most complicated tasks in the protocol converter development. MDIB should contain description of all metrics, alerts and operations applicable for the device. Each entity comes with many attributes that have to be set correctly. MDIB is a very sensitive part of the converter as any minor change can disrupt its functionality.

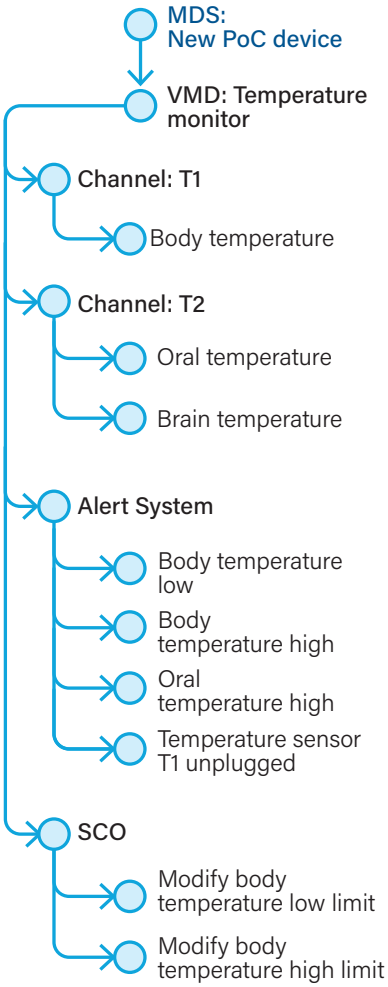
A legacy PoC device could be translated into MDIB in variety of ways due to flexible MDIB structure. Our best bet is using iteration model for the MDIB evolution:

- Get a list of PoC medical device signals provided via proprietary protocol
- Create proto MDIB mapping (featuring metrics and waveforms only), based on proprietary protocol signals
- Improve MDIB metrics description based on device physical structure using SDC terms such as bodysites and physical connectors
- Investigate PoC medical device run time scenarios, extend MDIB with alert conditions
- Investigate PoC medical device remote control scenarios, extend MDIB with remote control operations based on SDC services concept
- Implement complex scenarios design, identify uncovered SDC requirements, creating an exception list

2. USING XML FOR COLLABORATIVE WORK

The native MDIB format is XML, which has a lot of advantages in terms of the integration with various software. However, a big disadvantage of XML is that is not quite suitable for collaborative work, especially editing.

You could also use database and XML schema modeling tools to mitigate this issue. It is the best practice to store the descriptive part of MDIB in a relational database with the good frontend. This allows all stakeholders to benefit from a clear MDIB representation and avoid potential conflict of interest, as well as grant correct access rights to every user's type. In order to obtain the resulting MDIB XML file, a special database tool should be created and validated.



3. MDIB DESCRIPTIVE PART DEPENDENCIES

The MDIB descriptive part represents an abstract model of PoC device. It can be complex and heavy with thousands of entities and dependencies to be easily obtained by conversion algorithms. So, one must consider optimizing the run time access to the MDIB entities as the highest priority in the protocol conversion.

Auto-generated mapping proved to be one of the most useful optimization tools. Python scripts should parse the descriptive part of the MDIB settings and requirements in XML files during the project compilation time. This way all kinds of dependency tables are available in the proper internal format before execution, which reduces the settings flexibility, but not affecting reliability.

4. MAPPING OF LEGACY DEVICE PARAMETERS

Each mapping case is a subject for medical risk management assessment and requires mitigation measures with a confirmed scenario added to the list of known defects in product documentation.

Protocol conversion presumes one-to-one mapping of the legacy device parameters defined by SDC. In some cases, direct conversion is unfeasible due to lack of information about PoC device. For example, SDC requires determination time for all metrics and start time for numeric metrics, which is not supported by the most legacy protocols.

Using a state machine to record the device operation history and store its current state data might sound like a viable solution for conversion. However, please, note that the context information will be lost or become inaccurate in case of power outages, system reboot and loss of data packages.

5. TIME SYNCHRONIZATION

Timestamp conversion and time synchronization are probably the most challenging problems as the majority of legacy devices transmit timestamps (metric determination time, waveform timestamp, and alert timestamp) in their own epoch. After analyzing several legacy protocols Auriga developed a complex algorithm to adjust the received timestamps to variety of the commonly used epoch (UNIX, Gregorian, etc.)

Another time-related problem refers to clock synchronization between the protocol converter and the medical device. Although devices typically provide protocols for time synchronization, those will not capture minor time deviations, and might require special time-sync protocol support implementation.

6. NETWORK BANDWIDTH

For SDC it means approximately 80kB per second traffic for one PoC device due to a vast XML usage if compression if not employed. Typical clinic supports hundreds of patients that also contributes heavily to the network load, topped by the additional non-SDC network traffic such as video calls etc. Engineers should optimize SDC traffic routing or even assign dedicated hardware for SDC network segments to avoid the overload of the network nodes.

A typical medical device generates the following types of traffic:

- Device advertisement – slow flexible rate
- Numeric metrics advertisement – once per second each
- Waveforms signal – once per 200 ms each

7. CYBER SECURITY

Medical gateways require two networks: SDC and legacy. The legacy network is often unreliable in terms of cyber security:

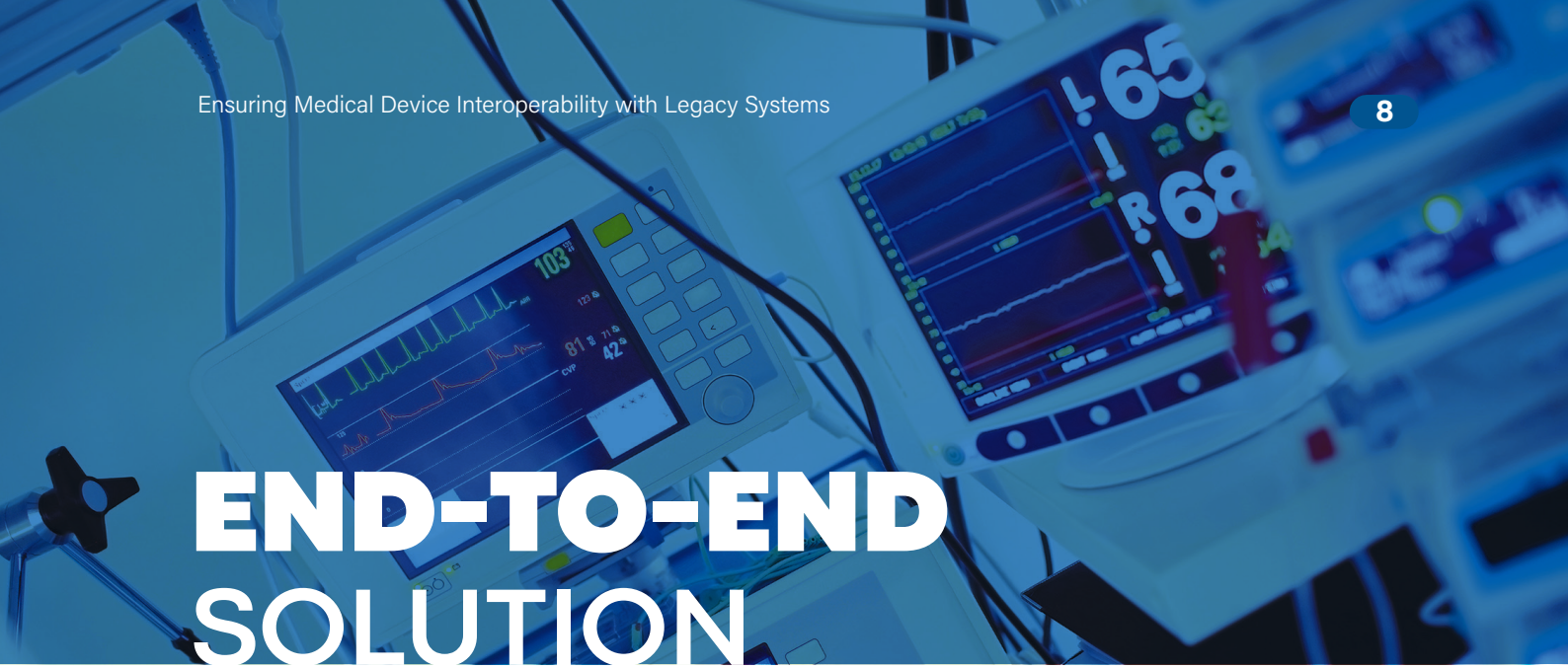
- No peer-to-peer authentication
- Messages are not encrypted
- Flexible TCP/UDP ports - hard to setup a firewall

The issue can be resolved by isolating legacy network segment or developing a low-cost hardware SDC coverter dongle to avoid intersections with other networks.

8. MEDICAL ALERTS

Legacy protocol data transfer could also be limited with another hidden problem. Some critical information (for example a fact of SDC medical alert) could not be delivered by proprietary protocol as explicitly registered event but might be deduced implicitly by PoC medical device outcoming traffic analyze. The converter does not simply transfer this information but is supposed to make a decision whether to create a medical alert or not.

The problem should also be a subject of medical risk analysis assessment potentially increasing the medical device safety class (according to IEC 62304).



END-TO-END SOLUTION

CONVERTER OPERATIONAL REQUIREMENTS

Appropriate protocol converter should work automatically and autonomously without any user interactions. For all supported protocols, it must be recognized as a native device acting as a proxy (Virtual Device Gateway).

The converter has to meet several critical requirements for correct operation:

- Detect a device type, protocol version and other important settings
- Recover automatically in case of any software or hardware failure
- Support automatic device discovery and hot-plug scenarios
- Comply with cyber security requirements like encryption, networks separation, white lists.

While the above mentioned challenges are rather common and can be solved by relatively simple workarounds, there might be problems that require additional engineering efforts in the proprietary protocols and testing tool environment. They may seem time and labor consuming but eventually are worth the pain as top-tier medical device manufacturers believe.

Here, we would like to focus on two potential bottlenecks that most likely compromise the workload of the converter if not addressed properly.

XML REQUESTS OPTIMIZATION

SDC standard assumes transmitting medical parameters and data via XML structures over the network. Some parameters are static or stable for a long time while others change rapidly in accordance with medical device outgoing traffic sample rate. Profiling recognizes XML conversion as an expensive operation leading to issues when MDIB parameters might not be properly or timely updated. Therefore, the data converter should not just repeat and transmit data, but also minimize XML writing requests by caching and packaging.

ENABLING FEEDBACK MARKERS

According to SDC standard, PoC medical device remote control actions must be verified by the feedback status while unconfirmed control operations should be treated as failure.

Some proprietary protocols do not fully support explicit remote control status reporting. We highly recommend using implicit PoC feedback to get the desired status so that mentioned result could be deduced by analyzing the PoC device outgoing traffic observing feedback markers. Each remote control has unique feedback markers that could be combined into set of rules allowing to get confirmation status from the incoming data within the expected period.

INVESTMENT THAT PAYS OFF

Being a strong adopter of IEEE 11073 SDC, Auriga helps medical device manufactures build protocol converters that provide reliable connectivity capabilities using proprietary network communication protocols.

REVERSE ENGINEERING

Proprietary protocol description may not be available or complete for various reasons. As a result, a number of unexpected device behavior related issues is likely to be identified and can be successfully fixed only by a significant reverse engineering effort.

MEDICAL DEVICE SIMULATOR

Most medical device manufacturers have patient hardware simulators in-place that generate the main set of patient measurements. However, for many complex scenarios, tricky alerts and measurements, patient simulator is not applicable. PoC device hardware errors and technical conditions are not reproducible by most patient simulators either. One of the most convenient and effective way to address this type of challenges would be development of the customized medical device simulator. This approach proved to be very handy in our practice. For example, in case medical device self-monitoring algorithms should detect the equipment breakdown (motors, pumps, valves, connectors).

Since artificial damaging of a real device would be barely reasonable, Auriga developed a medical device simulator to generate fracture conditions. The simulator is a network application that models device messages in legacy protocol format. It can also be used for auto testing to run predefined test scenarios in absence of the real medical device.

AUTOMATED VERIFICATION

A combination of testing approaches is always required to ensure the product meets functional and system requirements, i.e. unit, manual and automated testing. The explicit need of test automation in case of protocol conversion comes from the fact that conversion of over 900 physiological parameters cannot be timely verified with manual effort.

The following requirements are subject to automated testing:

- Conversion of physiological metric data
- Waveforms data accuracy
- Configuration parameters accuracy (e.g. alarm limits)
- Events such as alarm limits violation

Although the use of hardware vital signs simulator is valuable for running individual tests, a more general approach is required to cover all physiological data parameters. Auriga has developed special software Patient Simulator with scripting scenarios support that generates proprietary network traffic to cover all the PoC medical device parameters. Usually these are the values of thousand combinations are heavy handled by manual testing.

EMBRACING MANUFACTURER-INDEPENDENT CONNECTIVITY

Auriga offers a wide range of software R&D and consultancy services, including device prototyping and emulation, automated and interoperability testing, system V&V, legacy code refactoring, porting to CPUs and operating systems, third party platforms customization and integration, as well as maintenance and sustaining engineering.

Integration of relevant devices, applications and hospital IT systems enables emerging patient care and healthcare analytics technologies to operate with a larger amount of data acquired from different sources.

Therefore, manufacturer-independent data exchange is becoming crucial for the entire chain of healthcare providers which was forecasted by AAMI as the top medical device challenge back in 2012.

Auriga has been conducting a comprehensive research for almost a decade to improve medical device interoperability and information security. Auriga's expertise is enhanced by projects in creating documentation, compliance assessment, testing of proprietary communication protocols, delivering flexible parsing and conversion mechanisms for supporting HL7, DICOM and SDC standards.

To find out more visit us at www.auriga.com/medical-devices/
For any inquiries or questions, please contact andrey.shastin@auriga.com