From IEEE 11073 SDC Device Specializations to Assistive Systems: Rule-based Data Analysis for Minimal Invasive Surgery

Martin Kasparick, Björn Butzin, Frank Golatowski
Institute of Applied Microelectronics
University of Rostock
Rostock, Germany
firstname.lastname@uni-rostock.de

Jonas Pabst
Institute of Automation
University of Rostock
Rostock, Germany
firstname.lastname@uni-rostock.de

Hans-Joachim Cappius, Peter Westerhoff
WORLD OF MEDICINE GmbH
Berlin, Germany
firstname.lastname@wom.group

Björn Andersen
Institute of Medical Informatics
University of Lübeck
Lübeck, Germany
lastname@imi.uni-luebeck.de

Dirk Timmermann
Institute of Applied Microelectronics
University of Rostock
Rostock, Germany
firstname.lastname@uni-rostock.de

***Abstract*—Modern surgical devices are full of innovations and provide plenty of functionalities. However, they only perform to their full potential if they are properly configured. O nly a small subset of surgical device functionalities is used due to the high complexity of today's device systems and the omnipresent situation that only the circulating nurses can (re-)configure non-sterile devices. Hence, there is a huge need for safe and effective assistance supporting the operating room (OR) staff to continuously work with the potentially best device setting to increase patient's safety and clinical outcome. Therefore, we propose a concept for situation-aware parameter recommendations and automatic (re-)configuration. F ree a ccess t o a dequate d ata i s a prerequisite to extract knowledge for assistive systems and to provide situation awareness during execution. Thus, manufacturer-independent medical device interoperability is a basic requirement. Consequently, we use the new IEEE 11073 Service-oriented Device Connectivity (SDC) standards family, including *Device Specializations*. As part of the developing committee we introduce the idea and concept behind *Device Specializations* and highlight their use for assistive systems in the domain of minimal invasive surgery. To ensure safety and effectiveness of the assistive functionalities, our concept enforces deterministic rules providing a predictable and approvable behavior. We demonstrate our concept by the use case of a smart surgical double roller pump using an interpreter-based *Rule Execution Engine*.**

***Index Terms*—Rule-based Assistance, Open Standards, IEEE 11073 SDC, Device Specializations, Interoperability, Data Exchange, Situation Awareness, Minimal Invasive Surgery**

I. Introduction and State of the Art

Medical interventions and the used medical devices become more and more complex [1], especially in highly technologized fields l ike M inimal I nvasive S urgery (MIS). W hile modern and innovative medical devices often provide a huge amount of smart and innovative functionalities, physicians only use a small subset in their daily work. It is no longer possible to have proper knowledge of all aspects of the used medical device systems [2]. Thus, there is a massive need for assistive systems making context-aware recommendations and/or automatic (re-)configuration of device parameters in a safe and effective way. This will make the complexity manageable, support the use of innovative device functionalities, reduce the stress level of the therapeutic team and consequently improve patient's safety and clinical outcome. However, today medical devices mostly behave static, without deep context awareness and adaptive reaction on patients and caregivers.

Such assistive systems have a high demand for data. On the one hand, data is used to get situation awareness being the basis for beneficial assistive decisions. On the other hand, to extract prior knowledge to develop the assistive system [3]. Currently, medical devices do either not share data at all or in a manufacturer-specific ecosystem using proprietary hard- and software protocols. Manufacturer-independent interoperability is mostly not available [3], [4]. However, using open standards for medical device integration is one of the key enablers to overcome these challenges [5]. Currently, IEEE 11073 Service-oriented Device Connectivity (SDC) and HL7 Fast Healthcare Interoperability Resources (FHIR) are the most promising technologies for these purposes.

In this paper, we propose a comprehensive concept for safe and effective situation-aware assistive systems supporting the caregivers with recommendations for medical device parameters or autonomous (re-)configuration. We focus on the precondition of medical device interoperability based on IEEE 11073 SDC *Device Specializations (DevSpecs)* [6], on the data collection, as well as on the design principles of approvable context-aware assistive functionalities. We demon-

This work was supported by the German Federal Ministry of Education and Research (BMBF) (*AFluCoMIS* project grant number: 13GW0164) and Federal Ministry for Economic Affairs and Energy (BMWi) (*PoCSpec* project grant number: 03TNH017)

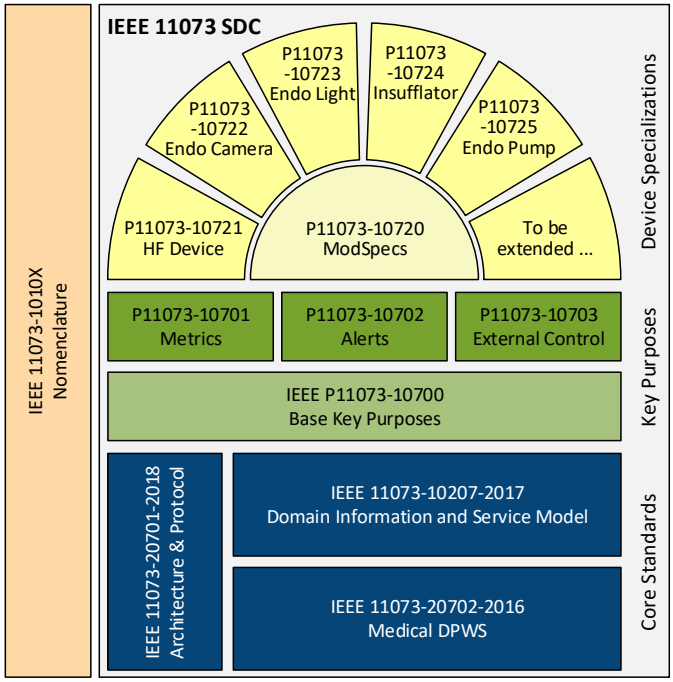

Fig. 1: IEEE 11073 SDC standards family: Approved and published *Core Standards* and standardization projects (indicated by "P" at the beginning of the standard number) for *Key Purposes* and *Device Specializations*.

strate our concept with the use case of a smart surgical pump providing situation-aware parameter recommendation and (re-)configuration.

II. Key Enabler: Device Interoperability

Manufacturer-independent, semantic interoperability is a key enabler for future innovations of medical devices within the OR, intensive care unit (ICU), and whole clinic. This includes exchange of information and control commands between devices at the point-of-care, as well as the interconnection to clinical and research repositories (information systems). Currently, there are two emerging technologies having the potential to solve these challenges: IEEE 11073 Service-oriented Device Connectivity (SDC) for the device-to-device communication and HL7 Fast Healthcare Interoperability Resources (FHIR) for the interconnection of devices to the information systems and repositories [7], [8].

IEEE 11073 SDC, short SDC, is a new series of standards within the well-established IEEE 11073 standards family [9], [10]. As SDC is based on the idea of the Service-Oriented Architecture (SOA), it follows a fundamentally different approach than the "classical" IEEE 11073. The structure of the SDC family is depicted in Fig. 1.

A. IEEE 11073 SDC Core Standards

The three *SDC Core Standards* [9] are already approved and published standards (see blue part at the bottom of Fig. 1): Medical Devices Communication Profile for Web Services (MDPWS), or Medical DPWS (IEEE 11073-20702) takes care about data exchange and dynamic discovery of network participants. The Domain Information and Service Model (IEEE 11073-10207) defines how the devices describe themselves with their capabilities, requirements, and state information. IEEE 11073-20701 defines the allover architecture as well as the binding between the *Core Standards* and to other standards, e.g. for time synchronization and Quality of Service (QoS) aspects. Together with the usage of comprehensive nomenclature standards, like the IEEE 11073-1010X series, this enables manufacturer-independent semantic interoperability.

B. Development of IEEE 11073 SDC Key Purposes

Basically, *SDC Core Standards* enable the modeling and interconnection of any imaginable medical device. However, for safety, security, and effectiveness of networked medical device systems, it is useful to define a broader set of requirements for the actual behavior of service providers and consumers in different roles. Therefore, the *SDC Key Purposes* are currently being developed (see green part in the middle of Fig. 1). These standards are independent from the particular medical use case by defining general requirements and behavior patterns for providing and consuming information (P11073-10701), alerts (P11073-10702), external control (P11073-10703), and general requirements for SDC-compliant participants (P11073-10700).

Key Purposes will enable manufacturers to build systems contributing safely and effectively in a distributed medical device system as they are more specific than the *Core Standards*. The content varies from basic issues all participants have in common, like device labeling, risk management for network connection issues (connection loss, QoS aspects, etc.), logging, usage of private semantic codes up to behavior specification in specific situations of external control or alert management. It is also intended to add test descriptions for the specified requirements.

C. Development of IEEE 11073 SDC Device Specializations

Within the *PoCSpec* project we contribute to the development of so-called *Device Specializations (DevSpecs)*. While the lower parts of Fig. 1 are use-case-independent, *DevSpecs* define requirements and propose models for specific classes of medical devices. The goal of *DevSpecs* is on the one hand to define the network representation of medical devices as strict as necessary to ensure manufacturer-independent interoperability without any need of knowing the particular communication partner during development of the medical device. On the other hand, *DevSpecs* will be as flexible as possible to be adopted by all manufacturers and to allow manufacturer-specific innovations.

It is intended to define modular *DevSpecs* to enable reusability of aspects that multiple device classes have in common. Such modules will be defined in the *Module Specification (ModSpec)* standard (P11073-10720). Currently, the focus is on high frequency (HF)[1] surgical devices and devices for endoscopic surgery (P11073-10721 to -10725). In the future,

[1]The term radio frequency (RF) is also commonly used. As the particular standard IEC 60601-2-2 uses high frequency (HF), we rely on this term.

more standards will follow dealing with other devices from other medical domains.

III. Concept

In the near future, the capabilities of manufacturer-independently interconnected medical device systems will be used to provide assistive systems for the surgeons and caregivers in the OR. The general idea is to collect data from different network participants to analyze this information and to combine it with device-internal knowledge, for example to recommend better device parameters or to change parameters automatically if this is in accordance to the risk management. The basic concept of the *AFluCoMIS* project is described more in detail by Benkmann et al. [11].

The challenge is to build up an approvable system. Therefore, it is necessary to approve the rules leading to recommendations or automatic parameter changes. Hence, these rules have to be deterministic. For the rule creation, a broad *Knowledge Database* for the specific use case is needed. Accordingly, the concept is divided into three parts:

(1) data collection,
(2) data analysis, rule creation and rule releasing, and
(3) rule execution during surgery.

A. Data Collection

The basic concept of data collection for research purposes and potentially artificial intelligence (AI) or machine learning (ML) applications is described in [8]. IEEE 11073 SDC, including *Key Purposes* and *DevSpecs*, enable the collection of semantically and contextually tagged information of medical devices, including device parameters, vital signs, etc. The context includes information about patient, medical procedure, surgeon, body-site, etc. Parts of this information will be transferred to the *Clinical Care Repository* (green part of Fig. 2), however, a significant amount might not be interesting to be stored in the Clinical Information Systems (CISs). With respect to the patient consent and mechanisms of de-identification done by a *Trusted Third Party*, the collected data is transferred to *Research Repositories* (right side of Fig. 2).

Information regarding recovery process, complications, clinical outcome, etc. will also be stored in the CISs (green part of Fig. 2) and transferred to the *Research Repositories* (blue part of Fig. 2), with respect to privacy issues.

Both kinds of *Research Repositories* for clinical data and for the device data will be aligned and consolidated to provide a comprehensive *Knowledge Database* (blue part of Fig. 2). Note that this data merging requires suitable de-identification mechanisms by the *Trusted Third Party*.

A component within the OR (orange part of Fig. 2) implementing the *Service Consumer* role collects all information of interest in the medical device ensemble aligned with contextual information concerning patient, surgeon, medical intervention, etc. The proposed concept does not specify which component is doing the data collection and transfer to the *Research Repositories*. It does not matter whether a dedicated component is doing this task or an existing medical device implements this additional functionality. The data transfer is done by an *HL7 FHIR Client* implemented by this component. No semantic information is lost during the transformation from the SDC to the FHIR representation of the information.

B. Data Analysis, Rule Creation, and Rule Releasing

The *Knowledge Database* provides a comprehensive set of information for the particular use case. Human experts can analyze this data in a (structured) way. In the future, methods of AI and ML will be used to extract relevant knowledge from the data. These systems will support the human experts to find causalities and correlations for example between the patient's constitution, used medical device settings during surgical intervention, potential complications, etc. and the clinical outcome. In the blue part of Fig. 2 this process is depicted as *Rule Creation Engine*.

The created rules are stored in the *Rule Database* for unchecked rules. Before the rules can be released to be used by the medical device in the OR, a clearance process has to be passed. Medical and technical experts check the validity, safety, effectiveness, usability, etc. It is possible to support this release process with test data being extracted from the *Knowledge Database*. The explicit releasing process ensures the quality and safety of the rules. Consequently, only a well-defined set of rules is released to the medical device.

The proposed concept forces the rules to follow a strict *Event-Condition-Action (ECA)* rule pattern. Thus, for a given situation it is possible to calculate/predict the behavior of the medical device deterministically. No matter whether probabilistic methods of AI or ML took place in the data analysis and rule creation, the rule itself is deterministic. As long as it is not possible to clearly track decisions of AI/ML systems, we state that this approach of deterministic rules and a strict releasing process shall be used for medical devices in high acuity environments.

C. Rule Execution and Usage During Surgery

After deploying the rules, the medical device can use them to assist the actors in the OR (orange part of Fig. 2). The rules can apply to a static context regarding a particular surgical intervention, like patient, surgeon, medical intervention, etc. information. For example, the *Rule Execution Engine* could propose the best parameter set for the given age, physical conditions like body mass index (BMI), pre-existing diseases and interventions, etc. of the patient. The device will present such recommendations for user's acknowledgement.

During the surgery, the *Rule Execution Engine* will react to dynamically changing situations. Therefore, the medical device will act as a *Service Consumer* to access information of other SDC-compliant participants. Eventing mechanisms according to the publish-subscribe pattern and explicit request mechanisms according to the request-response pattern are provided by SDC for this purpose. Based on events and conditions received via SDC, (e.g., changes of vital signs parameters, usage or reconfiguration of other medical devices

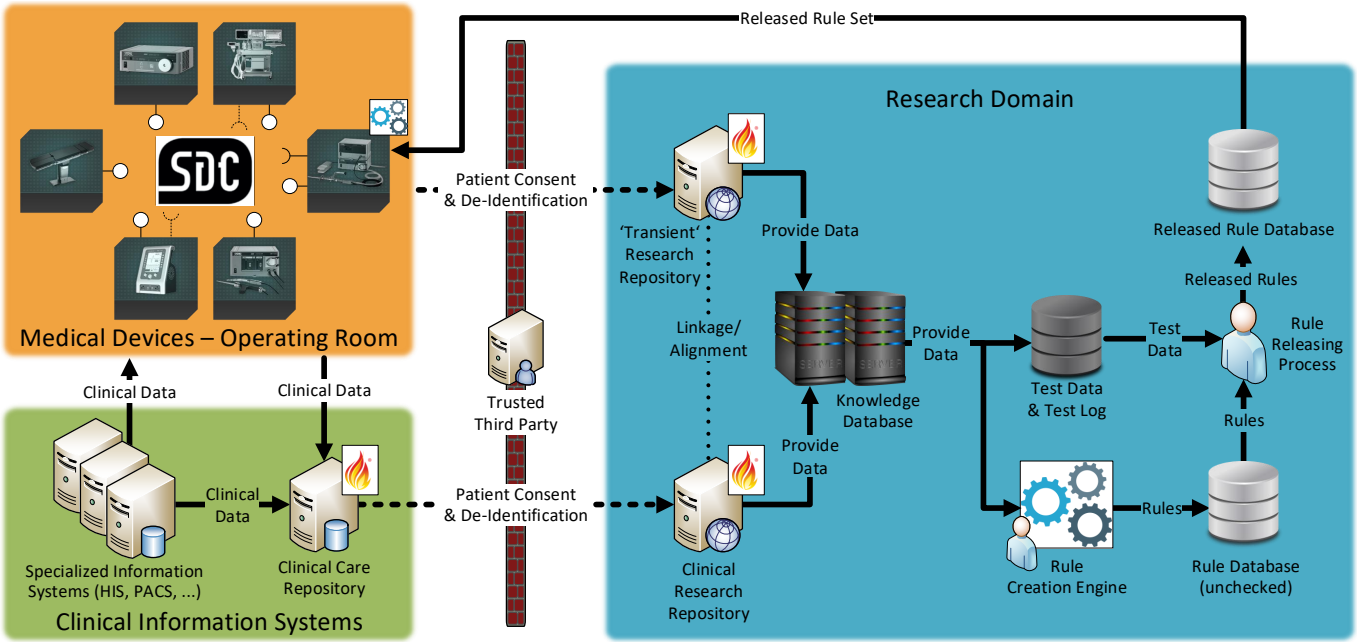

Fig. 2: Concept for situation-aware assistive medical devices, based on the general enabling concept and figure derived and extended from [8]. (Medical device with gear/engine symbol represents the smart device; full-circle symbols represent *service providers*; semi-circle symbols represent *service consumers*; flame symbol indicates HL7 FHIR)

being involved, etc.) recommendations will be calculated dynamically. In order to ensure safe and approvable operation, we introduced several mechanisms:

- Rules only affect the desired value. The actual output remains the task of internal, approved control algorithms.
- Rules are discarded in case of errors or missing data and cannot be partially executed, due to atomic execution.
- Initialization rules only propose initial device settings. They can be adjusted, if they not meet the requirements.
- Rules for dynamic parameter changes, require explicit opt-in by caregivers before starting the procedure.
- Constraints in combination with priority levels allow for fine grained control. Device constraints determine built-in safety constraints that cannot be exceeded. Critical constraints can only be exceeded by manual intervention but not by rules. Recommendation constraints tighten critical constraints and can only be exceeded by manual intervention and critical demands. Default constraints only become active if no other constraints are set.
- Constraints introduced by a particular rule can only be relaxed by the same rule. I.e., there is no possibility for unintended relaxation of constraints by other rules.
- Rules are checked by human experts before release.

The manufacturer-independent interoperability is the basic requirement for the *Rule Execution Engine* to get information. At development time of the component, it is not possible to know which particular devices will be used in the field. Thus, there is a necessity for the adoption of *Device Specializations* and *Key Purpose* standards (see Sec. II). This will enable the implementation of such *Rule Execution Engines*, as the basic information and interaction mechanisms will be standardized for classes of devices.

IV. Demonstrator: Advanced Surgical Pump

We demonstrate our concept within the use case of an advanced surgical pump used for MISs, like arthroscopies. These pumps provide dilation of the surgical site, such as a joint, to provide a suitable operation area. Therefore, a liquid (typically saline or glucose solution) is pumped into the situs with an appropriate pressure. Additionally, the pump provides a proper flushing to ensure a clear field of vision for the surgeon via the endoscope.

To demonstrate our rule-based assistive parameter recommendation and configuration we focus on these use cases:

- Rule-based loading of patient- and situation-specific parameters at the beginning of the surgery
- Target pressure should follow blood pressure of the patient (with a certain offset)
- Adjust the flow rate when the HF surgical device is used to carry away contaminants

A. General Scenario and Limitations

Currently, the first SDC-compliant devices enter the market, however, they are not yet broadly available. Thus, we use device data from prototypical research-driven medical devices. Nevertheless, the demonstrator shows the suitability of the concept. To generate FHIR compatible medical records, we used the open source tool *Synthea*[2]. In accordance with our use case we added a module[3] to *Synthea* which generates case records for an arthroscopic intervention, including causing conditions, patient encounters, and pre-operative anamnesis of relevant parameters. This data is stored in our *HAPI FHIR*[4]

[2]https://github.com/synthetichealth/synthea
[3]https://github.com/bbutzin/synthea-endoscopy-module
[4]https://hapifhir.io/

Fig. 3: Demonstrator (f.r.t.l.): research prototype of a smart surgical pump (implementing the *Rule Execution Engine*), knee phantom, HF surgical device and patient monitor.

server instance acting as a CIS and provides the patient context during our simulated surgeries.

B. Key Enabler: IEEE 11073 SDC-Compliant Device System

Our device setup consists of three IEEE 11073 SDC-compliant medical devices as shown in Fig. 3: a smart surgical double roller pump (see [12] for details), an HF surgical device, and a patient monitor. All devices offer their descriptive and state information in the network implementing the SDC service provider role. The pump additionally acts as a service consumer to collect information during the surgery to get context awareness for the *Rule Execution Engine* and to collect data for the *Knowledge Database*. The HF device and the pump follow the new *SDC Device Specializations (DevSpecs)* P11073-10721 and -10725 as well as utilizing the *Key Purpose* standards being in development. Thus, devices are able to identify relevant communication partners by the DevSpecs they implement. Their implementations are based on the prototypical research SDC Java library *SDCLib/J*[5]. The patient monitor uses a productive stack implementation, modified for demonstration purposes.

As an additional component, the *Context* or *Session Manager* (for a detailed concept see [9]) provides the contextual information being static during surgical intervention, like patient, surgeon, etc. The *Session Manager* is also responsible to define the particular set of devices being involved in the surgery. It is implemented using the *SDCLib/J* stack.

C. Data Collection

After our pump received the context information from the *Session Manager* it internally generates a storage object in which all relevant data will be collected. During surgery, the pump gathers all data of interest from the SDC network participants, including vital signs, device parameters, device activation, etc. and adds it to the current session. All this information has potentially high update rates and variability. Thus, the types of collected data and the collection frequency are configurable. As soon as the context is closed by the *Session Manager* the collected data and context information is sent to the *Research Repository*. During post-operative care, further data like the procedure outcome is sent to the *Research Repository*. Finally, pre-operative, intra-operative, and post-operative data is merged into the *Knowledge Database*.

As discussed in Sec. IV-A, we do not work with real medical data. Thus, we omit the de-identification and *Trusted Third Party* part in our demonstrator. However, lots of research and development is done in this field to enable a broad medical data access for research and development purposes [13]. As there are no real patients, we also get no real information about the post-operative patient care.

D. Data Analysis, Rule Creation, and Rule Releasing

At this stage the data in the *Knowledge Database* can be used to find patterns, best practices, or parameter correlations leading to best possible clinical outcomes. How this is actually done (e.g. by medical experts, data scientists or artificial intelligence) depends on the owner of the *Knowledge Database*. In our case we collected some preliminary rules at a workshop with physicians and technical experts, derived from their daily routine in the OR. However, in the future, based on real data, advanced studies can give more insight on the influence of certain parameters and how they should be adjusted and under which conditions. While quality assurance and release of rules should be an independent process, for demonstration purposes the rules were checked and released in the same workshop. During our test cases we already identified a set of good practices that ease validation:

- Rules should be concise and focus on a single use-case.
- Define a setup rule for each procedure type, specifying default values and critical constraints
- Define a setup rule for each static property (e.g., patients body weight, age, previous diseases, body site etc.) that might affect settings, to tighten constraints.
- Safety related settings should use *demand* priority, settings for improved clinical outcome and comfort should use recommendation priority.
- Intensively use maximum, minimum, and offset constraints but (absolute) set-points conservatively.
- Define a rule for each relevant run-time event.

As mentioned in [11] and above, one particular rule is to have the pressure generated by the pump close to the blood pressure. It should not exceed blood pressure to avoid extravasation into the tissue but should also not be too low to sufficiently dilate the surgical site. The resulting pressure might further be influenced by the condition of the patient's connective tissue that might be estimated by age, sex, and BMI. As a second example, the activation of an HF device results in a reduced view which should be cleared by increasing the flow for a certain amount of time.

1) Rule Concept: In order to capture these rules in a machine-interpretable but still human readable format we created our own rule language using the *Xtext*[6] framework and an interpreter for run-time execution. The expressivity of this rule language is strongly restricted to ease its validation compared

[5]https://bitbucket.org/surgitaix/sdclib/

[6]https://www.eclipse.org/Xtext/

to Turing-complete languages. This is the result of the lessons learned by one of our earlier approaches. There we used modules implemented in a common programming language. With the former approach, rules where hard to understand by domain experts, their creation was complex and approving such an approach for medical devices is nearly impossible. By using our own restricted rule language and its interpreter these problems are solved. In particular the interpreter can be approved for medical use. If the interpreter only allows for valid and safe actions that can be executed, new rules do not affect approval status. However, the release process of new rules may be subject to approval. In our case the interpreter guarantees that only a certain set of parameters can be changed in a predefined manner. If there is any syntactical fault in the rule or there is some required information missing the execution of the rule is stopped and no actions are taken.

2) Rule Syntax: As described in Sec. III-B, our rule language follows a strict and deterministic *Event-Condition-Action (ECA)* rule pattern. More particularly, rules consist of the four following parts (see Listing 1).

The first part describes enabling conditions of the rule. These might be the device type, its model version, the kind of procedure, the patient's age, sex, BMI, or any other property that determines the validity of the rule. A rule will only be enabled if this condition is met. This approach reduces the set of rules that have to continuously be evaluated by a device at run-time. The enabling part only relies on information from the static context. Thus, whenever the context changes, all rules are checked whether they are enabled or not. Within a steady context already enabled rules remain enabled.

The second part describes the trigger of the rule and thus relates to the "Event" part of ECA rule patterns. The trigger might be a singular event like a device activation or a new parameter value, but can also be more complex like a positive or negative trend of any vital sign, or a parameter in a specific range. Further, there are specialized rules for initial configuration, which are only executed once a context is set. This can be used to give recommendations on accessories, as well as personalized pre-configuration.

Whenever a rule gets triggered (i.e., the trigger condition is met) an additional condition can be specified to check if the rule can be applied safely and to decide which particular actions should be taken. This is necessary as a rule might have slightly different actions depending on the dynamic state. For this reason any kind of previously collected data can be accessed and used in calculations and decision making.

Finally, the fourth part is the description of the action to be taken. In our demonstrator we restricted the rules to only modify pump internal parameters, but no parameters of other devices. More particularly we only modify the target pressure and flow, while the pump controller tries to act accordingly to achieve the set-points. Thus, a parameter recommendation cannot directly influence, e.g., the roller speed. We made this decision to ensure a strict separation between parameter recommendation and control of the device. The rules can influence maximum and minimum boundaries, add a constant offset or propose a certain set-point. Further, we added different priorities, namely *default* as fallback, *recommend* for increased comfort and *demand* for critical restrictions, where *demand* set-points always supersede *recommend* set-points.

```
1 rule DefaultPressure (arthroscopy)
2 on startup
3 if procedure.location == knee_joint then
4     default pressure.setpoint = 45;
5 else if procedure.location == shoulder then
6     default pressure.setpoint = 60;
7
8 rule HighBMI (arthroscopy)
9 on startup
10 if (procedure.location == knee_joint or procedure
       .location == shoulder)
11     and patient.bmi > 40 then
12         recommend pressure.offset = 10;
13
14 rule FollowBloodPressure (arthroscopy)
15 on new MDC_PRESS_BLD_NONINV_SYS
16 define current MDC_PRESS_BLD_NONINV_SYS as
       bldPres;
17 if true then  //i.e. always
18     recommend pressure.setpoint = bldPres-30;
19     demand pressure.max = bldPres;
```

Listing 1: Three example rules. Be aware that these rules just serve as example to show some rule capabilities. They are not approved for clinical use!

3) Rule Example: Listing 1 shows some example rules. Each starts with the `rule` key-word followed by the name of the rule. The name allows a rule to be manually confirmed each time it is triggered or to authorize the rule to change parameters without further human confirmation. Enclosed in parenthesis the enabling condition follows. In this case the rules are enabled in arthroscopic procedures. Afterwards the trigger of the rule is defined. The rule in line 14 is executed every time a new non-invasive systolic blood pressure value measured in *mmHg* is received. The string representation in line 15 relates to the IEEE 11073-10101 nomenclature. In line 16 we define a variable that can be used in the latter condition and action descriptions. In our example we use the latest available non-invasive systolic blood pressure (i.e. the one that is received from the event). Other options could be the previous value for detecting changes, the average of the last seconds or minutes, or other vital signs and device parameters. In some cases we have multiple action blocks that are executed depending on the given conditions (lines 3-6). In any case only the first action block is executed where all conditions are met. It is possible to add as many blocks as needed by inserting `else if` blocks before the final `else`. In the first rule our action depends on the procedure location. The target pressure is 45 mmHg for the knee joint and 60 mmHg for the elbow.

E. Rule Execution and Usage during Surgery

The implementation of the rule engine is realized with the *Esper*[7] framework for *Complex Event Processing (CEP)*. As soon as a complete and valid context is available our *Rule Execution Engine* checks all available rules if they apply to the current context. When a rule is enabled, the event description from our rule language is translated into the

[7]https://github.com/espertechinc/esper

Esper Event Processing Language, representing a query that is continuously evaluated on the data stream. Every time the data collection (SDC Consumer) receives a new application layer message, the content is put into an event object that is passed into the global event stream. Whenever *Esper* detects an event of interest, the appropriate rule is executed by our interpreter, which checks the conditions and executes the first action where the condition is met.

V. Conclusion

In this paper we demonstrated the use of IEEE 11073 Service-oriented Device Connectivity (SDC) *Device Specializations* to create a situation-aware assistive system supporting the caregivers with recommendations or even autonomous (re-)configuration for medical device parameters. The comprehensive collection of data during surgery enables data driven analysis. Thus, it serves as a basis to create deterministic and approvable rules according to the Event-Condition-Action (ECA) pattern for assistive systems, reducing stress of the OR staff, improving patient's safety and clinical outcome.

We have chosen an interpreter-based approach for our *Rule Execution Engine*. This allows for a one time approval as medical device while still being able to continuously add new rules for parameter recommendation. Thus, newest analysis results can become part of productive systems much faster. In our demonstrator we have shown the principal applicability of our approach along with the smart surgical pump use case. It highlights that rule-based analysis of cross-manufacturer and cross-medical-domain exchanged information can lead to safe and effective assistance in the OR.

VI. Future Work

Proving the concept in a study in a real clinical environment would be one of the next steps. Prerequisite, however, are approved manufacturer-independent interconnected medical devices based on IEEE 11073 SDC.

In our demonstrator a device is responsible for its own assistive function. However, as mentioned, the concept is also suitable for a dedicated component for data collection. This approach can be extended to have a dedicated component implementing assistive functions for a whole medical device ensemble. This could improve parameter recommendation as it provides a global view. Thus, all devices have to allow external control and provide suitable risk management. Further, synchronization and priority management challenges arise, to deal with recommendations from multiple (device internal and external) sources. Additionally, a dedicated component does not have prior knowledge of specialized device functions. Hence, the *DevSpecs* have to prove their capability to describe specific functionalities in a machine-interpretable manner.

Another future aspect is the interoperability of rules themselves. For now we had a rule language and an appropriate interpreter developed on our own. However, extracted knowledge from collected data is not limited to a specific device. Respective rules might be of interest for many other devices. To enable free sharing of knowledge, the rule syntax and semantics have to be standardized. The W3C Rule Interchange Format (RIF) might serve as a baseline for future work.

References

[1] World Health Organization (WHO), "Increasing complexity of medical technology and consequences for training and outcome of care," 2010. [Online]. Available: https://apps.who.int/iris/handle/10665/70455

[2] L. S. Feldman, P. Fuchshuber, D. B. Jones, J. Mischna, and S. D. Schwaitzberg, "Surgeons don't know what they don't know about the safe use of energy in surgery," *Surgical Endoscopy*, vol. 26, no. 10, pp. 2735–2739, 2012. [Online]. Available: https://dx.doi.org/10.1007/s00464-012-2263-y

[3] A. E. W. Johnson, M. M. Ghassemi, S. Nemati, K. E. Niehaus, D. Clifton, and G. D. Clifford, "Machine Learning and Decision Support in Critical Care," *Proceedings of the IEEE*, vol. 104, no. 2, pp. 444–466, 2016. [Online]. Available: https://doi.org/10.1109/JPROC.2015.2501978

[4] K. Lesh, S. Weininger, J. M. Goldman, B. Wilson, and G. Himes, "Medical Device Interoperability-Assessing the Environment," in *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*. IEEE, 2007, pp. 3–12. [Online]. Available: http://dx.doi.org/10.1109/HCMDSS-MDPnP.2007.22

[5] H. Lemke and M. Vannier, "The operating room and the need for an IT infrastructure and standards," *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, no. 3, pp. 117–121, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11548-006-0051-7

[6] B. Andersen, S. Baumhof, and J. Ingenerf, "Service-Oriented Device Connectivity: Device Specialisations for Interoperability," in *Studies in health technology and informatics: MEDINFO 2019 - The 17th World Congress of Medical and Health Informatics*, Lyon, aug 2019, pp. 509 – 511. [Online]. Available: https://doi.org/10.3233/SHTI190274

[7] M. Kasparick, B. Andersen, H. Ulrich, S. Franke, E. Schreiber, M. Rockstroh, F. Golatowski, D. Timmermann, J. Ingenerf, and T. Neumuth, "IEEE 11073 SDC and HL7 FHIR - Emerging Standards for Interoperability of Medical Systems," *International Journal of Computer Assisted Radiology and Surgery (2018) 13(Suppl 1): CARS 2018 - Computer Assisted Radiology and Surgery: Proceedings of the 32nd International Congress and Exhibition*, vol. 13, no. Suppl 1, pp. 135–136, 2018. [Online]. Available: https://doi.org/10.1007/s11548-018-1766-y

[8] M. Kasparick, B. Andersen, S. Franke, M. Rockstroh, F. Golatowski, D. Timmermann, J. Ingenerf, and T. Neumuth, "Enabling artificial intelligence in high acuity medical environments," *Minimally Invasive Therapy & Allied Technologies*, pp. 1–7, 2019. [Online]. Available: https://doi.org/10.1080/13645706.2019.1599957

[9] M. Kasparick, M. Schmitz, B. Andersen, M. Rockstroh, S. Franke, S. Schlichting, F. Golatowski, and D. Timmermann, "OR.NET: A Service-Oriented Architecture for Safe and Dynamic Medical Device Interoperability," *Biomedical Engineering / Biomedizinische Technik*, vol. 63, no. 1, pp. 11–30, 2018. [Online]. Available: https://doi.org/10.1515/bmt-2017-0020

[10] S. Schlichting and S. Pöhlsen, "An architecture for distributed systems of medical devices in high acuity environments - A Proposal for Standards Adoption," in *IEEE 11073 / HL7 Standards Week*, San Antonio, Texas, USA, 2014. [Online]. Available: www.hl7.org/documentcenter/public/wg/healthcaredevices/20140116 An architecture for distributed systems of medical devices in high-acuity environments.pdf

[11] A. Benkmann, E. Smolinski, T. Jeinsch, P. Westerhoff, T. Wustrack, and H. Hommel, "Concept of iterative optimization of minimally invasive surgery," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2017, pp. 443–446. [Online]. Available: https://doi.org/10.1109/MMAR.2017.8046868

[12] E. Smolinski, A. Benkmann, W. Drewelow, T. Jeinsch, and H.-J. Cappius, "Observer-based controller design for the minimally invasive surgery," *Current Directions in Biomedical Engineering*, vol. 4, no. 1, pp. 41–44, 2018. [Online]. Available: https://doi.org/10.1515/cdbme-2018-0011

[13] M. Bialke, P. Penndorf, T. Wegner, T. Bahls, C. Havemann, J. Piegsa, and W. Hoffmann, "A workflow-driven approach to integrate generic software modules in a Trusted Third Party," *Journal of Translational Medicine*, vol. 13, no. 1, p. 176, 2015. [Online]. Available: https://doi.org/10.1186/s12967-015-0545-6