A Monitoring System for Operating Theaters at Heidelberg University Hospital

First Experiences Implementing Predictive Analytics Tools in a Clinical Routine Setting

Oliver Klar
Department of Medical Information Systems
University Hospital Heidelberg
Heidelberg, Germany
oliver.klar@med.uni-heidelberg.de

Gerd Schneider
Department of Medical Information Systems
University Hospital Heidelberg
Heidelberg, Germany
gerd.schneider@med.uni-heidelberg.de

Rasim Atakan Poyraz
Department of Medical Information Systems
University Hospital Heidelberg
Heidelberg, Germany
atakan.poyraz@med.uni-heidelberg.de

Oliver Heinze
Department of Medical Information Systems
University Hospital Heidelberg
Heidelberg, Germany
oliver.heinze@med.uni-heidelberg.de

***Abstract* - The rise of Artificial Intelligence (AI) is ubiquitous. In healthcare it is seen as a key technology supporting clinicians in their daily routine. The PART research project (Predictive Analytics of Robustness Testing) aims to develop an AI driven, vendor independent monitoring system, which has the focus on system monitoring, profitability analysis, and predictive maintenance of networked medical devices in a clinical environment. However, before working on AI driven monitoring solutions at Heidelberg University Hospital, we experienced a variety of difficulties according to networked medical devices, data acquisition, standards and protocols, and device interfaces, which must be addressed first. This paper stresses those difficulties and presents a monitoring system of networked medical devices from one operating theater at Heidelberg University Hospital. Continuous data streams of laparoscopic devices out of the surgery room are ingested into the system and analyzed in real-time. The results are stored in an on-premises data store and visualized according to profitability analysis and system monitoring in a dashboard. Further, an outlook is giving including the transformation of the presented monitoring system into the Medical Data Integration Center (MeDIC) of the Heidelberg University Hospital in the future and the connection of more surgery theaters.**

Keywords - clinical artificial intelligence; artificial intelligence in healthcare; medical device monitoring; real-time data stream processing; predictive maintenance; Apache Kafka; Apache Flink; Elasticsearch; Kibana.

I. BACKGROUND

The PART research project (Predictive Analytics of Robustness Testing) aims to develop an AI driven, vendor independent monitoring system, which has the focus on system monitoring, profitability analysis, and predictive maintenance of networked medical devices. However, at Heidelberg University Hospital we experienced a variety of difficulties building such a monitoring system including data acquisition, standards and protocols and device interfaces [1]. Dealing with those circumstances in this work, we present a flexible and extendable pipeline architecture for ingestion, processing, and storage of medical device data. As a first approach we focus on monitoring laparoscopic medical devices of our project partner Karl Storz GmbH & Co. KG from one operation theater. Use cases for this system from the perspective of the Heidelberg University Hospital are:

- Profitability Analysis

Acquisition and operation of a vast number of medical devices is expensive. Often several devices of the same type are used in the same clinic. So far, there are no numbers about device usage and whether the current number of devices is required. The monitoring system should collect key figures such as utilization and operating hours, which are then economically analyzed.

- Online Inspection

To conduct inspections on medical devices, such as safety related checks and metrological checks, it is necessary to put them out of operation. Those checks must be done in fixed predefined intervals and are known as preventive maintenance [2]. To reduce this costly downtime, the goal of the monitoring system is to go from preventive maintenance to predictive maintenance by using predictive analytics tools to determine when maintenance actions are required automatically.
Analyzing vast amounts of medical device data continuously, machine learning (ML) algorithms should

help putting medical devices out of order only when thresholds are overstepped, or critical errors occur. This should help avoiding unnecessary down-times, optimizing the maintenance schedules, and reducing maintenance costs.

- System Monitoring

To exchange data with clinical systems like a Picture Archiving and Communication System (PACS) [3] or Hospital Information Systems (HIS), medical devices are interconnected more often. To guarantee stability for such a growing networked system and manage a reliable exchange of data between devices and other IT infrastructures, a superior overall system is required for monitoring and alerting.

This work presents first experiences in designing and implementing such a vendor-independent monitoring system, facing the real world setting of a university hospital.

In Section II we describe the challenges we face implementing such a monitoring system. The PART pipeline architecture of our monitoring system is the focus of Section III. The dashboard of the monitoring system is presented in Section IV and we conclude with discussion and outlook in Section V.

II. Challenges & Objectives

There are several hurdles according to healthcare devices, medical device data, data processing, and data protection that must be taken in advance of realizing a powerful monitoring system. First, one must address problems caused by heterogeneity. Devices are mostly, due to reasons of independence, from different manufacturers. This ranges from infusion pumps to the latest CT or MRI scanners. Even though there are standards for networked medical devices in operation rooms (e.g., IEEE 11073), they are only implemented and promoted by a few manufacturers [4]. The communication of the most medical equipment works mostly via proprietary interfaces and protocols, and manufacturers are very reluctant disclosing those interfaces, or implementing given standards. Even if monitoring of those medical devices is possible in general, integrated sensors like in the Philips e-Alert System of MRI scanners [5] or in industrial environments, which produce data feasible according to predictive maintenance, are rather seldom and hence can restrict available information to log data of the machines [6]. Further, expensive devices like CT scanners usually have extensive maintenance contracts, which include that maintenance, repair, and service may only be performed by a service engineer of the manufacturer itself. Collecting relevant data from such devices, e.g., getting information about the condition and operating status is demanding.

Those manufacturers come up with own solutions for monitoring the medical device fleet [7][8]. They provide the customer with key performance indicators which help e.g., identifying over- and under-utilized devices, balancing the product use and lessen the strain placed on individual medical devices [9].

As the operator of all medical equipment, Heidelberg University Hospital ends up maintaining a monitoring system for each manufacturer, which is not expedient.

Another potential source of data, delivering information about the status of medical equipment, could be the usage of IoT sensors. However, gathering data by additional attached sensors in a sterile environment like an operating room is under serve restrictions due to aspects like patient safety. Hence, the situation is barley comparable to an industrial production line where predictive maintenance is quite common for prevention device failures.

Data quality is a problem since several decades. In contrast to big data, machine learning goes through with a different set of data quality concerns. The three components of ML algorithms are model representation, measures of valuating model accuracy, and methods for searching the best ML model [10]. Since these three components are highly related to each other, data quality for ML is very complex. One of the biggest concerns in big data is missing data as well as the well-structured datasets.

In PART the current question is not which data mining algorithms fit the most for our needs, the question is where the data is coming from in the first place. Therefore, we are looking in all directions and started working with simulated device data as well as getting familiar with the data mining approaches. Another subject is data protection and privacy. By monitoring devices, collecting, and analyzing data, it could be possible to draw conclusions about patients, treatment, and the work of clinical personnel itself. This is sometimes seen very critically by the clinic staff and requires a close examination and further steps like anonymization of the data.

Finally, an important point to mention is that according to predictive maintenance of networked medical devices, failures are quite often due to simple reasons like dropping the device or too much moisture when cleaning in a clinical environment. Two issues which are hard to handle by analyzing device data.

Although devices in clinical environments produce a high volume of data, it is quite challenging, as described above, to access, evaluate, and generate added value from this data treasure.

Keeping those challenges in mind in our first approach we have focused on descriptive analytics towards the development of an AI driven monitoring tool including ingestion, real-time analytics, storage, and visualization of networked medical device data.

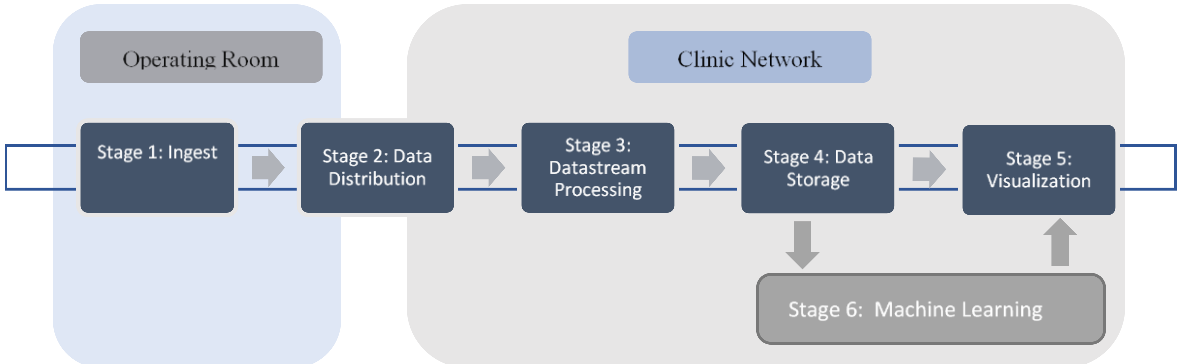

Figure 1. PART pipeline architecture with its 6 stages

III. PART PIPELINE ARCHITECTURE

As mentioned in Section II, there are several hurdles to overcome implementing a monitoring system in a clinical environment. Hence, the focus was to create a generic architecture which can be adapted and extended for future needs. This includes ingesting data from arbitrary sources, scalability, high availability, and failure safety.
In the following sections the PART pipeline architecture with its individual stages is described in detail, see Figure 1.

A. Stage 1: Ingest

Stage 1 described here is responsible for ingesting medical device data into our pipeline architecture. As a first step we connected one operating room and collected data from medical devices of our project partner Karl Storz GmbH & Co KG [11]. All devices are related to laparoscopic surgery like insufflators, endoscopic cameras, and light sources. At Heidelberg University Hospital those devices are located on a mobile cart which makes it possible to move them between surgery rooms. When the cart is moved into a surgery theater and plugged in, all devices start up and start sending records of data in 2 second intervals out of the surgery room to a specific Karl Storz machine, called Interface Control, over the Storz Communication Bus (SCB).

B. Stage 2: Data Distribution

For broadcasting the device data within the PART pipeline architecture, the distributed streaming platform Apache Kafka [12] is used, see Figure 1. It can store huge amounts of records in a fault-tolerant durable way and processes streams as they occur. Apache Kafka has three main components which are producers, brokers, and consumers. It is comparable to a message queue or an enterprise messaging system.
The interface control, see Section A, sends the device messages from the operating room via the serial interface RS232 to a machine in a technical room, located close to the surgery theater. On this machine the records of machine data are transformed into Fast Healthcare Interoperability Resources (FHIR) [13] formatted JSON [14] objects.
FHIR is a standard describing data formats and elements and an API for exchanging electronic health records created by HL7. One of its goals is to ease the interoperation between health care systems. It provides automatic and detailed electronic data capture of operational device data and offers data formats such as JSON, XML, and RDF.

Those objects are published as streams of records by the Kafka producer via TCP and Kafka protocol for subscription by consumers in the clinic network, see Figure 1. The streams of records are grouped in categories called topics. Each record consists of a key, a value, and a timestamp.
As shown in Table 1 currently 12 topics exist from four different devices including endoscopy camera (endocam), light source, operation room light and insufflator.

Table 1. Device topics consumed and processed in the PART pipeline architecture

Topic	***Unit***
INSUFFLATOR ACTUAL FLOW	liter per minute (l/min)
INSUFFLATOR ACTUAL PRESSURE	millimeter Mercury column (mm [Hg])
INSUFFLATOR INSUFFLATION ON	0x00 off 0x01 on
INSUFFLATOR GAS VOLUME	milliliter (ml)

Topic	*Unit*
INSUFFLATOR TARGET FLOW	liter per minute (l/min)
LIGHT SOURCE INTENSITY	percentage (%) [0-100]
LIGHT SOURCE STANDBY	0x00 off 0x01 on
ENDOCAM BRIGHTNESS	Low, Medium, High, Peak, Small Scope A, Small Scope B
ENDOCAM ENHANCEMENT	Off, Low, High, Fiberscope Filter A, Fiberscope Filter B
ENDOCAM SHUTTER	Auto, 1/50, 1/85, 1/125, 1/175,1/250, 1/350, 1/500, 1/700, 1/1000, 1/1500, 1/2100, 1/2800, 1/4000, 1/6000, 1/8500, 1/12000,1/17000
ORLIGHT1 INTENSITY	percentage (%) [0-100]
ORLIGHT2 INTENSITY	percentage (%) [0-100]

C. Stage 3: Data Stream Processing

As operator, the Heidelberg University Hospital wants to know when a device is used, how often it is used, and what is the ratio between hours of operation and device utilization. Those numbers help the management planning device replacements and optimizing the procurement process of the medical equipment.

As stated in Section III.B Apache Kafka is used to handle the vast amount of streams of records coming from the medical devices out of the surgery theater. As the next stage in the PART pipeline architecture Apache Flink [15] then consumes and analyses those streams of records in real-time.

The data stream processing stage, described here, focuses on specific continuous streams of records of the Karl Storz devices, which indicate whether the device is online or in standby, see Section III.B. Each record of those streams contains either a value equal to zero (standby) or equal to one (online). Hence, a device is switched on when the values of the records turn from zero to one and is offline if no data is sent from the device at all.

For all medical devices currently connected to the system, an Apache Flink program, called job, is implemented analyzing the incoming data streams in real-time.

All jobs are organized and managed as a standalone cluster on a machine in the clinic network and working as consumers subscribing to the specific topics sent by the Kafka producer to the Kafka broker. Each job can be separated into several steps, described subsequently.

For each stream:

- Transform
  Each incoming JSON structured record is disassembled and mapped onto a Flink container structure, which keeps all relevant parameters for the analytics.
- Split
  After the mapping, the new stream is split into a main stream (depicted in Figure 2 as the blue graph), containing all records of data and a sub stream (shown as the red curve in Figure 2), containing records of values equal to one. Note that the split of the sub stream is only done when the device is switched on.
- Process
  To work on infinite streams, Flink offers the concept of windows [16]. Session windows split the stream into chunks of data with finite size by sessions of activity. This is the case every time when a device is plugged in, and data is ingested into the pipeline architecture. In Figure 2 there are two session windows for the main stream, colored in blue, and three session windows for

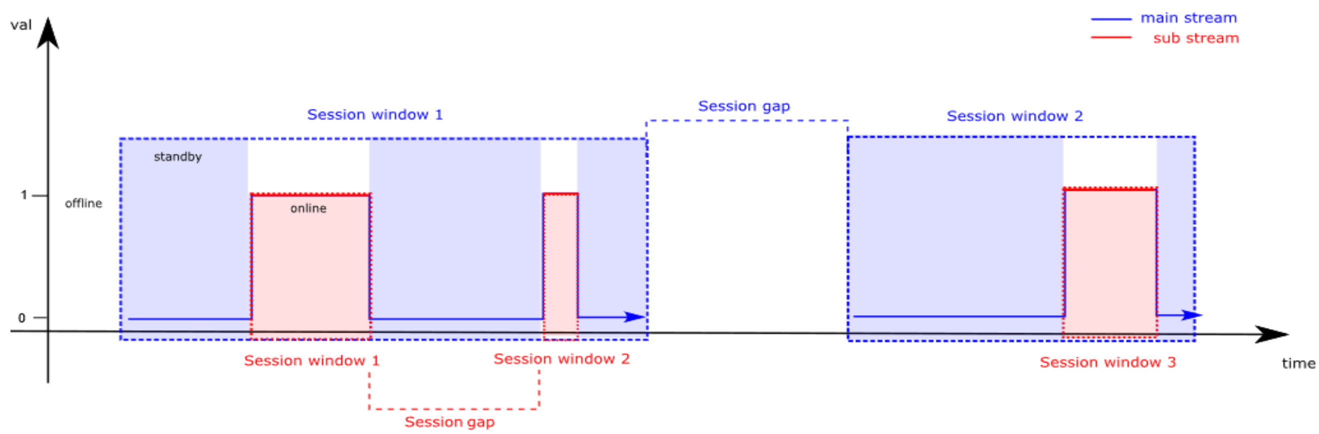

Figure 2. Data stream processing. The original stream is split into two streams. The main data stream colored in blue contains all incoming records. The sub stream colored in red contains records with values equal 1. Each partitioned into session windows separated by a predefined session gap. The processing is done on each individual window.

Figure 3. Insufflator dashboard. Key performance indicators of the observed device within a two-month interval

the sub stream, colored in red. Those windows do not have a fixed start or end. Flink closes those windows after a period of inactivity and assigns subsequently coming events to the next window. This period can be a predefined fixed interval or dynamically extracted and is called session gap.

To implement session windows, it is required to conduct the concept of event time [17] in the Flink application. This guarantees that all incoming events are ordered time wise and are processed when they happen. Even with delays, e.g., network traffic or how fast the stream is processed by the application, it yields to deterministic results. Contrary to this, the concept of Flink processing time would mean processing the stream of records when they receive the application.

Once Flink closes a session window, a process function is applied on that finite stream of data. It computes the start, end, and the duration time for the sub stream, indicating the time the device is online (depicted in Figure 2 red graph) and the values start, end, and operation time of the main stream (depicted in Figure 2 blue graph), indicating the whole period of time the device is plugged in and sending data. Note that the duration time of a device is always equal or less then the operation time.

Those results are then issued as new events separately, narrowing down an infinite stream of data, on the relevant information we were looking for.

- Store

  The results then are mapped back from the Apache Flink data structure into a valid JSON object, required by Elasticsearch [18] for storage in the PART pipeline architecture, see Figure 1. A Flink connector is used to create an Elasticsearch sink, writing the results via REST to the PART data store.

D. *Stage 4: Data Storage*

As stated in Section I, the monitoring system for networked medical devices at Heidelberg University has several requirements according to a data store. In the presented architecture medical equipment of Karl Storz is sending JSON formatted data via Kafka protocol to the database. However, to handle other types of data from different manufacturers, e.g., not related to surgery theaters, there is a need for a flexible system, which offers tools for ingestion of arbitrary sources of data. Connecting more of those devices gradually will yield to surging amounts of data. The data store must have the ability to adapt to changing conditions as needed overcoming problems of access times and failure safety.

The Elastic Stack offers open-source solutions for those mentioned requirements. For ingesting data there are lightweight data shippers called Beats [19] and the data collection engine Logstash[20]. Storage, search, and simple analytics is done by Elasticsearch, which runs as a cluster and scales horizontally. It is capable storing complex data structures represented as JSON objects by using RESTful APIs [21].

In the PART pipeline architecture, all observed data streams and the results of the real-time analytics are stored in Elasticsearch into an index. An index is comparable to a data table in the concept of relational database systems (RDBMS). For the index, an underlaying schema handles the mapping between JSON data fields and Elasticsearch

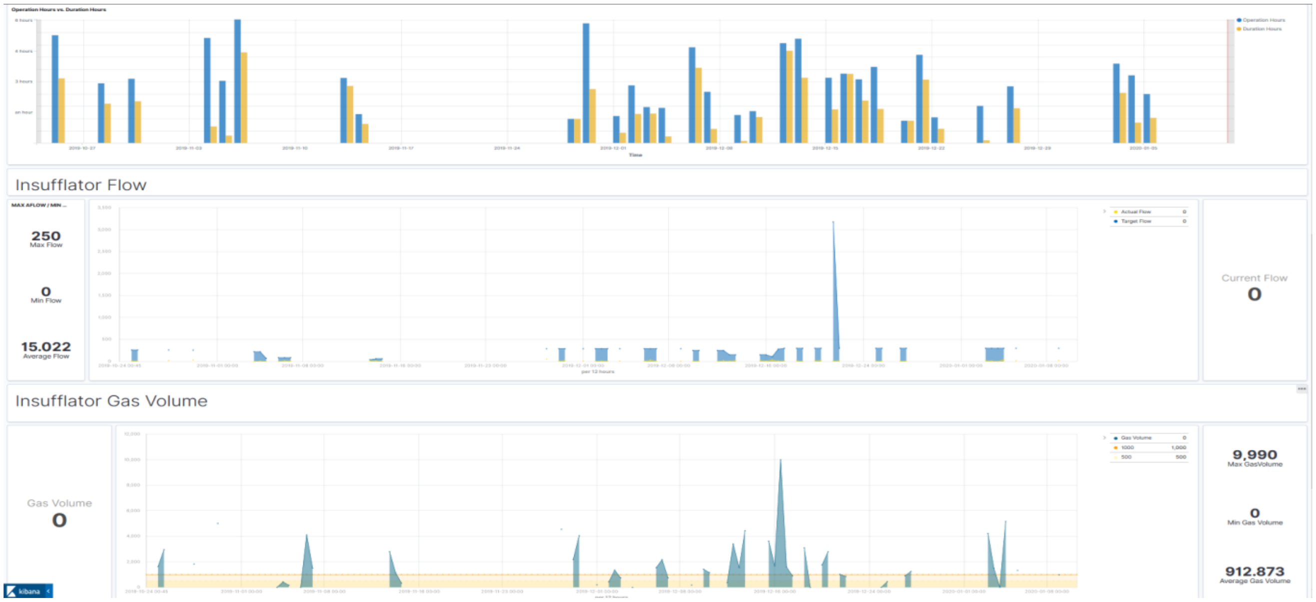

Figure 4. Insufflator dashboard. Recorded process values of the insufflator for a period of two month

data types during the ingestion. The data is then used for further analysis and visualization in stage 5.

E. Stage 5: Visualization

Kibana [22] is a data exploring and visualization tool and runs on top of Elasticsearch. It is part of the Elastic Stack and helps interacting with huge amounts of stored sensor data. It provides tools for searching, analyzing, and presentation and is used in a wide field of applications, e.g., Industrial Internet of Things (IIoT) [23][24].

Easy understandable, clearly structured visualizations embedded into good organized dashboards help conceiving the data more quickly and give first insights and hints about potential problems, e.g., device failures, even without applying sophisticated AI algorithms. Further, such dashboards help keeping track of a growing number of networked medical devices and increase the visibility and status of each individual one. Hence, Kibana is used in our PART pipeline architecture for descriptive analytics by creating dashboards combining simple histograms, line graphs, and more advanced time series aggregations for the monitored medical equipment.

F. Stage 6: Machine Learning

The problems with technical equipment during laparoscopic surgeries have been analyzed in early years which show us that there were different issues accordingly [25]. We aim to carry the current situation to a next step which would be beneficial to implement machine learning model on device data to be able to decrease the technical problems during the surgeries.

In machine learning, understanding the data is a key. Therefore, Jupyter Notebook [26] is used to analyze the data for machine learning implementation. In this phase of the architecture, the data from Elasticsearch is taken to Jupyter Notebook by using Pandasticsearch library [27]. In this way, we can create Pandas data frame for data analysis as well as implement a machine learning algorithm.

To be able to train machine learning algorithms, the data should be separated into a clean, annotated, well-structured dataset. These datasets then will be stored in MongoDB [28] as train, test, and validation datasets separately. With that, we can train our machine learning algorithm on training datasets and test the accuracy on test datasets. The aim of implementing machine learning algorithms will be used for the use case 2 “online inspection”, to find the anomalies of the devices through their data. Therefore, we assume that unsupervised machine learning techniques will fit the most to find anomalies and outliers.

IV. DASHBOARD

Our goal was to create an intuitive dashboard for the monitoring tool, including all medical devices in the surgery theater. As stated in Section III.E we use Kibana for

Figure 5. Insufflator Status. Device is offline, in standby or online

visualization of laparoscopy device data. For each device type, e.g., insufflators, one monitoring dashboard was created by assembling different kinds of visualizations.
Subsequently the PART dashboard (see Figure 3, Figure 4) with focus of profitability analysis and system monitoring for the insufflator is shown.
The dashboard has a date range field which enables looking at specific time ranges getting different insights of the key performance indicators (KPI). This makes it possible to examine historical periods of time, e.g., the last two years, shorter periods, e.g., minutes or even real-time data.

The presented dashboard is divided into two sections. Figure 3 shows the top section, including visualizations and numbers according to the profitability analysis, and the system monitoring for a time period of two month exemplarily. Followed by a section showing medical device data related to the process values, including the insufflator pressure, flow and the gas volume for the same time range, depicted in Figure 4. Subsequently, some of the visualizations and graphs of the dashboard are described in more detail. Figure 5 shows the top section of the dashboard which is the insufflator status. It plots the stream of records grouped by the topic INSUFFLATION ON, see Table 1. Note that in Figure 3 and Figure 4 a much wider range of time is selected, hence more data is visualized. In Figure 5 we basically zoomed into the data timewise, showing a two-hour period of recorded insufflator data for better understanding. The graph shows the usage of the insufflator during that time period. Three scenarios are possible:

- offline - no data is available
- standby - the device is sending records of zero data
- online - the device sending records with values of one

The x-axis shows the time, the y-axis has two scales. On the left-hand side it shows the status, zero for standby, one for online. On the right it shows the time in hours according to the device usage and operation time.
If there is no data shown on the timeline, which is the case at the very beginning and the end of the graph, the device is switched off. In the time period where the graph shows values of zero, the device is in standby. This is the situation usually right before a surgery, where the cart is already moved into the surgery room and plugged in, see Section III.A. When the graph values turn to one, it indicates that the device is now being used. After a short period of time it was switched off again. In this visualization a laparoscopic intervention is shown.
When switched off, Apache Flink, see Section III.C, calculates the operation and the usage time indicated by the yellow and the blue dot (see Figure 5). Operation time

Figure 6. Insufflator KPIs – duration time, operation time and usage counter

implies the whole time the device is powered on. Usage time represents the time the device is actually used. The difference between those measures is encoded in their height. In our example there is a long time where the device is in operation but only a small period of time where it is actually used. From an economic perspective and as the operator of thousands of devices at Heidelberg University Hospital those numbers are analyzed according to potential optimization.
In the section below, the status graph (see Figure 6) of the PART dashboard some KPIs, e.g., the device usage- and operation time of the insufflator, are shown. As depicted in Figure 6 the very left panel shows the usage of the device, in our case eight minutes, and the total operation time on the right, which is about two hours. In the middle of this section a panel shows the time when the device was online the last time, named as last seen, and a usage counter for the specific time frame. Note that this information can be exported as a report.
To give the user an overview about those numbers on a daily basis, a calendar was created for usage and operation time, see Figure 7. The heat map visualization of Kibana does a color coding from white to blue according to the duration, where blue indicates a longer usage duration. This utilization distribution immediately shows peaks and helps finding patterns in device and operating room usage, which can help optimizing utilization of medical equipment by balancing the product use and hence could yield to less inventory.

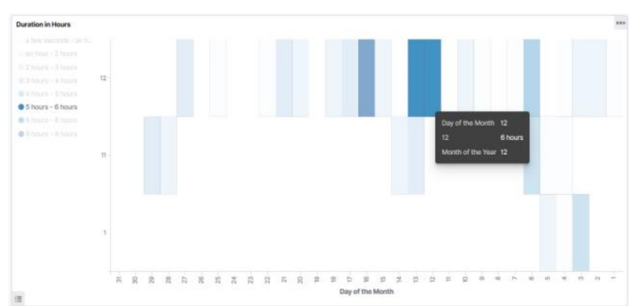

Figure 7. Calendar Visualization: Heatmap of the device usage per day. Dark blue indicates longer usage times

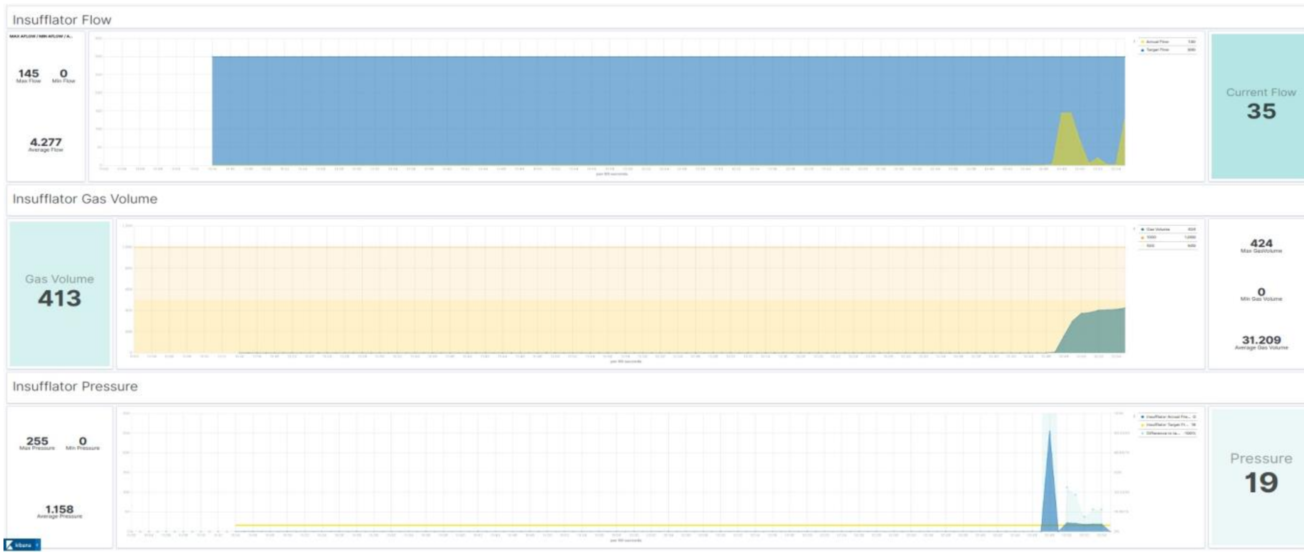

Figure 8. Detailed insufflator process values during a surgery. Flow, gas volume and pressure visualizations in the PART dashboard

The lower part of the insufflator dashboard has three panels each divided into three sections, see Figure 8. The middle section of each panel contains a graph drawing the process values. On the x-axis of this graph the time is indicated. On the far right, the graph shows the latest record of the data stream received by the monitoring tool. There is one section depicting minimum, maximum, and the average of the process values. One field shows the latest value recorded in Elasticsearch as a number, which is identical to the far-right value of the graph.

At the top of Figure 8 the insufflator flow is shown. The blue area depicts the target, the yellow graph the actual flow. Currently, the latter has the value 35. The progress over time of the insufflator gas volume is depicted in the middle. The accumulated value is 413. At the bottom, the user can monitor the pressure of the insufflator. The yellow line indicates the target pressure while the blue curve represents the actual pressure, currently set to 19, during a surgery. Those visualizations enable the user to monitor the devices and the whole surgery theater as being used in real-time detecting potential anomalies in the data only by descriptive analytics.

V. Discussion and Outlook

The goal of PART is to build an AI driven monitoring system for networked medical devices. This system should be vendor independent. As shown in Section II, there are several hurdles to overcome reaching this goal. Those circumstances made us focus building up a generic architecture which is a flexible, easy to extend infrastructure for ingesting, analyzing, storing, and visualizing medical device data. We started ingesting data out of one surgery room from laparoscopy related devices of one device manufacturer. We are analyzing those streams of records in real-time according to the use case profitability analysis and built a dashboard showing those results for the monitored medical devices.

Still, at Heidelberg University Hospital, our strategy is to add more devices of different manufacturers gradually. Hence, we work closely together with other device manufacturers, not only related to operating rooms, e.g., patient monitoring.

Further, we continue to work on the architecture to be prepared for growing amounts of medical device data in the future. This is done by moving the PART pipeline architecture into the Medical Data Integration Center (MeDIC) of the Heidelberg University Hospital. Including the transformation of each particular stage of the PART pipeline architecture from standalone components to orchestrated clusters. Making the monitoring system more robust against performance issues and failure safety. Additionally, a monitoring system for the PART pipeline architecture with its components should be implemented to handle the complexity of the monitoring system itself.

Another upcoming task is to connect more integrated operating theaters [15] with its laparoscopic devices from Karl Storz including collection, analyzing, and visualizing the retrieved data as presented here. Those extensions and the goal to support a multitude of devices from different vendors, requires a complete redesign of the PART dashboard in the future.

There is still a lot of information in the data which is not appropriately extracted and visualized. Hence, extending the dashboards by additional visualizations is planned. For example, including distribution numbers when specific devices are used during a day, month or year or adding a counter which shows since when a device is online.

With the experience made, we have lowered our aspiration towards an AI driven system for predictive maintenance and hence focused on a combination of descriptive analytics and real-time stream processing as a first approach. However, we still have the ambition to push this topic further by extending the PART pipeline architecture with machine learning tools and deepen the research on the other use cases. Therefore, we look at the data comprehensively to be able to create a well-structured dataset that can be used straightforwardly for further implementations. To be able to succeed this, we work with surgeons closely to annotate data.

Establishing such a generic monitoring system at Heidelberg University Hospital would have several benefits for the clinic. It should make the complete IT infrastructure more robust and stable. Detecting problems of networked devices in early stages, by real-time alerts or in a best-case scenario before they are going to happen, saves maintenance time and costs. Further, with workload statistics of the devices one gets a good tool for tracking the usage and can adapt the inventory accordingly.

VI. Acknowledgment

This work is funded by the German Federal Ministry of Education and Research under reference number 16KIS0785 as part of the PART project. We thank all partners within the PART project specifically Karl Storz GmbH & Co. KG supporting us with their knowledge and medical equipment.

VII. References

[1] O. Klar, R. Dees, G. Schneider, A. R. Poyraz and O. Heinze, “First Experiences Implementing Predictive Analytics Tools in a Clinical Routine Setting” in Proc. eTELEMED 2019, The Eleventh International Conference on eHealth, Telemedicine, and Social Medicine, pp. 114-115.

[2] T. Carvahlo, F. Soares, R. Vita, R. d. P. Francisco, J. P. Basto and S. G. S. Alcala, "A systematic literature review of machine learning methods applied to predictive maintenance," Computers & Industrial Engineering, 2019.

[3] R. H. Choplin, J. M. Boehme and C. D. Maynard, "Picture archiving and communication systems: an overview." RadioGraphics 12(1): pp. 127-129, 1992.

[4] Draeger, Medical device connectivity based on IEEE 11073 SDC. [Online]. Available from: https://www.draeger.com/en_me/Hospital/Acute-Care-Insights/Service-Oriented-Device-Connectivity 2020.01.28

[5] Philips e-Alert. [Online]. Available from: https://www.philips.co.uk/healthcare/product/HC895000/philips-ealert-alerting-solution-for-mri-systems 2020.02.12

[6] R. B. Patil, M. A. Patil, V. Ravi and S. Naik, "Predictive modeling for corrective maintenance of imaging devices from machine logs," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, 2017, pp. 1676-1679.

[7] Siemens Teamplay. [Online]. Available from: https://www.siemens-healthineers.com/de/digital-health-solutions/digital-solutions-overview/service-line-managment-solutions/teamplay 2020.01.27

[8] Philips PerformanceBridge. [Online]. Available from: https://www.philips.de/healthcare/services/performance-services/performancebridge 2020.01.27

[9] Stryker Smart Equipment Management. [Online]. Available from: https://www.stryker.com/us/en/surgical/services/smart-equipment-management.html 2020.01.27

[10] V. Gudivada, A. Apon, and J. Ding. ”Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations". In: International Journal on Advances in Software 10.1 (2017), pp. 1 – 20

[11] Karl Storz OR1. [Online]. Available from: https://www.karlstorz.com/de/de/karl-storz-or1.htm 2020.01.27

[12] Apache Kafka. [Online]. Available from: https://kafka.apache.org/ 2020.01.27

[13] B. Franz, A. Schuler and O. Krauss. "Applying FHIR in an integrated health monitoring system." EJBI 11.2, 2015, pp. 51-56.

[14] JSON. (2020). [Online]. Available from: https://www.json.org/json-en.html 2020.01.27

[15] Apache Flink. [Online]. Available from: https://flink.apache.org/ 2020.01.27

[16] Apache Flink Windows. [Online]. Available from: https://ci.apache.org/projects/flink/flink-docs-stable/dev/stream/operators/windows.html 2020.02.04

[17] Apache Flink Event Time. [Online]. Available from: https://ci.apache.org/projects/flink/flink-docs-stable/dev/event_time.html 2020.02.04

[18] Elastic. [Online]. Available from: https://www.elastic.co/de/ 2020.01.27

[19] Elasticsearch Beats. [Online]. Available from: https://www.elastic.co/de/beats 2020.02.06

[20] Elasticsearch Logstash. [Online]. Available from: https://www.elastic.co/de/logstash 2020.02.06

[21] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf. [Accessed 29 1 2020].

[22] Kibana, (2020). [Online]. Available from: https://www.elastic.co/de/kibana 2020.01.27

[23] IIoT with the Elastic Stack. [Online]. Available from: https://www.elastic.co/de/blog/industrial-internet-of-things-iiot-with-the-elastic-stack 2020.01.27

[24] M. Bajer, ”Building an IoT Data Hub with Elasticsearch, Logstash and Kibana. *Building an IoT Data Hub with Elasticsearch, Logstash and Kibana”,* 2017.

[25] E.G.G. Verdaasdonk, L.P.S. Stassen, M. van der Elst, T. M. Karsten and J. Dankelman, “Problems with technical equipment during laparoscopic surgery. An observational study“ , Surg Endosc (2007), pp. 275-279.

[26] Jupyter Notebook. [Online]. Available from: https://jupyter.org/ 2020.01.27

[27] Pandasticsearch. [Online]. Available from: https://pandasticsearch.readthedocs.io/en/latest/index.html# 2020.01.27

[28] MongoDB. [Online]. Available from: https://www.mongodb.com/ 2020.01.27

[29] M. Kasparick et al. “New IEEE 11073 standards for interoperable, networked point-of-care Medical Devices,” in Proc. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2015, pp. 1721-1724