Robust and Semi-automatic Electronic Health Record Dissemination Using the Devices Profile for Web Services

David Gregorczyk, Timm Bußhaus, Stefan Fischer
Institute of Telematics
University of Lübeck, Germany
{gregorczyk, busshaus, fischer}@itm.uni-luebeck.de

***Abstract*—Much conceptual work was done on medical device interoperability. Though many architectures, terminologies, and standards exist today, they have not achieved the desired acceptance yet, or provide ambiguous implementation directives. Recent research has shown that the Devices Profile for Web Services (DPWS) is suitable for device interconnectivity. Since DPWS was made to support a wide range of device types, it lacks concrete message process flows to enable added value in clinical environments. Hence, we first discuss existing approaches to interconnect devices in the operating room or intensive care. Afterwards, we introduce a protocol to transmit Electronic Health Records (EHRs) between hospital information systems and medical devices. Our evaluation shows that EHR distribution can be done almost automatically, while robustness is guaranteed against devices which will join a device ensemble both early and late, or against devices which will crash during intervention.**

***Keywords*-Web Services; DPWS; network protocols; e-health records;**

I. INTRODUCTION

Nowadays Operating Rooms (ORs) are equipped with numerous electronic appliances like navigation systems, operating microscopes, anesthesia machines, ventilators, and much more. While the amount of medical devices continuously increased over time, device interconnectivity has not been adapted in the same way [1]. The provision and consumption of medical information along the treatment path is either not existent or enabled by proprietary protocols, which are often very limited in terms of interoperability.

Generally, IEEE defines interoperability as the ability of two or more IT systems to exchange information and to utilize the information that has been exchanged [2]. By establishing interoperability in medical scenarios, right information would be provided at the right time, in the right amount, at the right location, and in the necessary quality [3]. As a consequence, caregivers would be relieved from workload and could benefit from decision support systems, resulting in higher patient safety. Some product solutions like OR.1 from Storz [4] or EndoALPHA from Olympus [5] already offer integrated medical device ensembles. Unfortunately, these complete systems prevent hospital operators from buying best-of-breed products. Beyond that, they do not cover plug-and-play-like device exchanges, which is an important feature for future integrated ORs.

So far, different international research and standardization activities pursue the establishment of medical device interoperability (see section II). This led to a lot of fundamental concepts. Some of the results were never used in practice, others miss concrete protocols for enhancing device collaboration or data exchange with Hospital Information Systems (HISs).

Recent work has revealed that a promising approach to interconnect medical devices are Service-oriented Architectures (SOAs), originally used in enterprise environments. SOA is a design paradigm for distributed IT systems. Functionality is provided in form of services. These services are connected by an abstract messaging backbone and offer the ability to be mutually consumed. By using a service directory mechanism, services are loosely coupled and can be dynamically exchanged at runtime. A more comprehensive introduction to SOA is given in [6]. By adapting a Service-oriented Device Architecture (SODA) [7], SOA principles were made applicable to (medical) devices.

Since SOA is just a conceptual model, technological specifications are required to implement a service-oriented environment. For clinical use, it has become apparent that the most common specification is the Web Service Technology. Invented in the early 2000, Web Services are based on the well-known Extensible Markup Language (XML) and the communication protocol SOAP. They further encompass a large set of extensions, called WS-*, meeting Web Services addressing, transactions, reliability, eventing, and further features. For more information please refer to [8].

Meanwhile, foundational interconnectivity concepts and technologies can be considered as sufficiently mature. As the next stage, investigations on elaborated protocols should be done to get medical devices start working together in a dynamic manner, generating added value. A frequently discussed use case in terms of medical device interconnectivity is the acquisition of Electronic Health Records (EHRs) to display patient demographics at the device side and re-use data for documentation affairs. An EHR is usually known as a systematic collection of digitalized patient information that should be sharable between different health care settings. We use EHR, patient data, and patient demographics synonymously.

In this paper, an automatic EHR dissemination process will be described to avoid the necessity to manually typing in patient data at every single device involved in a surgery. Section II gives a survey of prior work related to medical device interoperability and patient record acquisition. Section III describes prerequisites and assumptions on our approach. In section IV protocol and implementation details will be explained. The protocol is evaluated in section V. Section VI concludes our work, enriched with some impressions on future perspectives.

II. RELATED WORK

Many research projects and standardization efforts were carried out on medical device interoperability. Interesting work comes from Ibach et al. [9], Mauro et al. [7] and Pöhlsen et al. [10]. Ibach et al. introduce a SOA-based connectivity model including a dynamic service discovery mechanism and risk analysis. Mauro et al. modified the SOA paradigm to SODA (mentioned in Section I). Beyond SOA, their model contains a legacy wrapper pattern, a dynamic adapter pattern, and an auto-publishing pattern. Pöhlsen et al. have designed mechanisms and protocols for data security, reliability optimization, and discovery over subnet-boundaries. All approaches have in common that they depict Web Services as a middleware solution. Especially the Devices Profile for Web Services (DPWS) [11] plays a major role, because it comprises decentralized service discovery and publish/subscribe capabilities.

Further important work is done by Goldman et al. as part of the Medical Device “Plug-and-Play” Interoperability Program (MD PnP) [12]. They created the Integrated Clinical Environment (ICE) standard, defining functional elements for Point-of-care (PoC) related IT systems, especially focusing on communication of patient data, and on equipment command and control [13]. Though the ICE standard gives sophisticated information on conceptual system design, no concrete implementation details have been defined.

Besides, some standards deal with medical device interoperability. They are ISO/IEEE 11073, Health Level 7 (HL7) and Digital Imaging and Communications in Medicine (DICOM). While ISO/IEEE 11073 is specifically designed for device communication issues, the latter standards focus on data exchange between different clinical departments.

ISO/IEEE 11073 is a standards family separated into series 11073-1xxxx to 11073-7xxxx, of which the first three are the most important ones. ISO/IEEE 11073-1xxxx defines fundamentals for all subsequent parts, containing language elements, semantics, and an object-oriented Domain Information Model (DIM). The second part describes message exchange patterns between medical devices referring to the upper application layers of the ISO/OSI model. Physical interfaces are described as part of the 3xxxx serie. Today they are based on wired and wireless communication techniques (since infrared has never been accepted [14]).

HL7 is both a name for a not-for-profit, ANSI-accredited organization and a set of standards. It provides frameworks for integration and exchange of electronic health information between different vendors. Currently, two major versions exist: HL7v2 and HL7v3. Based on CSV-like formatted text data, HL7v2 is a pragmatic approach for message exchange, whereas version 3 uses XML and defines a comprehensive semantic Reference Information Model (RIM) of clinical processes.

DICOM is an open standard, preferably founded for the management of image data. It is typically used by radiology imaging systems, and supports encoding of one and two dimensional signal curves, and even video data. On top of that, it is possible to create work-lists and diagnosis-reports containing OR management data. Because the DICOM specification comprises more than 4000 pages, no end system supports the whole standard. Instead, DICOM conformance statements are used to confirm a certain set of functions.

In this context, another prominent initiative is Integrating the Healthcare Environment (IHE). Rather than specifying new standards, IHE is a group of health-care professionals and industry members harmonizing given standards and defining clinical processes on top of them.

All previous mentioned standards define conceptual workflows or data formats in terms of EHR dissemination. Unfortunately, most of them consider HISs, only. Otherwise, there are no implementation directives defined to technically accomplish data distribution in a plug-and-play like fashion. In this paper we propose to utilize concepts of the IHE Patient Demographics Query (PDQ) specification in connection with DPWS technology to give a concrete process flow for EHR distribution.

III. PREREQUISITES & ASSUMPTIONS

Starting to think of data distribution between (medical) devices seems to be a simple task, but gets very complex the more details come into play. Therefore, this section makes some prerequisites and assumptions on work that would exceed the scope of this paper.

A. Devices Profile for Web Services

DPWS is defined as a set of Web Service standards tailored to be run on constrained devices. Some useful enhancements like device discovery over subnet-boundaries [10] and dual-channel transmission [15] complements this profile. Our feasibility study [16] has shown that DPWS is suitable for the clinical environment. Hence, it forms the communication framework for our concerns.

B. Authentication

Security plays a major role in integrated clinical environments, where data exchange is made among devices of different vendors and HISs. Enterprise security concepts are widely adopted within the hospital management, but not in the scope of medical device interaction. Beyond confidentiality and availability, dealing with EHRs requires two important security aspects: data integrity and accountability. It helps preserving patient safety and guarantees information usability in court cases. Integrity and accountability can be established by using authentication and non-repudiation mechanisms, enabled through digital signatures and data logging. WS-Security [17]

```
<patient classCode="PAT">
 <id root="1.2.840.114350.1.13.99998.8734" extension="34827R534"/>
 <statusCode code="active"/>
 <patientPerson>
  <name>
   <given>Jim</given>
   <family>Jones</family>
  </name>
  <telecom value="tel:+1-795-555-4745" use="HP"/>
  <administrativeGenderCode code="M"/>
  <birthTime value="19630713"/>
  <addr>
   <streetAddressLine>8734 Blue Ocean Street</streetAddressLine>
   <city>Other City</city>
   <state>IL</state>
  </addr>
  <!-- ... -->
  </patientPerson>
 <!-- ... -->
</patient>
```

Listing 1. Sample HL7v3 EHR record [19] which is also part of the PDQ specification.

provides directives to handle digital signatures. Pöhlsen et al. [18] made a concept for distributed access control of medical devices including integrity and accountability. We assume that every exchanged message can be tested on these security parameters.

C. Context acquisition

Before EHRs can be safely transmitted, it is indispensable to ensure a common device communication context. This context helps grouping devices together such that they know each other and the subject they will be applied to. Unfortunately, it is not sufficient to simply argue that, for example, a sub-network provides a common device context. This consideration is obsolete because of wireless technology and even sub-networks which are spanned over more than one OR. Thus, it is mandatory to create infrastructure-independent device ensembles.

Context acquisition is done by designating a shared unique identifier either manually or automatically for a group of objects. Regarding to IEC 80001 [20], applying the context manually could be established by IT network risk managers the first time a device is placed in an OR. This works well for non-mobile units, but is not applicable to mobile devices. Gaining a context automatically by means of computer-supported localization techniques is a complete additional research area. Presently, we know no adequate way of automatic localization. Hence, for our protocol we assume that every device has already acquired contextual information.

D. Semantics

One condition to produce inter-operable IT systems is standardization of data formats and semantics. Only if every parameter is strictly regulated, devices of different types and vendors can work together. By using DPWS, data is XML-serialized by default, and can be structured and described with XML Schema. Meaningful data could be generated by applying semantic identifiers like they are defined in the HL7v3 RIM. The IHE initiative proposes to use a subset of HL7 terminology. An example data set is illustrated in Listing 1. As we can figure out, foundational elements are already included like name, gender and birth date. Beyond that, the inclusion of weight, size, medication information, and even Web Access to DICOM Persistent Objects (WADO) [21] could also be useful. The process of disseminating data is independent of any message payload, so we consider patient data to be given by a third party. Since XML offers dynamic language extensions, data is addable on demand.

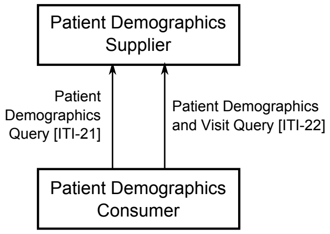

Fig. 1. IHE PDQ architecture as described in [22].

IV. EHR DISSEMINATION PROTOCOL

The acquisition of EHRs are twofold. First, what type of data should be provided, and second, which steps are necessary to enable data distribution. As mentioned above, the first aspect is out of scope. Regarding the second point, the IHE initiative has created an IT Infrastructure Technical Framework (ITI TF) [22], including architectures and transactions to obtain patient data. Fig. 1 shows the IHE proposed architecture to resolve EHRs. Patient data consumers send *Patient Demographics Queries* or *Patient Demographics and Visit Queries* to a patient supplier. The supplier in turn receives these requests, obtains data by means of proprietary interactions with third party systems, and returns them to the requesting consumer. This system works great if there is only one device and the caregiver has to confirm patient data once. As soon as two or more devices need data, a significant amount of additional work is generated: the caregiver has to confirm patient data at every single device. Furthermore, if a device crashes or is rebooted anyway, data has to be requested and confirmed again. It is likely that such systems will not be accepted by clinical staff.

A. Coarse-grained Procedure

Fig. 2 depicts the physical device infrastructure of an OR. Every connected device and IT system know their neighbors by means of WS-Discovery (WS-DD) [23], that is part of DPWS. Devices are grouped by using the context identifier as a scope parameter of a WS-DD Probe request. Therefore, context identifiers should be representable as a URI.

Typically, an OR Management System (ORM) builds a bridge to the HIS, using protocols like DICOM and HL7. In the following, an IT system that fetches EHRs, will be referred as a gateway unit. Since most ORMs speak protocols that will not suit to DPWS (even Web Services based on Basic Profile 1.1 differ in the SOAP version that DPWS prescribes),

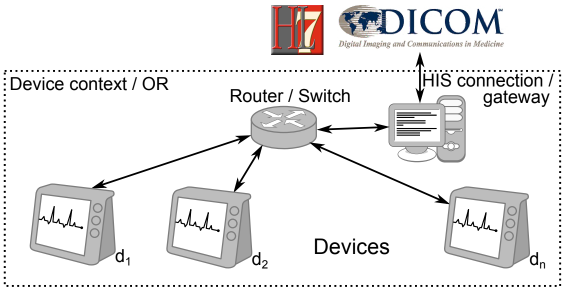

Fig. 2. Physical infrastructure of an OR. The HIS might be connected through a switch/router or—like it is illustrated here—connected through an OR Management System.

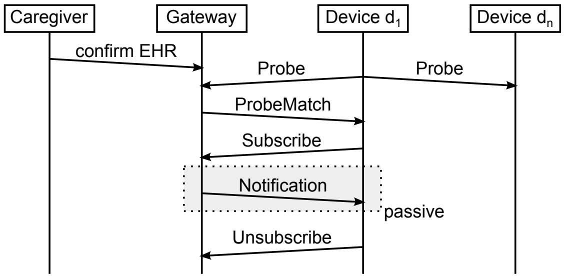

Fig. 3. Sequence diagram of the basic process flow. A notification is send once per intervention. To keep things clear, the process is illustrated for d_1 only.

a certain converter unit is required to translate messages. Since our system is SOA-based, theoretically any device could play the converter role. However, in most cases the converter will be deployed on the gateway unit or on a separated computer.

The process flow comprises four stages:

1) EHRs (please note the plural) of different patients are requested and will be stored on the gateway unit. Since DPWS is SOA-based, it is possible to deploy the gateway on the ORM or on any medical device.
2) As soon as EHRs are available, they are transferred to the converter unit. Usually, ORMs do not actively publish their data, so that the converter unit has to periodically pull them.
3) A certain patient record must be confirmed. This step is required due to regulatory affairs. It implicates a human actor and a human-machine-interface either on the gateway unit or on any other medical device. The caregiver authenticates to the system and confirms a record out of a list. This list could be filtered, e.g., by date, by using bar codes or near field communication, or by means of a mobile unit that provides patient information. Finally, in every case the caregiver has to confirm an EHR due to legal reasons.
4) Data is distributed by using a passive and active request sequence. This stage is described in detail in subsection IV-B.

B. Dissemination Protocol

Electronic, computer-supported systems or devices always suffer from failures and will seldom be used as intended. Because of that, it is very important to design failure resistant systems. For our concerns, to compensate connection losses and reduce workload overhead, we introduce a passive and an active EHR retrieval using WS-DD and WS-Eventing (WS-E) [24]. Passive means that data is distributed via publish/subscribe. Any medical device, which is interested in patient data, subscribes to a gateway unit and passively waits for incoming messages. Active means that any medical device is getting patient data by directly sending a request to the gateway unit.

A (desirable) process flow is given as depicted in Fig. 3. First, every device d_i sends a WS-DD Probe request with a proper context id. To achieve a more precise result set, an EHR supplier type could be defined. Due to the fact that WS-DD scopes and types can be freely selected, they have to be standardized to ensure interoperability. After a suitable gateway g was found, devices d_i have to send a WS-E Subscribe message. The delivery mode [25] and filter dialect [26] are prescribed by DPWS. Similarly to scopes and types, filter URIs have to be standardized. The parameter `Expires` must be set to any realistic value, and the underlying subscription should be renewed accordingly. Any subscription runs until the corresponding software is logged out from the system or shut down. As soon as an EHR is available, it will be published via WS-E Notification messages.

The aforementioned process flow discusses an optimal behavior, but sometimes devices will not be available, when the gateway is ready to send EHR notifications. Beyond, it is even desirable to confirm patient data at the device side, not exclusively at the gateway. To overcome these issues, another step is required: the active mode. As soon as a device has subscribed to the gateway unit, it asks for any existing EHR. The gateway responds either with a single confirmed record or, if no patient data was confirmed yet, with a list of records. In the latter case patient records can be confirmed at the device side and then be sent back to the gateway. Since the gateway computer is the central instance to manage EHRs, the device on which the patient was confirmed, has to wait for the WS-E Notification (see Fig. 5) instead of using the recently confirmed record. Any WS-Addressing action identifiers for these requests, or any mechanisms to confirm patient records, are out of scope and have to be defined by a standardization committee.

The active and passive process flow is shown in Fig. 4. Due to communication delays, it could be possible that a device actively requests patient data and receives an additional notification. Usually, EHRs contain an unambiguous identifier representing the complete patient's hospital stay. If this identifier is missing, an alternative approach is necessary to guarantee EHR uniqueness. Therefore, EHRs should be signed with a timestamp/clock/node based UUID [27], designated as u_{origin}. If a device resolves patient information the first time, it persists this UUID as u_{last}. When another EHR is received

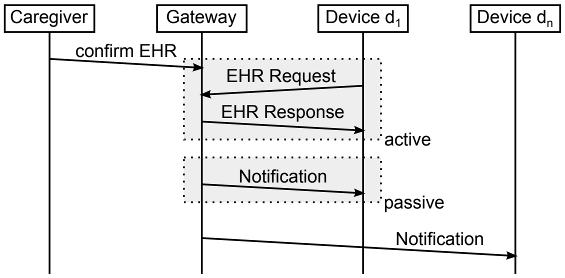

Fig. 4. Sequence diagram of the enhanced process flow. Here the patient record is confirmed in advance. Devices d_i fetch the data actively and passively. It should be noticed that EHRs might be confirmed at any point in time. To keep things clear, the process is illustrated for d_1 only. Any probes and subscriptions were already performed.

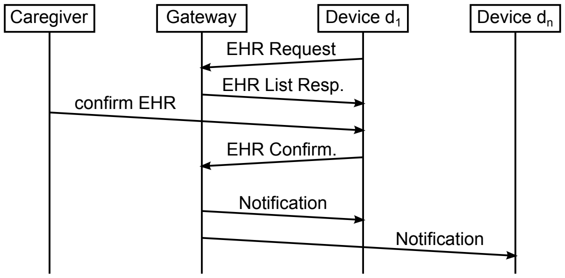

Fig. 5. This diagram shows the process of confirming an EHR at the device side. First, a requesting device receives a list instead of a single confirmed record. Then it visualizes the list and waits for data confirmation. Afterwards, the confirmation acknowledgment is send to the gateway followed by a WS-E Notification—only the notification data is relevant to the device.

or requested, $u_{origin} = u_{last}$ indicates that patient data had not changed, while $u_{origin} \neq u_{last}$ reveals that a new patient was allocated.

On the other side, if a gateway computer crashes, a WS-DD Hello message indicates that devices have to re-subscribe to the gateway.

V. EVALUATION

The goal of our work was to create an automatic and robust patient data dissemination protocol. To reduce workload, data is distributed almost in an automatic manner using WS-DD and WS-E. By sending EHR requests and waiting for WS-E Notification messages, devices are able to automatically resolve current patient data independent of system crashes and early or late group joining. We do not consider the case when a device discovers two or more EHR suppliers during a Probe request. Due to SOA's lose coupling directive, every supplier should provide the same patient information consistent with the current HIS/ORM data. A device could benefit from several data sources by performing plausibility checks. Since plausibility checks are not part of our communication protocol, they were not considered any further.

A. Experimental setup

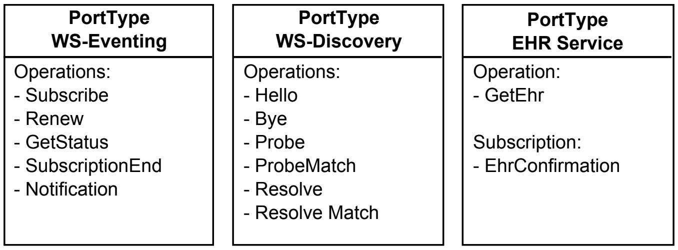

Fig. 6. Interface definition required to disseminate patient data. Most functionality is derived from well-known standards, which are also part of DPWS. WS-E operations are defined in [24] whereas WS-DD operations are defined in [23]. Any EHR supplier has to implement at least a *GetEhr* operation for active mode and an *EhrConfirmation* event source for passive mode.

To evaluate the performance and feasibility of our system, a prototypical implementation has been set up on a Microsoft Windows 7 64-bit based Java Virtual Machine using the Web Services for Devices Java Multi Edition Stack (WS4D-JMEDS), version 2.0beta8 [28]. Fig. 6 illustrates the WSDL-based service interface a gateway has to realize. Two kinds of measurements are taken. First, the amount of messages exchanged to disseminate EHRs. Second, the elapsed time until every communication participant has received patient records when using 1, 2, 4, 8, 16, 32, and 64 consumers. Since our approach is a novel consideration to automatically enable patient data on multiple medical devices, it is difficult to compare with other systems. A general performance gain is fundamentally given through the fact that just a single EHR confirmation is required instead of N confirmations, where N is the amount of devices interested in any patient data. The underlying setup uses no authentication yet.

The experimental setup comprises of a single PC, equipped with a 2.4 GHz quad core CPU and 16 GB of RAM. All devices, including the gateway, run on one machine within different Java processes. In the following, they are called virtual devices. Payload data is taken from the IHE PDQ [19]. It is about 10 KB of data. To take time measurements, a monitoring application is connected to the gateway and consuming devices d_1 to d_n. This application is responsible to collect event triggers by using a primitive Java Remote Method Invocation (RMI) application. Since virtual devices run on a single PC, no network traffic is generated. Hence, in real-world scenarios additional transmitting time has to be expected. Furthermore, Java cross-optimizes shared objects, which will lead to additional performance gains.

To illustrate feasibility of the approach, following test cases are considered:

1) The gateway is started and the EHR is confirmed. Hereafter, devices d_1 to d_n are turned on.
2) The gateway is started. When booted, devices d_1 to d_n are turned on. Then, the EHR is confirmed.
3) The gateway is not available. Devices d_1 to $d_{\frac{n}{2}}$ are turned on. When done, the gateway is started and the EHR is confirmed. Afterwards, devices $d_{\frac{n}{2}+1}$ to d_n are turned on.
4) When patient data is distributed, a single device is synthetically shut down and turned on again to simulate

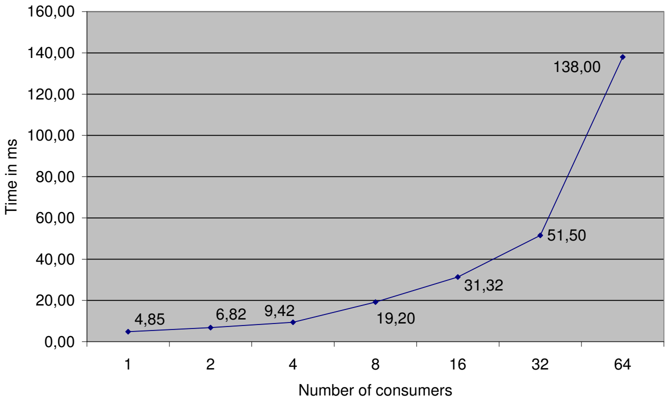

Fig. 7. Different measurements of time span between confirming an EHR and disseminating it to interested sinks.

a crashed PC. This is done by sending and even omitting a WS-DD Bye message.

5) When patient data is distributed, a gateway crash is simulated. This is also done by sending and omitting a Bye message.

B. Results

Regarding the foregoing test cases, point one and two work as expected. In the first case, data is retrieved using the active mode (operation *GetEhr*). In the second case, data is retrieved using the passive mode (event source *EhrConfirmation*). Test case three forces d_1 to $d_{\frac{n}{2}}$ to wait in passive mode, while $d_{\frac{n}{2}+1}$ to d_n can obtain their data using the active mode. The fourth scenario is twofold. If the single device d_i is shut down sending a Bye message, it can properly be removed from the gateway's subscription list. In the other case, the gateway does not know if d_i is out of order. To avoid maintaining obsolete subscriptions, client devices are encouraged to send heartbeats in form of WS-E Renew messages. After d_i is restarted, data is retrieved using the active mode. To detect a patient exchange, d_i can compare the unique EHR identifier described in section IV-B. In the last case an EHR supplier crash is simulated. If a Bye message is sent, every device clears their subscription and renew them as soon as the gateway enters the network again. If no Bye message is sent, client devices will not notice any changes. The gateway persists subscriptions and send WS-E SubscriptionEnd messages after restart. Client devices receive these messages and quit the outdated subscription to subsequently create a new one.

Fig. 7 shows the efficiency measurements in terms of the time span between confirming an EHR record and disseminating it to interested parties. Hundred measurements were taken for 1, 2, 4, 8, 16, 32, and 64 consumers. Having a rough overview of Fig. 7, doubling the amount of consumers causes a doubled time consumption. However, due to the fact that four cores process the gateway and consumers, time is not exactly increasing proportionally with the amount of consumers. In the end, there is no unreasonable time delay when distributing data up to 64 devices. Unfortunately, the monitoring instance had noticed data loss in some measurement runs. The reason for that behavior is not conclusively clarified. May be there is a software bug occurred in JMEDS, or there is an operating system issue when processing loads of TCP connections pointing to the localhost.

TABLE I
TRANSMITTED MESSAGES PER NUMBER OF CONSUMERS.

Number of consumers	Transmitted messages
1	13
2	26
4	52
8	104
16	208
32	416
64	832

The amount of transmitted messages required to disseminate EHRs is based on a theoretical approximation. Hereby, sending a SOAP message counts as sending one message. Sending a multicast message conforms to sending K messages, where K is the number of multicast channel subscriptions. The minimum amount of messages M is given by the following formula:

$$M = 13x + xp + \lfloor \frac{r}{t} \rfloor x$$

The variable $x \in \mathbb{N}$ is the number of consumers, $r \in \mathbb{R}_+$ is the time until a WS-E Renew message has to be sent, $t \in \mathbb{R}_+$ corresponds to the time until the gateway is shut down, and $p \in \mathbb{N}_0$ is the amount of patient exchanges occured during system runtime. This formula does not cover any failures and any devices joining the network late in time. It also does not cover the existence of further devices providing any services, which would increase the amount of WS-DD Hello and Bye messages. Therefore, the formula defines a lower bound for the amount of transmitted messages. The term $13x$ describes the amount of messages initially transmitted, comprising of discovering devices, subscribing to the gateway, quitting subscriptions and network participations, and retrieving patient data the first time. The term xp covers notifications sent by the gateway due to a patient exchange. The last term represents the number of messages transmitted to renew subscriptions. In our test scenario, it is: $t < r$ and $p = 0$. Hence, the amount of messages is simply based on $M = 13x$. TABLE I illustrates different message counts. In comparison to simple get requests like they are done using PDQs, the amount of messages is very large. But it is reasonable regarding to the benefit of reduced workload and a discovery process to dynamically plug in any devices.

VI. CONCLUSION & FUTURE WORK

In this paper, we have created an EHR dissemination protocol suited to dynamic, interconnected ORs meeting robustness against system crashes and early or late group joining. By using a passive and active mode, every device is synchronized with the latest EHR. In comparison to existing systems no additional, human-involved configuration or confirmation is required, which reduces or at least does not increase the caregiver's workload. Due to the best-effort behavior of Ethernet, this protocol does not address transmission failures. Hence, it is not robust against physical interferences.

Further research has to be done both on mechanisms to handle documentation affairs on the basis of transmitted EHRs, and on localization techniques to (semi-)automatically compose device ensembles. Beyond that, optimizations regarding to WS-E filter techniques could be useful to reduce data load. Finally, the evaluation could be extended to real-world device setups with authentication enabled.

REFERENCES

[1] K. Lesh, S. Weininger, J. M. Goldman, B. Wilson, and G. Himes, "Medical Device Interoperability-Assessing the Environment," in *Proceedings of the 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSSMDPnP)*. Cambridge, Massachusetts, USA: IEEE Computer Society, June 2007, pp. 3–12.

[2] "IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries," *IEEE Computer Society Press*, p. 1, January 1991.

[3] A. Schweiger, A. Sunyaev, J. M. Leimeister, and H. Krcmar, "Toward Seamless Healthcare with Software Agents," *Communications of the Association for Information Systems (CAIS*, pp. 692–709, 2007.

[4] Karl Storz, "Kartl Storz OR1," 2012/10/19, Version: 2013-05-13. [Online]. Available: http://www.karlstorz.com/cps/rde/xchg/SID-CF744F30-788546A4/karlstorz/hs.xsl/522.htm

[5] Olympus, "Olympus - EndoALPHA," Version: 2013-05-13. [Online]. Available: http://www.olympus-europa.com/endoalpha

[6] E. Thomas, *SOA Principles of Service Design*, 1st ed. Prentice Hall International, Jul. 2007.

[7] C. Mauro, A. Sunyaev, J. Leimeister, and H. Krcmar, "Standardized Device Services - A Design Pattern for Service Oriented Integration of Medical Devices," in *2010 43rd Hawaii International Conference on System Sciences (HICSS)*, Jan. 2010, pp. 1–10.

[8] C. Werner and S. Fischer, *Semantic Web Services: Concepts, Technologies, and Applications*. Springer, 2007, ch. 2, p. 25ff.

[9] B. Ibach, J. Benzko, and K. Radermacher, "OR-Integration based on SOA – Automatic detection of new Service Providers using DPWS," *International Journal of Computer Assisted Radiology and Surgery*, vol. 1, pp. 195–196, 2010.

[10] S. Pöhlsen, S. Schlichting, M. Strähle, F. Franz, and C. Werner, "A Concept for a Medical Device Plug-and-Play Architecture based on Web Services," *SIGBED Rev.*, vol. 6, no. 2, pp. 1–7, Jul. 2009, Version: 2013-05-13. [Online]. Available: http://doi.acm.org/10.1145/1859823.1859829

[11] E. Zeeb, G. Moritz, D. Timmermann, and F. Golatowski, "WS4D: toolkits for networked embedded systems based on the devices profile for web services," in *2010 39th International Conference on Parallel Processing Workshops (ICPPW)*, Sep. 2010, pp. 1–8.

[12] "MD PnP Program," Version: 2013-05-13. [Online]. Available: http://www.mdpnp.org

[13] J. M. Goldman, "Medical Devices and Medical Systems — Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) — Part 1: General requirements and conceptual model," ASTM International, 2008, Version: 2013-05-13. [Online]. Available: http://www.mdpnp.org/uploads/F2761_completed_committee_draft.pdf

[14] M. Galarraga, L. Serrano, I. Martinez, and P. de Toledo, "Review of the ISO/IEEE 11073 - PoCMDC standard for medical device interoperability and its applicability in home and ambulatory telemonitoring scenarios," Electrical and Electronic Engineering Dept. Public University of Navarra (UPNA). Pamplona. Aragon Institute of Engineering Research (I3A). University of Zaragoza (UZ). Zaragoza. Biomedical Engineering and Telemedicine Group (GBT). Polytechnic University of Madrid (UPM). Madrid., Spain, Tech. Rep., 2006.

[15] S. Pöhlsen, W. Schöch, and S. Schlichting, "A Protocol for Dual Channel Transmission in Service-Oriented Medical Device Architectures based on Web Services," in *3rd Joint Workshop on High Confidence Medical Devices, Software, and Systems & Medical Device Plug-and-Play Interoperability*, 2011.

[16] D. Gregorczyk, T. Bußhaus, and S. Fischer, "A Proof of Concept for Medical Device Integration using Web Services." IEEE, Mar. 2012, pp. 1–6, Version: 2013-05-13. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6198124

[17] A. Nadalin, C. Kaler, and R. M. P. Hallam-Baker, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)," OASIS Open, Tech. Rep., February 2006.

[18] S. Pöhlsen, S. Schlichting, and C. Werner, "Praktische Umsetzung einer verteilten Zugriffskontrolle für Webservices anhand ihrer Eigenschaften," *PIK – Praxis der Informationsverarbeitung und Kommunikation*, vol. 33, no. 2, pp. 140–146, Jun. 2010, Version: 2013-05-13. [Online]. Available: http://dx.doi.org/10.1515/piko.2010.021

[19] Jan. 2013, Version: 2013-05-13. [Online]. Available: ftp://ftp.ihe.net/TF_Implementation_Material/ITI/examples/PDQV3/02_PDQQuery1Response.xml

[20] S. Eagles, "An Introduction to IEC 80001: Aiming for Patient Safety in the Networked Healthcare Environment," *IT Horizons*, vol. 4, pp. 15–19, 2008.

[21] *DICOM Standards Commitee, 2010b. Supplement 148: Web Access to DICOM persistent Objects by means of Web Services: Extension of the Retrieve Service (WADO Web Service)*, National Electrical Manufacturers Association (NEMA) Std., 2010, Version: 2013-05-13. [Online]. Available: ftp://medical.nema.org/medical/dicom/Supps/sup148_17.pdf

[22] "IHE IT Infrastructure (ITI) Technical Framework – Volume 1 (ITI TF-1) Integration Profiles," IHE International, Inc., Tech. Rep., 2012.

[23] V. Modi and D. Kemp, "Web Services Dynamic Discovery (WS-Discovery) Version 1.1," OASIS Open, Tech. Rep., Jul. 2009.

[24] D. Davis, A. Malhotra, K. Warr, and W. Chou, "Web Services Eventing (WS-Eventing)," W3C, Tech. Rep., 2011.

[25] "WS-Eventing Delivery Mode," http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push.

[26] "WS-Eventing Filter Dialect," http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01/Action.

[27] P. Leach, M. Mealling, and R. Salz, "RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace," IETF, Tech. Rep., Jul. 2005, Version: 2013-05-13. [Online]. Available: http://www.ietf.org/rfc/rfc4122.txt

[28] "JMEDS (Java Multi Edition DPWS Stack)," Version: 2013-05-13. [Online]. Available: http://sourceforge.net/projects/ws4d-javame/files/ws4d-jmeds-v2/framework/v2.0.0-beta8/