ELSEVIER
MASSON

CrossMark

Disponible en ligne sur
ScienceDirect
www.sciencedirect.com

Elsevier Masson France
EM|consulte
www.em-consulte.com

IRBM

IRBM 37 (2016) 198–209

A Reconfigurable Middleware for On-demand Integration of Medical Devices

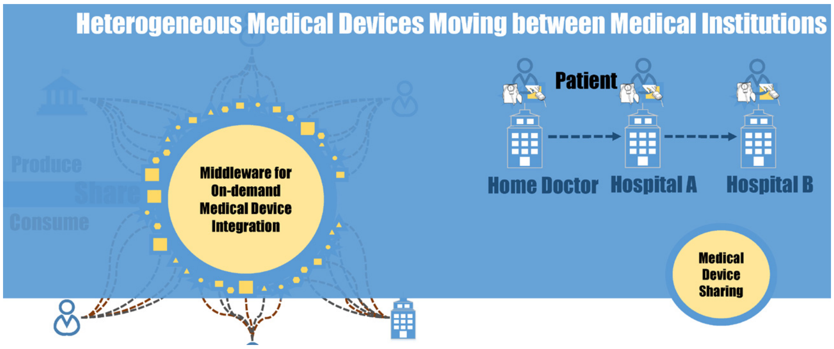
A. Kliem [*], A. Boelke, A. Grohnert, N. Traeder

Technische Universität, Complex and Distributed IT Systems, Einsteinufer 17, 10587 Berlin, Germany

Received 4 May 2015; received in revised form 9 May 2016; accepted 10 May 2016

Available online 9 July 2016

Graphical abstract


Heterogeneous Medical Devices Moving between Medical Institutions
Patient
Home Doctor
Hospital A
Hospital B
Produce
Share
Consume
Middleware for On-demand Medical Device Integration
Medical Device Sharing

Abstract

Objective: Systems designed for medical device integration often suffer from a limited set of supported devices. Justified by resource and mobility constraints, integration systems like smart phones or gateways are restricted with regard to the set of device control and data aggregation logic deployable. This contradicts the requirements of patients and physicians in terms of vendor-independence and adaptability of the system. Often a large variety of medical devices needs to be considered to provide a meaningful survey of a patient's condition. Moreover, especially in case of emergencies, the set of required devices is often changing rapidly.

Method: Physical integration and spontaneous interoperability are core aspects of open and modular device integration solutions. Therefore, we first introduce the OSGi Device Access Specification as a technical foundation to our approach. Furthermore, we discuss an external representation of device descriptions and the impact of semantic interoperability with regard to the e-Health domain. Based on these findings, the architecture is introduced and described with respect to its components and behavior.

Results: As a mitigation to the problem of medical device integration, we present a middleware platform that is able to adapt itself to the requirements of patients and Care Delivery Operators. Based on a modular design, medical devices can be integrated on-demand. The middleware is composed of a device integration and a data aggregation layer. The aggregation layer allows transforming heterogeneous data streams according to the demands of the respective clinical information system. Both layers utilize external knowledge repositories to deploy currently required device control and data aggregation logic at runtime. A proof of concept is given by a prototype developed for a tele-rehabilitation project.

[*] Corresponding author.

E-mail addresses: andreas.kliem@tu-berlin.de (A. Kliem), anett.boelke@tu-berlin.de (A. Boelke), anne.grohnert@tu-berlin.de (A. Grohnert), nicolas.traeder@tu-berlin.de (N. Traeder).

http://dx.doi.org/10.1016/j.irbm.2016.05.003

Conclusion: Based on a proof of concept prototype developed for a tele-rehabilitation project, we show that a modular and open medical device integration middleware can be developed on top of existing industry standards like OSGi. The presented approach allows Care delivery Operators to integrate semantically aligned medical data streams into their information systems without requiring to reconfigure their integration systems.

Keywords: E-health; Telemedicine; Self-adaptive; Dynamic reconfiguration; Medical device integration; Interoperable medical devices

1. Introduction

The evolution of mobile embedded devices in recent years encourages their widespread adoption in the healthcare domain [1]. Especially domains like intensive care or telemedicine are proposed to benefit from ICT based sensors and communication networks [2–4]. A typical application domain is the monitoring of vital signs with (mobile) sensors connected through Body Area Network (BAN) or Personal Area Network (PAN) [5]. Basically, the aim of such systems is to record and analyze the streams of medical data emitted by the sensors in order to support physicians in their decision making process. However, proper decision making is based on a meaningful survey of a patient's condition. This requires to consider a large variety of typically heterogeneous medical sensors and devices [6]. Furthermore, treatment decisions often have to be made under time constraints. This constitutes the need for an aggregated view of the available streams that should be generated close to the data sources. However, each stream utilized can differ regarding its specific characteristics. This can include real time requirements, used data formats and nomenclatures or the communication protocol implemented by the medical devices [7].

The resulting device integration and data aggregation problems are highly heterogeneous. Since customized and application specific software components are required, proprietary solutions are preferentially implemented. A lot of vendors already provide e-Health solutions especially in the area of vital signs monitoring and telemedicine. However, these systems are often based on closed boxes. Only a limited set of medical devices and sensors, usually from the same vendor or based on the same (proprietary) protocol specification, are integrated. Using proprietary solutions, medical device vendors or system integrators gain market exclusivity, which often forces Care Delivery Operators (CDOs) to be dependent on a vendor (i.e. vendor lock-in). Proprietary solutions hinder the development of open and fully integrated e-Health systems, which are required to efficiently deliver cost-effective health services. The problem is intensified if the characteristics of nowadays treatment processes are considered. These usually involve a lot of different CDOs (e.g. family doctor, specialist, hospital). Since each CDO might rely on different solutions, interoperability cannot be achieved. Moreover, on-demand access to the medical devices of a patient is difficult to realize. Due to the aforementioned variety, interoperability in the e-Health domain requires that medical devices can be integrated at any location on-demand, regardless of the protocols or data formats (proprietary or standard-based) they are based on.

It is often depicted, that the device integration problem can be mitigated by standardization efforts. Appropriate standards like ISO/IEEE 11073 (x73) [8] or the Bluetooth Health Device Profile (HDP) [9] exist. However, a widespread standardization in a reasonable time span is unlikely. Moreover, a lot of standards allow for vendor defined extensions and in particular vendors of complex sensors and medical instruments often rely on proprietary protocols to protect their innovations. As a mitigation to this problem, a medical device integration middleware should be designed such that:

- it is able to handle both standard and proprietary devices
- it is adaptable to cope with the rapidly changing requirements (decreased time to market, new sensors, updated data formats and patient profiles)
- it allows handling all the existing medical devices and preserves interoperability at application level, if devices are replaced (which extends the integration with a migration challenge)
- it can be deployed on resource constrained devices like smart phones or gateways, which are often used as integration systems [10]

Given these constraints, it seems impossible to statically deploy all software components required for device integration and data integration to one middleware platform. Instead, the middleware needs to be able to adapt itself to the requirements of the environment. It has to recognize the environment by detecting currently available devices on-demand and adapting itself in order to handle the devices properly. Knowledge required to cope with so far unknown devices has to be gathered dynamically from external repositories and injected during runtime. This keeps the amount of required resources low, since only software components currently in use have to be maintained. Therefore, the approach we like to present relies on a modular, OSGi based [11], device integration (machine to machine – M2M) and data aggregation middleware platform.

The rest of the paper is structured as follows. Section 2 gives an overview of fundamentals to the solution, such as the layout of the external knowledge repositories or the relation of semantic interoperability to the framework. Section 3 explains the architecture of the middleware and its relevant components. Finally Section 4 highlights some related work and Section 5 concludes the paper.

2. Background

This section introduces the OSGi Device Access Specification (DAS), which acts as the foundation for the device integration capabilities of the middleware. Furthermore, the *Device Directory (DD)*, which implements the knowledge repository