

Generic Sensor Framework enabling Personalized Healthcare

Sven Beckmann*, Stefanie Lahmer*, Moritz Markgraf*, Oliver Meindl*, Julia Rauscher**, Christian Regal*, Henner Gimpel* and Bernhard Bauer**

Abstract— Data and sensor fusion can enable clinical healthcare systems to improve conditions of a patient. However, hospitals are not the only application field of connected medical devices. Domestic monitoring gets more important day by day and applies Internet of Things with mobile sensors, like wearables. Through data processing data is transferred to smart data and personalized recommendations are improvable, if sensors can be chosen individually. Therefore, we developed a generic medical sensor framework which is able to merge any needed sensor and collect data to improve personalized health of an individual. To evaluate our framework and to prove the added value of sensor fusion we present a sensor-based stress detection game.

KEYWORDS: Sensors Fusion, Wearables, Data Processing, Framework, Personalized

I. INTRODUCTION

Monitoring is unstoppable and omnipresent. Changes are recognizable in many circumstances, even in the healthcare area. Sensors are spread anywhere which collect a tremendous amount of data. It is taking place unconsciously in public (e.g. CCTV) or consciously, like self-monitoring with wearables. This trend can involve improved healthcare and offers the possibility to be more informed and aware of its own health condition. By the usage of wearables or e-textiles further personalized monitoring can be conducted. All this collected data can accomplish an important knowledge for health improvement, but the potential is foregoing frequently. Through sensor and data fusion this deficit can be compensated, as combined data exhibit added value [1]. Especially privately collected data are often unused, whereas this data are suitable for smart living or to improve personalized health of an individual by the usage of Internet of Things and its connected data. Here sensor fusion represents the possibility of personalized healthcare, based on the variable choice of used devices. In the medical sector sensor fusion can be deployed in three different application fields. Most common area is clinical sensor fusion in hospitals within Medical Application Platforms [2] for concurrently smart monitoring patients' various health values. In addition, sensor fusion can be used in domestic tele medicine or tele rehab in combination of Internet of Things of Medical Devices (IoTMD). Closely associated is the wellbeing and prevention area, which appears also in combination with IoTMD and Smart Homes. This area represents the most potential and also the most risks of personalized healthcare, since prevention is highest affected by individual's health needs. A personalized medical Smart Home which is using sensor fusion requires different devices, for instance wearables, nearables and

sensors for each patient's medical precondition. However, this fact represents one of the most complex issues of sensor fusions. Devices are heterogenic and difficultly combinable.

We address the above named issue by developing a generic medical sensor framework which is able to combine sensors and collect data independent of devices. Through this solution we make sensor fusion generic, thus enable personalized health needs and the usage of desired sensors. Furthermore, we offer continuous measuring including collecting data and enable further processing. In this way it is applicable on plenty health monitoring use cases. The potential for utilizing a framework for generic medical sensor fusion was recognized by many research approaches. For example, the authors in [3] proposed an infrastructure based on contextual objects in distributed client/server information systems, whereas [4] developed a toolkit for context application programming. Furthermore, the FUNF framework [5] is capable to acquire data over third-party sensors, supporting different transmissions protocol and the Open Service Architecture for Sensors (OSAS) framework, an event-based programming system for sensor networks. In contrast, our solution differs through the fact that we use physiological data of patients and combine these with ordinary sensors and medical sensors, and minimize limitations of personalization through unbounded sensor choice.

As already mentioned, digitalization brings many opportunities. However, these opportunities also involve challenges. The increasing usage of information technology in business and private life negatively impact health, e.g. stress. Individual stress sensitivity represents an example for a personalized wellbeing and prevention use case. We avail our use case "Stress Detection and Prevention" for the concluding evaluation. Therefore, we will discuss three main theses including the functionality of our framework and the successful personalized stress monitoring. In addition, it will be shown that sensor fusion-based data offer more potential for healthcare improvement than single sensor-based data. Along with this we can reveal the health improvement potential of the combination of ordinary sensors and wearables.

The remaining paper is structured as follows: After introducing our approach for a personalized healthcare sensor fusion we present our Java Data Collecting (JDC) Framework. The architecture, components and functions will be shown. Afterwards we evaluate the framework in terms of proper functionality and health improvement by conduction an experiment using a stress game.

*Sven Beckmann, Stefanie Lahmer, Moritz Markgraf, Oliver Meindl, Christian Regal and Henner Gimpel are with the Research Center Finance & Information Management and the Project Group Business and Information Systems Engineering of Fraunhofer FIT, E-mail: name.surname@fim-rc.de

**Julia Rauscher and Bernhard Bauer are with the Department of Computer Science of University of Augsburg, Germany, 86153 Augsburg; E-mail: name.surname@ds-lab.org

II. JDC FRAMEWORK

Today, there exist a great variety of sensors in the healthcare sector. The authors in [6] categorize sensors used in healthcare monitoring systems according to their spatial distance to the patient. 1) Medical body sensors like heart rate monitors, the closest to the individual, 2) environmental sensors e.g. room temperature sensors and 3) location sensors like indoor relative localization receiver more and more move further from the patient. More generally we can differentiate sensors by their transfer medium e. g. wired or wireless, their behavior e. g. active or passive or their type to deliver data e.g. time or event-based. Typically, these sensors represent a specific view and partial model of the patient and his environment. To increase the rate of integrity, more and more data in healthcare systems is collected. On top of this, several sectors exist that enable the usage of sensors. However, sensors which are traditionally not used for the assessment of various information about the user's health (e.g., keyboard, mouse), can provide important aspects of the user's current behavior and hence also with his current physiological or psychological state. This information, acquired by different sensors and devices with diverse initial purposes, is used to build, for example, individualized information systems that adapt themselves according to the user's needs [7]. However, because of its manifold use of application, it requires a highly generic approach to provide this system with the gathered information, independent of the underlying technology and inherent workflows.

The ease of deployment, data acquisition efficiency and application layer challenges as "one of the hardest challenges" in healthcare systems [8]. Especially the huge number of different sensors, results in a big challenge for systems to assess, process and store these data. Having different kind of sensors, the need of a scalable and flexible data persistence increases. In particular, the data persistence (e.g. database, file) can vary or dynamically change depending on the usage. To cope with these challenges generated by this diversity of sensors, we aim to develop a framework that connects multiple sensors and achieve the combination of different ways of data persistence regardless where the sensor is located or how it is connected. Hence, we developed the Java Data Collecting Framework (JDCF) that is capable to assess data and information from different types of sensors and provides this value resources for further systems by storing this information in various data spaces, called persistence forms. JDCF enables an easy way to log all kind of data by the usage of Java objects and change the persistence form easily to address challenges.

In a classic input-process-output point of view JDCF is placed in the middle of two interfaces (*Person & Environment* and *Usage*). In Figure 1 we present JDCF as the connecting part between multiple sensors on the left-hand side (Connection) and different persistence forms on the right-hand side (Persistence). In the first place we describe JDCF on a high abstraction level and go into technical details afterwards.

We developed a five-phase approach from a specific sensor, connected in a personalized healthcare system, to the usage of its collected data. The three middle phases represent JDCF. Every step in this simplified process has its own challenges JDCF helps to overcome through a highly-generic software architecture, called the JDCF data flow:

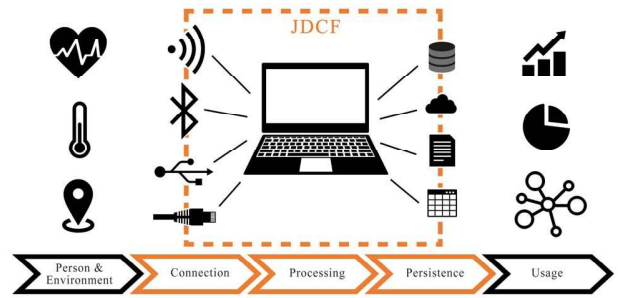


Figure. 1 JDCF data flow

Person & Environment: Tracking the physical or psychological state of a patient e.g. stress, involves a great number of sensors. Since this amount is not static, applications need to be scalable and should allow new, maybe yet unknown, sensors or sensor networks to connect easily. This represents the precondition of JDCF utilization.

Connection: Requiring different kind of sensors, the integration into healthcare systems gets more difficult. JDCF enables to easily connect miscellaneous sensors by providing generic interface place holders to implement existing drivers, libraries or frameworks and integrating these in the data-flow within JDCF to conduct sensor fusions.

Processing: The development of new applications for personal healthcare can be divided in two main challenges: 1) implementing new modules or algorithms to scientifically assess physiological issues and questions as well as 2) building up the software-technical infrastructure to receive the relevant data. To reduce the effort to master these, JDCF delivers a unified but generic data-flow between a sensor and the selected type to store its data or information. Thereby the developer is able to select the type and granularity of the data to be collected as well as the type of the persistence. Furthermore, he has the possibility to simply connect new, not yet implemented, sensors or kinds to store their data. Using existing interfaces and abstract classes allows the integration of new components in the JDCF data-flow.

Persistence: Inconsistent and heterogeneous data pools are time-consuming for further processing (e.g. medical sensor fusion application). Moreover, every preprocessing step to increase the data homogeneity can result in a loss of data quality. Using JDCF helps developers to receive structured data and to store these in dissimilar persistence forms. Beyond the generic persistence structure, we achieved our aim of allowing the connection of new persistence forms. This means it is possible to change the type of persistence on demand. Thereby JDCF can react on a heterogeneous IT-environment just like it is needed in healthcare systems e.g. an application for electronic devices to monitor patients in emergency departments or a personalized health platform to assess stress.

Usage: Especially in personalized health platforms it is mandatory to use existing data for different analyses. Transforming semi-structured data in heterogeneous persistence forms normally implies a costly overhead and a poor system performance. A sensor integrated in the JDCF data flow can have multiple equal forms of data output as well as different persistence forms. Hence, JDCF enables through a generic persistence structure a quantity- and type-independent reuse of the collected data. Using e.g. the heart rate data as a

part of a stress detection model has no negative effects on creating a dashboard for cardio-activities.

A practical demonstration of these 5 phases is offered in section III. In general, this framework supports personal healthcare developers throughout the whole process from sensor-binding to data persistence. It also enables an easy way to log all kind of data and dynamic changes in the persistence form. As well, JDCF can be used by multi-users without violating privacy rules, since it is a self-hosted framework which stored data locally. To further detail the processing of JDCF, we explain the data flow of collected sensor data and go increasingly into detail on the way through JDCF (Figure 2) as far as possible at this point. More details are out of scope.

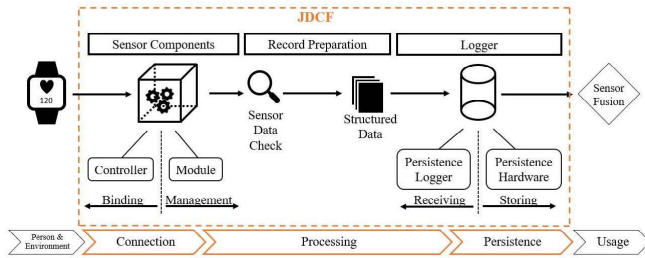


Figure. 2 JDCF Components

Sensor Components: Every sensor in JDCF uses its own SensorController and SensorModule. The main task of the SensorController is the binding between JDCF and the sensor. Therefore, JDCF offers an interface placeholders which can be used by the sensor. The controller enables the usage of external libraries and every other customized implementation to receive data. The SensorModule includes the whole managing part to enable or disable logging or activating/deactivating the sensor connections. This allows very easy handling of the sensors. Furthermore, JDCF offers two different standardized types of controllers and modules. On the one hand event-based sensors, which trigger the controller for new data during events of the hardware sensor and on the other hand time-based sensors that trigger the hardware sensor for new data in a certain time interval. All necessary components like time management for time-based sensors are already included in JDCF. Regardless of the type we use, the controller is the executing point when the sensor is triggered. Every time the controller is executed a new SensorRecord will be created. The SensorRecord is a central object of JDCF which includes the whole received data of one sensor at one receiving moment with a unique identifier and the time of creation.

Record Preparation: Theoretically, a single sensor can give different and unsorted data. To assure the received data is in a sorted structure, every sensor implements its own SensorRecordStructure which determines the data type with enumerations. This structure includes the regulation for data sequence and datatypes for every sensor. The framework includes control mechanisms for every Java object to respect this structure by monitoring every incoming data value. By the usage of generic Java objects data can be stored type-independent. The SensorRecord considers the structure during the creation. In case of not considering the structure JDCF throws a SensorRecordStructure Exception. Thus, a data validation is conducted. The self-checked and correctly filled SensorRecord is send to the Logger.

Every *Logger* is a special type of the JDCF PersistenceLogger. Since there are different forms of persistence, it is important to allow a generic saving of the data. JDCF provides the needed specifications for the logger to be implemented. Due to the information contained in the structure, each logger has all the necessary information to store the data. Thus, every single implemented PersistenceLogger is enabled to log correctly based on the SensorRecordStructure included in every SensorRecord.

To manage several sensors at the same time as easy as possible, we need a solution, holding the SensorModules. Therefore, JDCF contains the SensorManager. It opens the possibility for holding multiple sensors named by a central enumeration. This enumeration is mainly responsible to easy access the sensors by their names and assure there are no duplicates. Furthermore, the user can create multiple sets of sensors by creating multiple enumerations for the SensorManager. This enables the partition of sensors by their use case or type. This implementation of JDCF makes it possible to access any sensor using the SensorManager in combination with the central enumeration. The maximum number of processible sensors in general, is only limited by the available resources of the used hardware for JDCF.

A summary of the described elements of JDCF is provided in table 1. After we presented JDCF, we are going to demonstrate our approach with a stress study. This is followed by practical handling and evaluation of the results.

TABLE 1. Elements of JDCF

Name	Short Description
SensorModule	Contact partner for managing a sensor
SensorController	Connection to the hardware sensor. Executed to create new SensorRecord
SensorManager	Container of SensorModules in combination with an enumeration for easy access
SensorRecord	Sensor data containing the result of a single controller call with an id and timestamp
SensorRecordStructure	Regulation of the sequence and datatypes for sensor data

JDCF provides developers of sensor environments with different structures to facilitate the collection, processing, and storing of acquired data. However, to satisfy the intended purpose and the consideration of legal rules (e.g. data privacy and security) and moral principles (e.g. confidentiality of information) is within the user's range of tasks.

III. EVALUATION

Stress is the second largest health problem in Europe and continuing stress can cause serious diseases. An essential step to cope with stress is to identify it individually. However, the evaluation of the stress level is an enormous task of the wellbeing area. This is based on its high degree of individuality and various other aspects (e.g. stressors, strains, resources, coping). Identifying symptoms like decreasing performance, nervousness or lack of concentration can be measured. Sensor fusion within JDCF enables to measure combined different influences. To evaluate our framework, we conducted an experiment using JDCF to perform a sensor fusion, gathering and preprocessing data. For this experiment, fifteen participants played a stress game [9], which puts participant

under stress by inducing different stimuli. Therefore, the game involves two phases; the “low-arousal-mode” and “high-arousal-mode”. In each phase different human senses (e.g. audio, visual) are addressed to induce stress. Through correct or wrong user answers a score resulted. In addition, a high score list was visible during the whole experiment to induce social comparison and, thus, stress. The participant’s physical condition (e.g. skin conductance, heart rate variability) is essential for the subsequent stress analysis. Hence medical sensors were integrated into the framework. The participants wore a self-tracking band which had to be attached a certain time before to be calibrated with the skin. Additional, during the game, data from mouse, keyboard and the self-tracking band was collected via the framework. Moreover, we further log the rate of accurate statements in order to assess the user’s current stress level. All generated data were collected and preprocessed by JDCF. Afterwards we analyzed these data to train the analysis algorithm whether the user is stressed or not. This knowledge can be used in future real-time stress analyses. In addition, we observed the result’s impact of medical and non-medical sensors. Regarding the principles of Monte-Carlo-simulations, the order and distribution of training and test data were randomized. For example, we evaluated (non-) stressed-states and measurements with solely technical or medical input data and combined measurements. On average the accuracy for assessment of the user’s stress level for all non-stressed-states is about 99% and for all stressed-states about 70%. A higher accuracy can be achieved through a bigger learning data set. The “arousal-modes” were used as verification. Within the evaluation of the model we also determined the influence of every factor measuring the probability of a decent prediction. For instance, excluding the skin conductance data decreased the accuracy by 5.1%. Similar, not considering the skin temperature data meant a decrease by 4.9%.

In summary we were able to successfully apply the framework that enabled us to combine individual chosen sensors and to gather data. Afterwards we were able to determine a person’s stress level. As a result of the experiment, we were capable to verify our approach and to substantiate that combining different data with sensor fusion improves the accuracy of an assessment whether a user is stressed or not.

IV. OUTLOOK

As demonstrated the framework supports collecting and combining heterogeneous data in the area of individual health management. Furthermore, the framework can be used in the other medical sectors, e.g., medical application platform and telemedicine. Depending on medical applications, analyzing represent a complex challenge. Thereby data of diverse kinds need to be handled. The proposed framework manages to handle varying datasets by implementing every issue as a single sensor enabling various ways to save, combine and analyze them. Telemedicine or rehab represent the idea of clinical health care at a distance. Thereby, verifying the accurate execution of rehab exercises is a huge challenge. JDCF can simplify this task by integrating the mobile phone as sensor. By holding the phone within the execution of exercises the patient data will be collected via the framework and can analyze the efficiency of exercise afterwards. Recently, an executable version for android is tested to enable

JDCF to run on the mobile phone itself for further flexibility concerning location and time. Hence, the functionality of JDCF will be proven in further different use cases, while refining the system itself with every single instantiation.

V. CODE OF ETHICS

According to Deutsche Forschungsgemeinschaft an ethic approval is only required for studies which expect risks or emotional or physical drain of participants. However, our study did not fall into one of these categories, thus was exempt from ethic approvals. Nevertheless, we satisfy the code of ethics by respecting human rights, dignity and privacy and did not harm any living being.

VI. CONCLUSION

We presented JDCF, a generic medical sensor fusion framework, which simplifies collecting data from different sources and combine arbitrary sensors in a consistent way. The architecture and components were described, as well as functionality and versatility. We exemplarily demonstrated the usage and functionality of JDCF by assessing the stress level of a user during a self-advanced stress game. The collected data were reprocessed with data-analytics techniques to recognize stress symptoms. We were able to prove the applicability of JDCF, the stress detection and the added value of medical sensors in stress assessment by excluding gradually sensors and showing the decreasing accuracy. As JDCF is able to generically integrate seamlessly new sensors, it is qualified to be used in different fields of health care or other personalized life science applications.

REFERENCES

- [1] Van Der Wal, A. J., and Shao, M., “Sensor fusion: the application of soft computing in monitoring and control for railroad maintenance,” in *Intelligent Control and Automation, Proceedings of the 3rd World Congress on, Vol. 1*, pp. 341-346, 2000 IEEE.
- [2] Hatcliff, J., King, A., Lee, I., Macdonald, A., Fernando, A., Robkin, M., Vasserman, E., Weininger S., and Goldman, J. M., “Rationale and architecture principles for medical application platforms,” in *Proceedings -IEEE/ACM 3rd International Conference on Cyber-Physical Systems*, pp. 3–12, 2012
- [3] Couderc, P. and Kermarrec, A.-M., “Improving Level of Service for Mobile Users Using Context-Awareness,” *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, pp. 24-33, 1999
- [4] Dey, A. K., Abowd, G. D., and Salber, D., “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications,” anchor article of a special issue on Context-Aware Computing. *HCI Journal*, Vol. 16, pp.97-166, 2001
- [5] Ranganathan, A., and Campbell, R. H. , “A Middleware for Context-Aware Agents in Ubiquitous Computing Environments,” *Proceedings of Middleware, ACM/IFIP/USENIX International Middleware*, pp. 143-161, 2003
- [6] Lee, H., Park, K., Lee, B., Choi, J., and Elmasri, R., "Issues in data fusion for healthcare monitoring," *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. ACM, p. 3, 2008.
- [7] Adam, M. T., Gimpel, H., Maedche, A., and Riedl, R., "Design blueprint for stress-sensitive adaptive enterprise systems." *Business & Information Systems Engineering* 59.4, pp. 277-291, 2017.
- [8] Alemdar, H. and Ersoy C., "Wireless sensor networks for healthcare: A survey," *Computer Networks* 54.15, pp. 2688-2710, 2010.
- [9] Schaaff, K. and Adam, M. T., “Measuring emotional arousal for online applications: evaluation of ultra-short term heart rate variability measures,” in *Affective Computing and Intelligent Interaction (ACII), Humaine Association Conference on* (pp. 362-368). 2013, IEEE.