Context-Aware Healthcare Adaptation Model for COPD Diseases

Hamid Mcheick[1(✉)], John Sayegh[2], and Hicham Ajami[1]

[1] Department of Computer Sciences and Mathematics,
University of Québec at Chicoutimi, Chicoutimi, QC G7H 2B1, Canada
{Hamid_mcheick, Hicham.ajamil}@uqac.ca
[2] Lebanese University, Hadath-I, Beirut, Lebanon
johnsayeghl994@gmail.com

Abstract. Nowadays, ubiquitous computing and mobile applications are controlling all our life's aspects, from social media and entertainment to the very basic needs like commerce, learning, government, and health. These systems have the ability to self-adapt to meet changes in their execution environment and the user's context. In the healthcare domain, information systems have proven their efficiency, not only by organizing and managing patients' data and information but also by helping doctors and medical experts in diagnosing disease and taking precluding procedure to avoid serious conditions. In chronic diseases, telemonitoring systems provide a way to monitor the patient's state and biomarkers within their usual life's routine. In this article, we are combining the healthcare telemonitoring systems with the context awareness and self-adaptation paradigm to provide a self-adaptive framework architecture for COPD patients.

Keywords: Software architecture · Self-adaptation · Context-aware system · COPD · Healthcare systems

1 Introduction

Chronic obstructive pulmonary diseases (COPD) have attracted research interest as a major public health problem, because according to the World Health Organization (WHO) [1], COPD is currently considered the fourth, and will soon become the third, most frequent cause of death worldwide. It is also a disabling disease and therefore associated with high costs for treating and managing patients. As the disease progresses, patients become more susceptible to respiratory exacerbations which cause frequent hospital admissions and, thus, have a huge impact on patients' quality of life and healthcare costs [2, 3]. Monitoring patient's health conditions from home and transmitting these data to a healthcare center could be a great solution that facilitates the management of the growing number of patients with COPD and reduce the burden on health services. This approach is called Home Tele-monitoring, which can be used for a timely assessment of acute exacerbation or as a mechanism to generate alarms to the patients and/or healthcare professionals when clinical changes that may constitute a risk to the patient occur [4]. There are many systematic reviews and studies on the topic of telemonitoring in respiratory patients, specifically in patients with COPD [5–8]. All these

studies are focusing on proving the effectiveness of applying home telemonitoring on COPD patients, by studying the services that could be provided and their impacts on the patient's quality-of-life. However, none of these studies has provided a comprehensive proposal for a telemonitoring system that helps to control the burden of COPD.

We aim to design a telemonitoring healthcare application that helps COPD patients with self-management and improve their quality of life, therefore reducing pressures on healthcare resources. We must develop a system that uses this data to provide an effective intervention that prevents exacerbation through the early recognition of symptoms and prompt treatment which may reduce the risk of hospitalization and control the burden of COPD. Based on this healthcare requirement, we realized the need of combining context awareness and self-adaptation with health telemonitoring, which will give our system the ability to be aware of the patient's data and context, then to adapt the required changes and act accordingly.

The remainder of this paper is structured as follows. Section 2 introduces the concept of context-awareness and reviews the most common forms of self-adaptation frameworks. Section 3 highlights the characteristics of self-adaptive systems. Section 4 presents our self-adaptation healthcare system for COPD. Section 5 then validates the proposed approach. Finally, Sect. 6 presents our conclusions.

2 Context Awareness and Self-adaptation Systems

2.1 Background

The notion of context has appeared implicitly for the first time in the ubiquitous computing area in 1993 by Weiser "all the information that should be taken into consideration for an adjustment" [9]. Lieberman et al. [10] proposed another interpretation of the context that exists within the field of computing: "context can be considered to be everything that affects the computation". This definition focuses on the application instead of the user, but nowadays with the widespread of mobile applications that focus on the user's lifestyle, health, and activities the factors that affect the user and these that affect the computation process become almost the same.

In software systems, context awareness notion is mostly coupled with self-adaptation capability otherwise, there is no point in collecting contextual data. Self-adaptation is a set of simultaneous and successive processes as an organized reaction to changes in the resources or environment of the system [11]. Self-adaptive systems dynamically modify their behaviors to respond effectively to changes in their operational environment [12]. In the next section, an overview of many self-adaptation frameworks is provided.

2.2 Self-adaptations Frameworks

Rainbow framework provides general mechanisms for developing reusable self-adaptive systems at a variety of different levels [13]. Model-Driven Approach is an automated self-adaptive model that supports adding and removing technical resources at run-time [14]. Meta-Self is a service-oriented framework that provides a solid platform

for the development of SAS [15]. This framework allows designers to identify system properties, architectural patterns, and different adaptation mechanisms. FUSION is a reusable feature architecture that incorporates a learning-based adaptive cycle. The adaptation cycle consists of three main steps—detect, plan, and effect [14]. MOSES is a service-oriented framework that focuses on quality of service (QoS) requirements at runtime. MOSES provides a reusable implementation strategy of the adaptation logic following the MAPE cycle (Monitoring, Analysis, Planning, and Execution) [14]. The Contract-Based Adaptive Software Application framework (CASA) [16] is specialized in handling resources instability. The framework assumes that a system should not make any assumptions about the resources that will be available and should be prepared for any resource availability scenario cases. Service-Oriented Architectures (SSOA) is a software framework that specifies any kind of adaptation by decomposing of functionalities [17]. Each of these functionalities shall be specialized to fit a particular purpose. CareDroid is an adaptation framework for android context-aware applications [18]. This framework CareDroid monitors the contexts at run-time, and active methods only when it intercepts calls to sensitive methods.

3   System Requirements and Self-adaptation Characteristics and Taxonomy

3.1   Requirements Extraction and Gathering

The first step to designing a self-adaptive system is to well identify the system requirements; we will depend on W5H-Pattern [19], which presents six questions that would help us in eliciting adaptation requirements (Table 1).

Table 1. Requirements extraction

Where	Where do we need to make a change inside our system when a context's change does happen? Depending on the model presented by Ajami and Mcheick [20], the change needs to be done in the Application layer on both sides: user interface and physician interface
When	When do we need to make these changes? Whenever an urgent update happens in the user contextual data like vital signs, environmental risk factors, and planned activities or periodical changes like the evaluation of treatment and decision support suggestions
What	What do we need to change? We need to update some system attributes that present the system state and these attributes in its turn could trigger new functions or activate new components
Why	Why these changes are required? In healthcare monitoring applications especially these related to chronic diseases, taking precluding actions is crucial in treatment plans. Also being able to notify the patient and the medical experts about any threatening situation or abnormal signs make these kinds of applications more efficient
Who	Is any human intervention is required in the adaptation process? From the patient side, all his biomedical data and surrounding environments data will be collected from sensors. However, because physical activities do affect the COPD patient's state, he needs to detect his planned physical activity (running, swimming)
How	How to determine what changes and actions are needed to be done in the adaptation process? Ajami and Mcheick [20] provided a rule-based reasoning engine, depending on these generated rules all the required actions and changes can be deduced

3.2   Adaptation Characteristics and Taxonomy

Christian et al. [21] presented a taxonomy of the different properties of self-adaptive software. We will analyze this work and do a projection on our system requirements and use the results to build our system.

Time: Handte et al. [22] provided two perspectives of temporal aspects: (i) Reactive is when we have to adapt whenever a change in the context does happen. (ii) Proactive is when the monitored data is used to forecast system behavior or environmental state [21]. In our case, the adaptation will be reactive depending on the changes that happen in the user contextual data.

Reason: The adaptation could be triggered for three reasons: (i) change of the context, (ii) change in the technical resources, and (iii) change in the users. In our case, the adaptation is triggered due to contextual changes, which provide a potential solution for the multiscale nature of COPD.

Level: In our system, the change needs to be done on the application layer, where we need to update the acceptable range for the different datasets or we need to activate new components or call new functions.

Technique: McKinley [23] provided two techniques for adaptive software: parameter adaptation and compositional adaptation. Parameter adaptation achieves a modified system behavior by adjusting system parameters. Whereas compositional adaptation enables the exchange of algorithms or system components dynamically at runtime. We will use the first approach because it is suitable for a rule-based system.

Adaptation Control: Two approaches for implementing the adaptation logic can be found in the literature. The internal approach, which twists the adaptation logic with the system resources. The external approach splits the system into adaptation logic and managed resources, The IBM Autonomic Computing Initiative provided MAPE Model [24], which is an external, feedback control approach. Another aspect of the adaptation logic is the degree of decentralization. We will follow a decentralized approach by implementing independent units that control different aspects of adaptation.

4   Self-adaptation Healthcare System for COPD

4.1   Proposed System

Ajami and Mcheick [20] have proposed an ontology-based approach to keep track of the physical status of patients, suggest recommendations and deliver interventions promptly, by developing a decision support system based on an ontological formal description that uses SWRL rules. The main goal of this paper is to provide an adaptation architecture design for the application layer, which will address the connection between three different entities:

1 - The end-user application: which is supposed to provide a certain service for both patient and physician.

2 - The data sources (sensors and patient's records): that provides a continuous stream of contextual data and historical data about the patient.

3 - The rules base: which presents the knowledge base in our system (Fig. 1)
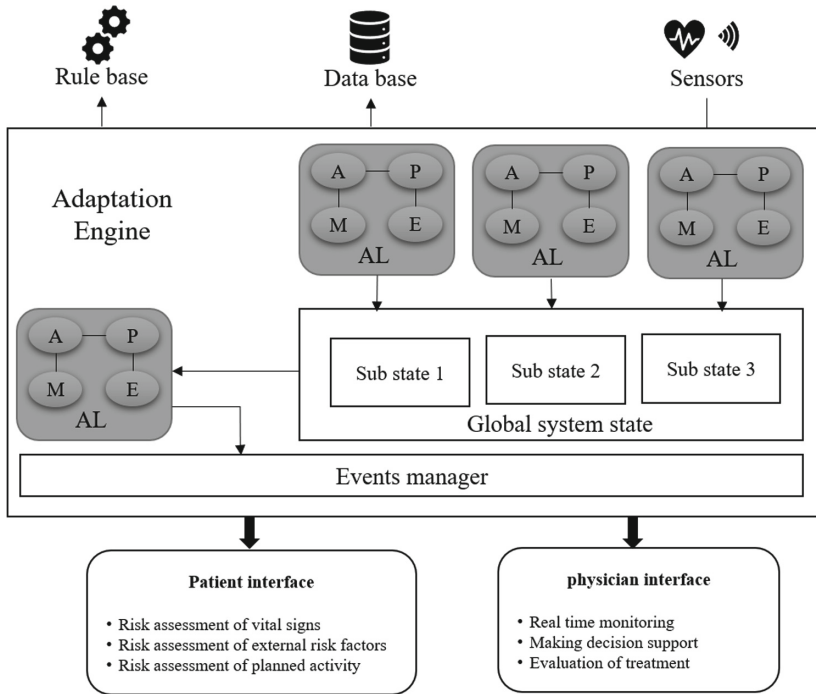
Fig. 1. Architecture for COPD context-aware system

## 4.2 The Adaptation Engine and Monitoring Units

The adaptation engine consists of a central adaptation unit and multiple sub adaptation units. Each subunit is responsible for monitoring and managing changes for a specific category tuple (data, rules, services).

The system variables will be saved in a shared memory called the global state, which is a composition of sub-states. Each sub-state is considered as a container for saving category-specific data and it will be updated and managed by the adaptation subunit that is responsible for monitoring the same category. We will divide all our sets of data, rules, and services into three categories (Table 2):

Table 2. Categorization of data

	Biometrics	Environmental	Activities
Data	Biometrics data	Environmental data	Activities data
Rules	Biometrics rules	Environmental rules	Activities rules
Services	Biometrics services	Environmental services	Activities services

Now each subunit will be responsible for a specific category of the tuple. Therefore, we will have three subunits: 1 - The biometrics unit 2 - The Environmental unit 3 - The Activities unit (Fig. 2).
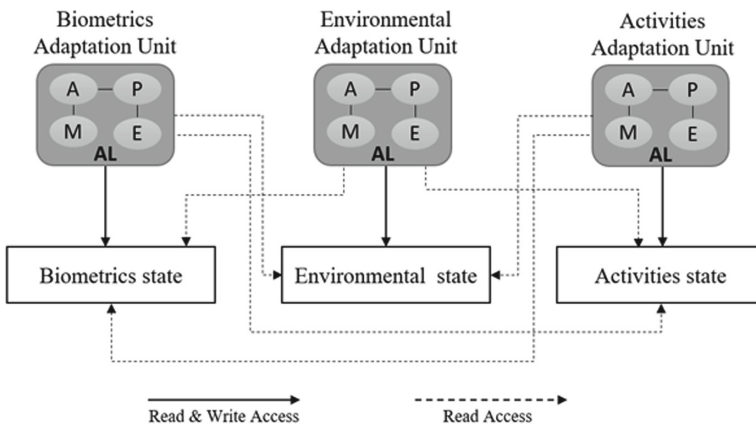
Fig. 2. Subunits and sub stats

– The Biometrics Unit will be responsible for monitoring all the patient's biomarkers upcoming from the biometrics sensors, and it will read all the stored data in the other two sub-states (the environmental state & the activities state), then depending on the vital related rules in the rules engine, it will update The Biometrics State with the safe ranges for all the vital biomarkers and the current measurements for them. The same workflow will be applied in both the Environmental Unit and the Activities Unit.

– The Central Adaptation Unit: it is considered the main core of our system; it will be responsible for monitoring the global state, which will get rid of the burden of dealing with the continuous streaming of the patient's biometrics and the environmental data. By collecting all the contextual data in the global state, each category in its sub-state, we will have access to all the current biometrics and external factors value with the safe range for each one of them as well the current physical activity and the planned list of activities.

Depending on the previous data, the central unit will be able to detect any potential risk or abnormal situations by comparing the current value of each factor in the sub-states with its normal range, which had been adapted by every sub adaptation unit.

When an abnormal situation is detected, the central unit will detect what action should be taken to prevent an exacerbation in the patient's health state.

5   Validation

In order to test and validate our proposed system, we implemented a simulation app using some data obtained from medical records to simulate the streamed data and a set of COPD rules extracted in [25] to create some testing scenarios. The main focus of the validation process was on the efficiency of the system to provide continuous monitoring of the patient status, and the ability to apply and adapt the required changes to prevent any dangerous exacerbation. The testing scenarios we had performed, proofed the ability of our system to handle the complexity of monitoring the enormous amount of contextual data, and keep track of the latest updates in the global state. Also, following an aspect-oriented approach facilitates the implementation of the adaptation logic, by separating the categories of data that each Adaptation Unit needs to be responsible for observing. After testing some rules that lead to call a sequential set of actions and multiple updates in the state units, the system was able to adapt the safe ranges for the different environmental and biometrical factors and detect suitable action in an abnormal situation. Nevertheless, our system still needs to be tested when it is connected to the whole rules engine when all COPD rules are inserted into the engine, which will be done in future work.

6   Conclusion

In this paper, we have presented an architecture for a context-aware self-adaptive system that is used to develop a COPD healthcare telemonitoring system. The system is backed out by a medical rules engine in the COPD domain that is used as the knowledge base to determine the safe ranges for patient's biomarkers and external factors, then detect the precluding actions needed to be taken to prevent severe exacerbations in patient's health state.

Our main contribution in this work is providing a context-aware self-adaptive system architecture that is dealing with the huge variety and complexity of contextual data and different sets of services by implementing a decentralized adaptation unit, which makes the monitoring and adaptation task easier and less complex by applying the separation of concerns principle.

References

1. The top 10 causes of death. In: World Heal. Organ. http://www.who.int/mediacentre/factsheets/fs310/en/. Accessed 3 Jun 2017
2. Casas, A., et al.: Integrated care prevents hospitalisations for exacerbations in COPD patients. Eur. Respir. J. **28**(1), 123–130 (2006)
3. Global Strategy for the Diagnosis Management and Prevention of COPD, Global Initiative for Chronic Obstructive Lung Disease (GOLD) (2011)
4. McKinstry, B.: The use of remote monitoring technologies in managing chronic obstructive pulmonary disease (2013)
5. Bolton, C.E., Waters, C.S., Peirce, S., Elwyn, G.: Insufficient evidence of benefit: a systematic review of home telemonitoring for COPD. J. Eval. Clin. Pract. **17**(6), 1216–1222 (2010)
6. Polisena, J., et al.: Home telehealth for chronic obstructive pulmonary disease: a systematic review and meta-analysis. J. Telemed. Telec. **16**, 120–127 (2010)
7. Jaana, M., Pare, G., Sicotte, C.: Home telemonitoring forrespiratory conditions: a systematic review. Am. J. Manage. Care **15**(5), 313–320 (2009)
8. Bartoli, L., Zanaboni, P., Masella, C., Ursini, N.: Systematicreview of telemedicine services for patients affected by Chronic Obstructive Pulmonary Disease (COPD). Telemed. J. E Health **15**(9), 877–883 (2009)
9. McLean, S., Nurmatov, U., Liu, J.L., Pagliari, C., Car, J., Sheikh, A.: Telehealthcare for chronic obstructive pulmonary disease: Cochrane Review and meta-analysis. Br. J. Gen. Pract. **62**(604), e739–e749 (2012). https://doi.org/10.3399/bjgp12X658269
10. Lieberman, H., Selker, T.: Out of context: computer systems that adapt to, and learn from, context. IBM Syst. J. **39**, 617–632 (2000)
11. Krupitzer, C., Roth, F.M., VanSyckel, S., Schiele, G., Becker, C.: A survey on engineering approaches for self-adaptive systems. Pervasive Mob. Comput. J. **17**(Part B), 184–206 (2015)
12. Oreizy, P., et al.: An architecture-based approach to self-adaptive software. IEEE Intell. Syst. **14**(3), 54–62 (1999)
13. Cheng, S.-W.: Rainbow: cost-effective software architecture-based self-adaptation. Ph.D. Dissertation. Carnegie Mellon University (2008)
14. Becker: Comparison of approaches for developing self-adaptive systems. In: Proceedings of ACM, New York, NY, USA, 14 p. (2018)
15. Di Marzo Serugendo, G., Fitzgerald, J., Romanovsky, A.: MetaSelf - an architecture and a development method for dependable self-* systems. In: Proc SAC, pp. 457–461. ACM (2010)
16. Mukhija, A., Martin G.: CASA a contract-based adaptive software architecture framework. In: Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN 2003), Berne, Switzerland, pp. 275–286, July 2003
17. Andr, F., Daubert, E., Gauvrit, G.: Towards a generic context-aware framework for self-adaptation of service-oriented architectures (2010)
18. Elmalaki, S., Wanner, L., Srivastava, M.: CAreDroid: adaptation framework for android context-aware applications. In: Proceedings of Annual International Conference on Mobile Computind Network (2015)
19. Gaasbeek, J.R., Martin, J.N.: Getting to requirements: the W5H challenge. In: Proceedings of the 11th International Symposium of the International Council on Systems Engineering (2001)

20. Ajami, H., Mcheick, H.: Ontology-based model to support ubiquitous healthcare systems for COPD patients. Electronics **7**, 371 (2018). https://doi.org/10.3390/electronics7120371
21. Krupitzer, C., Roth, F.M., Van Syckel, S., Schiele, G., Becker, C.: A survey on engineering approaches for self-adaptive systems. Pervasive Mob. Comput. **17**(Part B), 184–206 (2015)
22. Handte, M., Schiele, G., Matjuntke, V., Becker, C., Marrón, P.J.: 3PC: System support for adaptive peer-to-peer pervasive computing. ACM Trans. Auton. Adapt. Syst. **7**(1) (2012). Art. 10
23. McKinley, P., Sadjadi, S., Kasten, E., Cheng, B.H.C.: Composing adaptive software. IEEE Comput. **37**(7), 56–64 (2004)
24. IBM: An architectural blueprint for autonomic computing (2004)
25. Ajami, H., Mcheick, H., Mustapha, K.: A pervasive healthcare system for COPD patients. Diagnostics **9**, 135 (2019)