Design and implementation of a context-aware health service platform (CAHS)

Siham Belhadi, Rachid Merzougui
Department of Telecommunication, Tlemcen University, Algeria

Article Info

ABSTRACT

Computers are no match to humans in deducing situational information from their environment and in using it in their interactions. The advent of the context-aware applications seems to offer a way out to the computer that is not context-sensitive. The context aware applications can adapt their behaviors according to the perceived context or situation, without explicit user intervention, thereby providing human-centric services. To simplify the complexity of developing applications, context aware framework, which introduces context awareness into the environment where the applications are executed, is highlighted to provide a homogeneous interface involving generic context management and adaptation solutions. This papier has focused on the design of Context-Aware Health Services (CAHS) platform, which provide a health applications framework embedded on mobile devices. Our proposed platform is capabilities for context manager and adaptations according to context changes. It is designed to base on the SOA principles for achieving a flexible and dynamic architecture.

Corresponding Author:

Siham Belhadi,
Department of Telecommunication,
Tlemcen University,
STIC Laboratory, Algeria.
Email: belhadi.ttl@gmail.com

1. INTRODUCTION

Nowadays, mobile technology is increasingly being used by all age groups of people. Moreover, the appearance of more and more mobile applications in different domains has helped to install various applications on mobile devices. Unfortunately, this has had a negative impact on their capacity, speed, etc. The context-aware framework has been introduced for the purpose of overcoming this problem. Its main objective is to provide the execution environment that helps to manage applications and also to adapt to that environment. The present paper focuses on the new *concept of mobile health (m-Health)* which has become an important sub-segment of the field of electronic health (e-Health) [1]. The different mobile health applications can create, collect, store and also transmit information in order to ensure the safety of both the patient and the physician [2]. In order to make these applications smarter and more automatic, the context-aware framework [3-8] for health applications is proposed for the purpose of improving the quality of health care.

Context-aware health services need systems [9] that can automatically acquire personal information, i.e. the living environment, physical characteristics, activities; they must be able to provide and adapt their services accordingly, as shown in Figure 1. Therefore, automatic management of mobile health services and their adaptation to the context leads to propose a context-aware framework through the usage of mobile devices. This proposed framework is commonly called the Context-Aware Health Services (CAHS) framework.
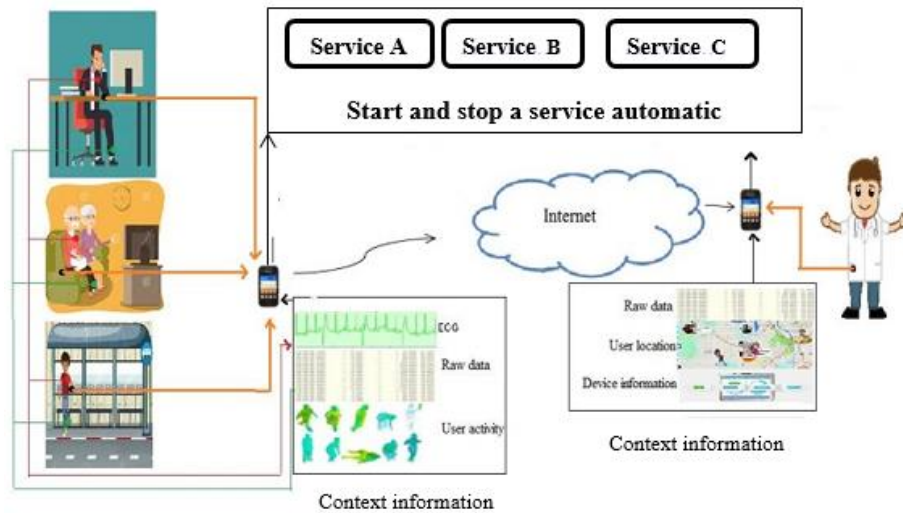
Figure 1. Architecture of the platform system

CAHS have capabilities for offering the manager of the context and services by using the technique of context-awareness and the adaptation. This framework can be designed by dividing their functionalities into modules. For a same functionality, we can provide multiple implementations, each suited for a different context situation.

2. RELATED WORK

Our interest revolves around mobile health and the associated topics. This domain knows a major development in recent times. We mention Vital Wave Consulting [1] who define a mobile health as "Using mobile communications-such as PDAs and mobile phones-for health services and information". The mobile health survey [2] presents the various mobiles health applications, which aims to help the patients and the health professionals. This survey arrives the important issues such as the cooperation between m-Health applications that need a more comprehensive study.

Using the mobile applications in a health services have an operational constraints that make them unsuitable if the execution conditions are changing rapidly. This allows creating a context-aware platform. There are three important axes: the context, the context manager, and the adaptation strategy that allows for the flexible and the adaptable service execution.

In the area of context, the several researchers were defined of the context some of them was based on the enumeration of the contextual information (localization, time, date, temperature, identity, environment, etc.) like proposing in [10-12], and the others researchers were given the definitions which befit their studies, as in [13] is defined the context as "All the information which starts a service or changes the form of a service if its changes of value". However, the definition most general suggested by [14] which is defined as " any information can be used to characterize the situation of entity which is person, place, and object". We consider this entity as the set elements, which represent all aspect, which influences on the mobile service execution in the health domain.

The second issue concerns the context manager which is composed the various step (acquisition, classification, and model). There are the differents methods of the context manager. In a related work [15], we mentioned certain these methods, which are used in the proposed context aware framework dedicated to the different domain (health, transport). Authors of the work [16] propose the manager of the context elements. This context manager can evaluate the state of the context on demand, a continuous basis, when certain elements need to be monitored. But, certain elements has monitored in the deferent time so they not relate to the relation between the context elements for starting their monitor.

We would propose the context manager for the health services. In this domain, there are the differents types of the relations between the context information. These relations must be used to ensure the best presentation and the proper interpretation of the context elements in order to establish a correct behavior of the running applications. For this, we would like to create monitor for each context element, and to start a group of the monitor in the different time. This provide the relation between the elements for facility of the service and the context manager, and the reduce time of the reaction.

A third issue of an importance in our platform is the adaptation strategy. Obviously, there are the various research papers focus on the theme of an adaptation [17-19]. So, there are mainly four kinds of an adaptation: a) content adaptation, b) adaptation of behavior (services), c) presentation (or interface) adaptation and software components adaptation [20] and [21]. In work [13] proposed a context-aware dynamic service adaptation approach. the service adaptation can be done in two ways: a) automatic triggering of a service according to the current context or b) changing the quality (form) of a provided service according to the current context because one or more of the contextual information has or have changed its or their value(s). According to [16], the author has chosen to follow the principles of Service-Oriented Architecture (SOA). A service is an interface for representing the contract between the service providers and clients. The service can have the same functionality with the different implementations, each one adapted to a certain context situation. They call "equivalent services" the different implementations of the same interface, Service S (1) and Service S (2).

Based on related works, Popovici *et al*. [16] developed the Context-Aware Transport Service (CATS) framework and presented its advantages in managing context and services, and providing the interaction between various applications. Therefore, it is interesting to suggest the context-aware health service (CAHS), which is a potential version of the CATS framework. This context includes the functionalities of context manager and adaptation strategies that are suitable for the health domain, which is characterized by various relations between information and its context. For this purpose, the present work makes an attempt to create a context classification in order to ensure the best presentation of these relations. The context manager of this framework evaluates each context element, using the proposed classification. On the other hand, the combination of the two approaches proposed in [13] and [16] allows creating a new strategy that offers flexibility in the adaptation of services.

3. CONTEXT-AWARE HEALTH SERVICES (CAHS) FRAMEWORK

Our proposed model related to the health-oriented applications for the handheld mobile devices and their adaptation to the current context. In this section, we will give a view about the term "context and context-aware". Then we will present "context-aware service adaptation" for designing of the mobile applications.

3.1. Health

We focus about a health domain, because this domain has a certain constraints. In addition, the change in the context necessarily effects on the execution of the some services. All health services should be able to execute on the different contextual situation at any time, this concept is called an adaptation depend to the context. Our solution is based on a manager of the context and the services, in a context-aware framework, which has a capability the surveillance of two or more maladies. Since, among services that our framework must provide we include displaying and analyzing of the different medical data and the wireless communication between the patients and their doctors. In our solution, the application must choose a series of an appropriates services according to the current contextual situation. Moreover, the services must exist in any time. For this, we prefer local services (i.e., having all needed services on the device), as they avoid the time losses to take a decision on the patient's situation.

3.2. Context and context-aware

Nowadays, the main axe for building a mobile application is the context. We took this notion as a started point for our framework CAHS. We identified the elements that influence the health services execution in the particular situation of the users. We propose a simple generic structure for representing any context element (CE) in the Table 1.

Table 1. The characteristic of the context element

Characteristic	Context element
Name	The name of the context element
Type	The type of an information of this context element
Value	The value of this context element at the particular time
Class	The relative classification

A context element is identified by its name, which must necessarily be unique. It is characterized by its value at a given moment, and its class. We identify the three types that can be presented the CE: Boolean, discrete, and continuous [16].

In our work , the service adaptation can be done in two steps: a) specifying of a service which must be trigger according to the current context or b) identifying of the appropriate service version according to the current context because one or more of a contextual information has or have changed its or their value(s). This is enough abstract and limits the set of a contextual information. The Figure 2 shows the services and the sets of information that causes the adaptation, which considered as parts of the global context.

In now computing, the applications react to the changes in their context. These capabilities are called context awareness [9], which requires the contextual information . these information must be collected and presented to the adaptation application. Because these information are charactered by diversity, heterogeneity and great quality, we suggested to classify these information in order to make the adaptation operation easier. Our classification of the context elements was presented in the following title. The state of the context changes according to time. This event allowed to use modules called Context Monitors. Context Monitors monitored the value of the single context element.
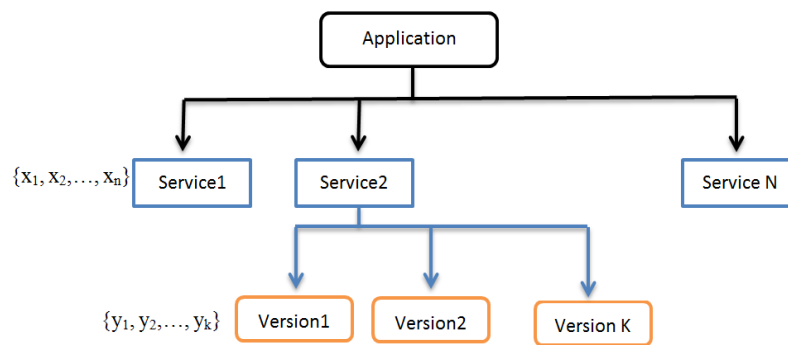
Figure 2. The application composition

3.3. The context classification

There are the different types of the relations between a context information. these relations must be used to ensure the best presentation and the proper interpretation of these elements in order to establish a correct behavior of running applications. In order to the relationship is the result of the certain dependence between the context elements. For this reason, it is suggested to classify all the important context elements in the form of a tree, as reported in [3]. This classification presented in the Figure 3.
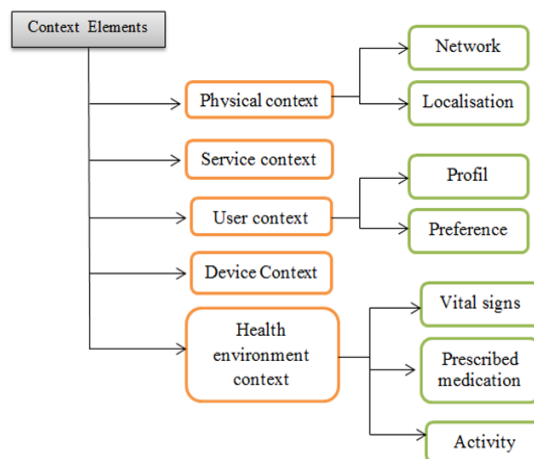
Figure 3. The context tree

This classification allows the facility of the context and the service manager and the reduce time of the reaction in the case of the context changes. We have made a classification in two categories that seems to us more expressive and helpful: a) a context information whose are responsible for starting or stopping a service version and b) a context information that used by the started service. This classification include five

the main classes as used Service context, Physical context, User context, Health environment context, and Device Context. It is more expressive and complete, because it covers all aspects of the context, which influences on the execution of the mobile applications dedicated to the health domain, and facilitates the task of an adaptation. It is more organized, because each context element contains the sub-elements, that have the different types of the relationships between them, for example, a hypertensive patient, a change in the activity context value (still eating) may affect the value of his vital signs (increase in systolic pressure), these relationships facilitate the context manager.

3.4. The context-aware service adaptation
The adaptation strategy is based on starting an appropriate service according to the current context. For each context situation, the set of a services can start. each service has the set of a versions (in our work, the analysis service has two versions). For each service, there are a set of contextual information, which is responsible for starting the service version.

Our proposed work, the context tree consists a five classes, which noted Cct ={Service context, Physical context, User context, Health environment context, Device Context}. For each class has a matrix (MC) which presents an information of a monitored context element. On the other hand, the service S can have the same functionality with the different implementation; each one adapted a certain situation. We call "service version" the different implementation of the same interface. A service will be started a specific version if the context situation is an appropriate to this version. In addition, for each service has a default version that assure the continuous of an execution when the situation contextual is not verified.

A principle of the adaptation is based on a correlation between the context elements and the services. A matrix MC used in this strategy that allowa to identifying the relation between CE and S. The dimension of a matrix MC (i, j) defined as following: i: specifies an identification of CE in its Class, and j: specifies a version of the used service.

The matrix coefficient(α) is equal 0 or 1 (α = 1, if CE is monitored, or α= 0, if CE is not monitored). So, if class= Health Environment Context (HEC) has three CEs which presented the information that focus on the health domain(Vital signs , Activity , Prescribed medication) then the matrix (MC) of this class is presented as following:

$$MC_{HEC} = \begin{matrix} & Vs_1 & Vs_2 \dots & Vs_m & \\ & \begin{bmatrix} \alpha_{11} & \alpha_{12\dots} & \alpha_{1m} \\ \alpha_{21} & \alpha_{2\,2\dots} & \alpha_{2m} \\ \alpha_{31} & \alpha_{32\,\dots} & \alpha_{3m} \end{bmatrix} & \begin{matrix} CE_1 \\ CE_2 \\ CE_3 \end{matrix} \end{matrix}$$

From a coefficient are determined the contextual situation. Since, all coefficient values are 1, we consider to create the current contextual situation for specifying the adapted service version. Therefore, the adaptor picks the services version according to a contextual situation.

4.   ARCHITECTURE OF THE CONTEXT-AWARE HEALTH SERVICES (CAHS) FRAMEWORK
As shown in the Figure 4, the framework CAHS provides an execution environment that adapts the health services depending to the context situations at run time, without explicit demand from the user. On the other hand, the framework aims to manage of the set applications that monitor of several patients with the different chronic diseases. Therefore, CAHS is composed of three specific modules (context manager, adaptor, controller-SDC) which function continuously as long as the framework is started.
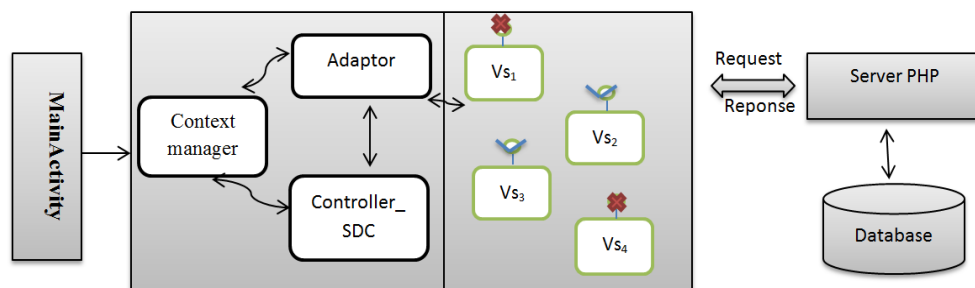

Figure 4. The CAHS framework architecture

4.1. The context manager module

The context manager module is responsible for monitoring the certain context elements in the different time. Each context element has monitor their value and trigger the other monitors of the context elements which belong to the same class (device context, user context, environment health context, physical context, service context) by using file XML CDC. The process about the context manager described in the Figure 5. The context manger contains two components:
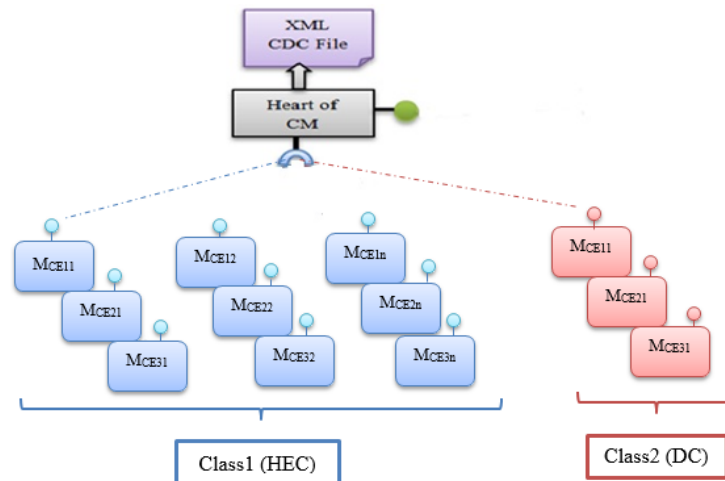

Figure 5. The context manager

a. Monitor of CE

Monitor of CE allows to evaluate the context state. It acquires a value of the context element. This is allowed to reduce the charge at the level of Heart of context manager. The contextual situation is consisted by the set of the context elements and its presentation is:

$$Sc_j = \{CE_{1j}, CE_{2j}, \ldots, CE_{nj}\} \quad j=1,\ldots, m.$$

For each context element has the monitor. The monitor is noted MCE_{ij} such as: i presents number the context element in Class, and j presents number the situation which created by the set of context element.

b. Heart of context manager

Heart of context manager is noted "Heart of CM". Heart of context manager is the main pole in the context manager module. It is responsible for reading the Xml file of the context classification, starting the needed monitor of each context element, and notifying Conctrollor-SDC on the contextual situation. In case the first use, Heart of context manager determins the user information. For this, heart of CM create two txt file: first file is named Profil.txt, which contains an information of the user (First name, family name, age, status (patient or doctor), username, password), and other file is named Status-User.txt. This file is created when the user is patient. Prescribedmedication.txt contains an information of the patient (maladies, number of medication, name of medication, the time to take a medication, and dose of medication).

4.2. Controller-SDC

The main goal of Controller-SDC creates a matrix MC according to the status of MCE_{ij}, as showned in Figure 6. Controller-SDC acts as a correlation point between the context element and the service in our framework. The policies of Contoller-SDC are based on the variation status of MCE. It updates the matrix (MC) for starting the better service version depending to the contextual situation. Typically, the functionality of Controller-SDC is the preliminary step of the adaptation strategy. In the different time, the context manager demands to trigger the monitor of the context elements. These elements belong the same class. In this time, Controller-SDC created the matrix (MC) which depends this monitored class.

This method allows to start the monitors of a context element as a group. It assures two importants points in the context-aware health system:
- Manage the great quantity and the heterogeneity and the diversity of the context information without be the charge in the only component level of the CAHS framework.

- The different types of the relations between the context information. These relations must be used to offer the best contextual situation in order to establish a running appropriate service version.
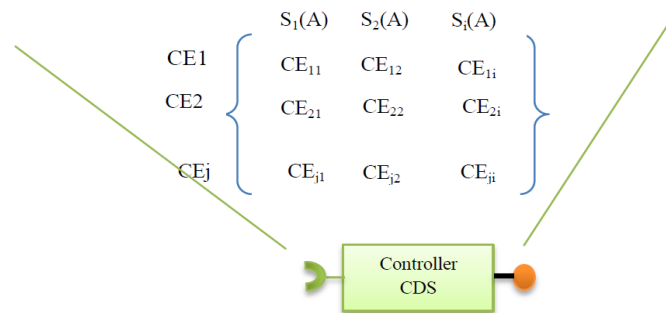

Figure 6. Controller-SDC

4.3. Adaptor

Adaptor is responsible for managing the services as shown in the Figure 7. When creating the matrix (MC) about the monitor status of a context elements from Controller-SDC, Adaptor will checks the incompatibility between the notifications from the Controller-SDC and the service version. We recall the contextual situation (Sc) in the section (3.3). The problem is to find the most suitable service version in the service set existed locally on the device.
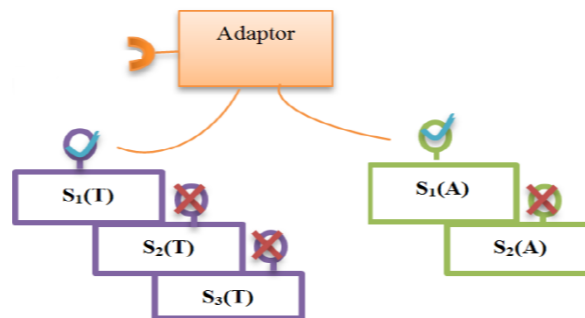

Figure 7. The service manager

The service can have the same functionality with the different implementation. This service takes the same name. On the other hand, for each version of this service wears the different number. In the more detail, the service has the name "T". We can mention the versions of this service $S_j(T) = \{S_1(T), S_2(T),…, S_m(T)\}$. Adaptor compared between the number of the service version and the contextual situation. If the numbers are equal, Adaptor will start this service version according to this contextual situation.

5.   THE IMPLEMENTATION OF THE PROPOSED FRAMEWORK

In this section, we present an implementation of our framework. We have evaluated the behavior of our framework on Android smartphones. We chose to implement this framework using the OSGi technology and the iPOJO [22] component model.

5.1. The CAHS framework with iPOJO

The CAHS Framework is composed by the multiple services compatible with both Android and OSGi. Our framework developed as an Android activity which embeds the Felix framework and Ipojo. Launching Felix inside an Android program is not difficult. It is built of components, which are bound at the execution. Each component must provide its own the functional, and is implemented as bundle running, manages the non-functional capabilities. The manager modules were already described in the previous section (Context manager, Controller-SDC and Adaptor). We introduce the implementation explanation of the manager modules by using iPOJO and Android.

The Context Manager can evaluate the context state on demand, by starting or stopping the monitor of the context element (MCE). In the first time, Heart of CM reads the XML CDC file which presents the context tree as shown in the Figure 3. This reading allows to start the group of the context elements monitors. These context elements belong to the same class. The context manager notifies Controller-SDC on the changed status of CE for creating the matrix (MC). The adaptor is responsible for executing the services. It reads the manifest.fm file of the all services versions that are available on the CAHS framework and can decide to start or stop services based on the matrix (MC).

5.2. The experimental results of the proposed framework

The goal of our implementation is to evaluate the following aspects: the behavior of the CAHS framework on the phones and the different solutions for managing the context, and the adaptation of the health service. We measure the time necessary for installing and starting the services, because this time takes the important place in the adaptation time, which considered as an important factor for estimating the effective of the CAHS framework. For this, we measure also the adaptation time which is the needed time for taking utile the suited all services which are ready for use. In studied domain, among the services that our framework must provide we include displaying and analyzing of the medical data, wireless communication for sending the health situation of the patient to the doctor, and the medical data storage using the PHP web service, essential to an user in the health domain. The user is either a patient or a doctor, for that, he needs the different context element monitor. We would base on two classes of the context element: (1) Health environment context (2) Device context. Here are the used elements in the evaluation:

a. The first element is the context element which presented as following:

CE1 {Name = Blood sugar level; Type = discrete; Class= Health Environment}.
CE2 {Name = ECG; Type = continuous; Class= Health Environment}.
CE3 {Name = Activity; Type = discrete; Class= Health Environment}.
CE4 {Name = Prescribed medication; Type = discrete; Class= Health Environment}.
CE5 {Name = Memory; Type = discrete; Class= Device}.
CE6 {Name = Battery; Type = discrete; Class= Device}.

b. The second element is the monitor of the context element which presented as following:

The context Monitor for "Blood sugar level" is a service that evaluates the Blood sugar level for the diabetic patient.
The context Monitor for "ECG" is a service that evaluates the values of the ECG signal for the cardiac patient.
The context Monitor for "Activity" is a service that detects the activity for the diabetic or cardiac patient.
The context Monitor for "memory" is a service that evaluates the memory of the mobile phone for the different user.
The context Monitor for "battery" is a service that evaluates the battery of the mobile phone for the different user.
The monitors evaluate the context elements every time in the group way. The context elements belong the same class .these context elements evaluated the every same time.

c. The third element is a service which presented as following:

- Display service: it is responsible for displaying the medical data. It has three versions which presented $S(display)=\{S_1(display), S_2(display), S_3(display)\}$

$S_1(display)$ is the first version for the diabetic patient. It depend on Sc (HEC) = {CE$_{11}${name= Blood sugar level }, CE$_{21}${name=activity}, CE$_{31}${ name= Prescribed medication }} and Sc(DC) = { CE$_{11}${name=memory}, CE$_{21}${name=battery}}. We use the android GUI element, as shown in the Figure 8(a).

$S_2(display)$ is the second version for the cardiac patient which depend on Sc (HEC) = { CE12{name=ECG}, CE22{name=activity}, CE32{name= Prescribed medication }} and Sc(DC) = { CE12{name=memory}, CE22{name=battery}}. We use the android GUI element, and the GraphView-4.0.1.jar library, as shown in the Figure 8(b).

$S_3(display)$ is the third version for the doctor which depend on Sc(DC)={CE13{name=memory}, CE23{name=battery}}. We use the android GUI element, as shown in the Figure 8(c) and the Figure 8(d).

- Analyze service: it is responsible for analyzing the medical data. It has two versions, which presented: S (analyze) = {S$_1$ (analyze), S$_2$ (analyze)}.

$S_1(analyze)$ is the first version for the diabetic patient. It depend on Sc(HEC) ={CE11{name= Blood sugar level }, CE21{name=activity}, CE31{ name= Prescribed medication }} and Sc(DC) = { CE11{name=memory}, CE21{name=battery}}.

S_2*(analyze)* is the second version for the cardiac patient. It depend on Sc (HEC) = { CE12{name=ECG}, CE22{name=activity}, CE32{name= Prescribed medication}} Sc (DC) = { CE12{name=memory}, CE22{name=battery}}. It analyzes the ECG signal for detecting the abnormal situation. It allows to determine the Heartbeat, and the QRS duration.

- Communication service: it is responsible for sending and receipting the medical data and the patient situation. For this service, we use Socket. It has two versions which presented S(communication)={S_1(communication), S_2(communication)}.

To demonstrate the capacity and effectiveness of the CAHS framework, it was decided to present the events that occur during its execution, as shown in Figure 8. Time ($T_b + \alpha$) is considered as the duration during which the context is monitored, T_s is the moment the service is selected, T_{on} the moment the selected service is activated and T_{off} the moment when the inappropriate service is disabled. The different times are measured and given in Figure 9.
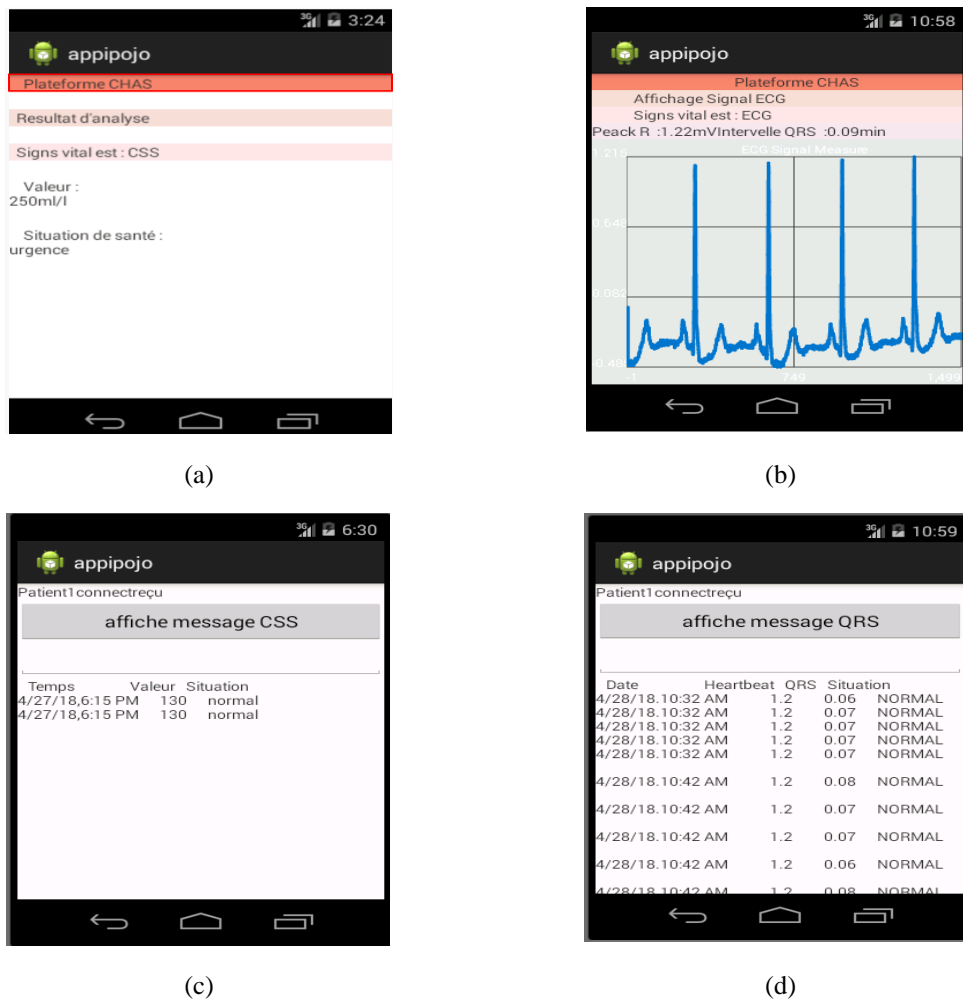

(a)

(b)

(c)

(d)

Figure 8. The CAHS framework: services, (a) S_1 (display), (b) S_2 (display),
(c) S_3 (display) for diabetic doctor, (d) S_3 (display) for cardiology doctor
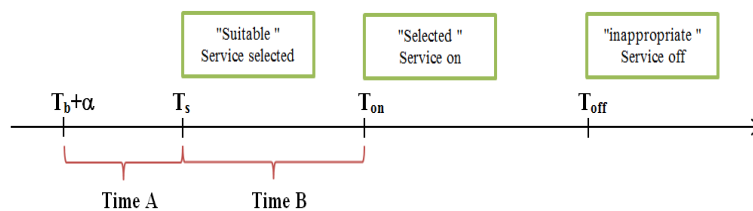

Figure 9. The experimental measurements

Time A - After context monitoring, the MC matrix is developed, from the moment it was created until the appropriate service is selected. The time required to select the adequate service version is represented by the following equation:

$$\text{Time A [Si (T)]} = \text{Ts [Si (T)]} - (\text{Tb} + \alpha)$$

when the monitoring of the context element is completed, then the values of the context elements are evaluated according to Tb. The parameter α represents the period of time required to create the matrix MC. In our evaluation, three versions of the display service and two versions of the analysis service were considered, with m = 3 for the first service and m = 2 for the second one. The service versions were selected from the context situation, during the time period Ts [Si (T)], with i = 1, ..., m.

Time B - This is the period of time required for the CAHS framework to start the version of the selected service. This time is determined using the following equation:

$$\text{Time B [Si (T)]} = \text{Ton [Si (T)]} - \text{Ts [Si (T)]}$$

Ton [Si (T)] is the time when the version of the selected service is started. It should be noted that the context element can take several values. For example, for the disease of diabetes, the value of vital signs is discrete but for the heart disease the value of vital signs is continuous. It is therefore important to distinguish the four cases given below:

Case 1   : The user is a patient who suffers from diabetes. All context elements are discrete.
Case 2   : The user is a patient with a heart disease. Some context elements are discrete and others are continuous.
Case 3   : The user is a doctor who treats diabetes.
Case 4   : The user is a doctor who treats heart disease.

The time elapsed from the context element monitor to the selection of the appropriate service version is represented by Time A. In all four cases, it is easy to notice that Time B > Time A; iPOJO needs a certain time between the selection of the new service and the start of this service. This time is always greater than 0. It is therefore possible to conclude that the number of services and their sizes do not affect Time A. Therefore, it can be concluded that Time-adapt is larger than the other periods. Figure 10 shows that Time-adapt follows the same path as Time B; it should be noted that these two periods of time are identical. This observation can be explained considering the expressions of the two periods below:

$$\text{Time\_adapt [Si (T)]} = \text{Time A [Si (T)]} + \text{Time B [Si (T)]}; \quad \text{Time\_adapt [Si (T)]} - \text{Time B [Si (T)]} = \text{Time A [Si (T)]} \approx 0.$$
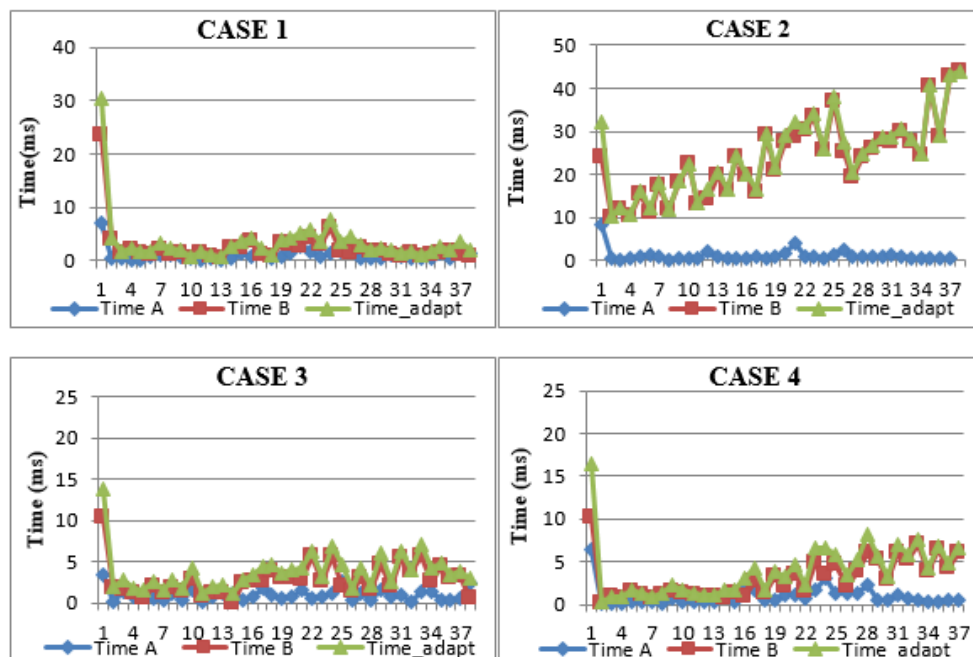

Figure 10. The average of Time A, B, and Time-adapt

The difference between Time B and Time-adapt is almost zero, which implies that the time required to adapt the services depends on Time B. The averages of Time-adapts with α and without α are presented in Table 2, it is clear that the time required for the creation of the matrix MC does not affect the adaptation time.

The second experiment allows achieving three objectives. The objective first is to estimate the impact of the different service versions, the second is to assess the impact of the number of services, and the third one is to evaluate the impact of the nature of the contextual elements. These elements are used by the appropriate service version, such as the vital sign. The value of β, which represents the difference between the times for starting the different service versions, in all four cases, was also calculated. The characteristics for each case are summarized in *Et al* refer Table 3; this makes it possible to determine the impact of the number of services and the types of context elements. For example, Case 1 uses a certain number of service versions to obtain the best application. In this case, a set of service versions is noted {S1 (Display), S1 (Analysis), S1 (Connection)}. The results of this section are presented in Figure 11. It should also be noted that the values of β for S (Display) and S (Analysis), in cases 1 and 2, are very different. S (Display) and S (Analysis) are started with a time lag equal to 1.83 ms and 25.56 ms for cases 1 and 2, respectively. The time elapsed between the start of S (display) and that of S (analysis) was examined and is presented in Figure 11. For a clearer view of the results, the characteristics of Case 1 and Case 2 were compared. To do this, the size of the services and the type of context elements were taken into account.

In Figure 11(a), it is important to note the obvious impact of the size of service versions and the type of context elements. Since services use the continuous type of context elements, the value of β is longer than that of services that use the discrete type of context elements. This value increases steadily from 1.83 ms for case 1 and from 25.56 ms for case 2, respectively. It is also noted that the difference between the sizes of service versions in case 1 is smaller than that in case 2. This has resulted in a longer time to start services.

The time elapsed between the start of S (analysis) and S (connection) is shown in Figure 11(b) for all four cases. The sizes of services are rather similar. The averages are 0.76 ms, 0.11 ms and 0.17 ms for case 1, case 3 and case 4, respectively. However, the average of β in case 2 is slightly longer than in the other cases. Therefore, it can be deduced that the size of service versions has no effect on the start of services.

These results lead to the conclusion that the number and size of services have no influence on the adaptation time. On the other hand, the type of context element, when it is continuous, does have an impact on this time. This type gives a set of values that are called samples. The number of samples has a significant influence on the total adaptation time. As can be seen in Figure 12, the average of β with 750 samples is 8,098 ms; it is equal to 6,731 ms for S (Display) -S (Analysis) and S (Analysis) -S (Connection), respectively. For 1500 samples, the average of β is equal to 25.56 ms; it is equal to 10.4 ms for S (Display) -S (Analysis) and S (Analysis) -S (Connect), respectively. Therefore, it can be deduced that the average of β increases with the increase in the number of samples. This result makes it possible to conclude that the adaptation time is influenced by the sample size of the context elements, when this type is continuous. These elements constitute an easy way to optimize our infrastructure by simply setting the appropriate number of samples that are required by the services to get the best results.

Table 2. The average of Time-adapt with α, and without α

	Case 1	Case 2	Case 3	Case 4
The average of Time_adapt with α (ms)	3.417	24.831	4.103	3.922
The average of Time_adapt without α (ms)	3.344	24.782	4	3.891
The coefficient of the variation (%)	1.53	0.14	1.79	0.56

Table 3. *Et al*. The cases characters

	Case 1	Case 2	Case 3	Case 4
Services versions set	S (display) S (analyze) S(connection)	S (display) S (analyze) S(connection)	S (display) S(connection)	S (display) S(connection)
Context elements	CE1 CE3 CE4 CE5 CE6	CE2 CE3 CE4 CE5 CE6	CE5 CE6	CE5 CE6
Services versions Size (KB)	14 05 13	54 15 13	29 15	29 15
Type of CE	Five CE → discrete	Four CE → discrete One CE → continuous	Two CE → discrete	Two CE → discrete

(a)   (b)

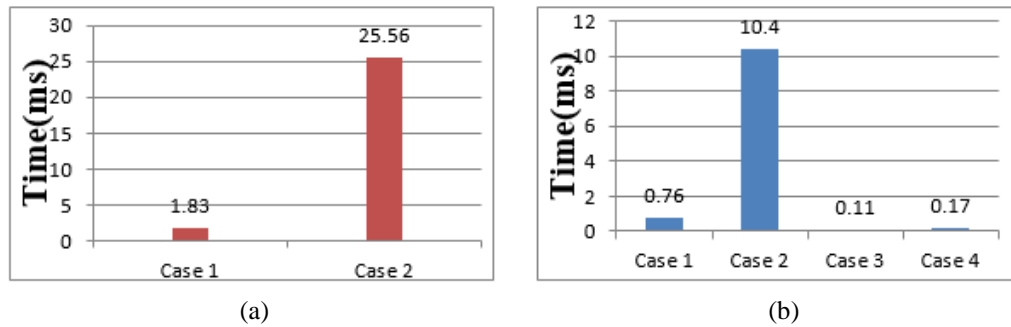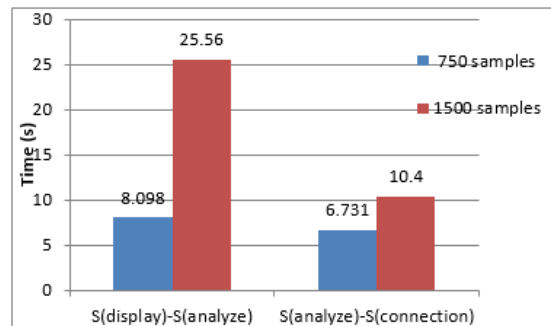Figure 11. The average of β, (a) S(display)-S(analyze), (b) S(analyze)-S(connection)

Figure 12. The average of β for the different sample number

The CAHS and CATS results were compared [23]. Time A was considered for the implementation of the CATS framework, as this time is suitable for the study of the CAHS framework. It is easy to deduce that the adaptation time of CAHS varies between 3 ms and 25 ms in the four cases; however, for CATS, it varies between 60 ms and 100 ms for three cases. Our framework has achieved the best performance and accurate results have been achieved in record time.

6.   CONCLUSION

In this work, a CAHS framework has been proposed to support mobile services dedicated to the health field, such as the monitoring of diabetic and cardiac patients. The proposed model involves a comprehensive framework for managing the behavior of mobile services at runtime, based on contextual elements. Based on Service Oriented Architecture (SOA) principles, the CAHS framework has provided a good runtime environment for health-oriented services. Our framework has the capacity to monitor more than one disease. An adaptation strategy to manage the service versions (start, stop, install), based on context criteria, has been proposed. Note that the CAHS framework starts the appropriate service according to the current situation, without the intervention of the user. In this article, it was possible to evaluate the time needed from the start of evaluation of the contextual elements until the installation of an adapted service. A series of components to assist the manager's framework in the process of context monitoring and service adaptation were introduced. In the first step, the component worked with monitors of context elements that regularly update the value of a given element. When the context elements are monitored, the relationship between the context elements and the service is provided by an MC matrix that was created based on the current context conditions. An adequate service was identified by the Manifest.mf file which contains the name of the version of the service, such as S1 (display). Finally, a default service was used when the correlations between services and context elements were not verified; this would avoid stopping the framework.

The effectiveness of our framework has been proven from different angles and under different scenarios. The present work aimed at improving healthcare and facilitating the development of mobile applications dedicated to the healthcare sector. This may be possible by dividing these applications into various features that are managed by our framework. A robust implementation was presented for each component of our framework. The implementation of our framework has demonstrated the great benefits that may result from monitoring various diseases.

The status of the service (stop and start) is influenced by the type of context elements. Although the adaptation time may vary, it is still reasonably fast. Note that in order to better optimize the performance of our framework, it is highly recommended to identify the number of values of the continuous type of context elements. These values are used by services to provide the best outcome regarding the patient's situation. It should be mentioned that issues related to security and privacy were ignored. These issues are studied by authors in [24]. In addition, the case of monitoring the patient who suffers from several diseases was not considered. It is important to mention that the effects of certain diseases on others have not been taken into account. Thus, as part of our future work, it would be desirable to include these different issues in the CAHS framework.

REFERENCES

[1] Vital Wave Consulting, "mHealth for Development: The Opportunity of Mobile Technology for Healthcare in the Developing World," Washington, D.C. and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership, 2009.

[2] B. M. C. Silva, *et al*., "Mobile-health: A review of current state in 2015," *Journal of Biomedical Informatics*, pp. 265-272, 2015.

[3] F. Abdur, *et al*., "CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living," *Future Generation Computer Systems*, vol. 35, pp. 114-127, 2014.

[4] H. K. Pung, *et al*., "Context-aware Middleware for Pervasive Elderly Homecare," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 510-524, 2009.

[5] A. R. M. Forkan, *et al*., "BDCaM: Big Data for Context-aware Monitoring—A Personalized Knowledge Discovery Framework for Assisted Healthcare," *IEEE Transactions on Cloud Computing*, vol. 99, 2015.

[6] B. Yuan and J. Herbert, "Context-aware hybrid reasoning framework for pervasive healthcare," *Pers Ubiquit Comput,* vol. 18, pp. 865-881, 2014.

[7] F. Ongenae, *et al*., "A probabilistic ontology-based platform for self-learning context-aware healthcare applications," *Expert Systems with Applications,* vol. 40, pp. 7629-7646, 2013.

[8] C. C. Lo, *et al*., "Ubiquitous Healthcare Service System with Contextawareness Capability: Design and Implementation," *Expert Systems with Applications*, vol. 38, pp. 4416-4436, 2011.

[9] M. Baldauf, *et al*., "A survey on context-aware systems," *Int. J. Ad Hoc and Ubiquitous Computing*, vol. 2, pp. 263-277, 2007.

[10] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, pp. 22-32, 1994.

[11] P. J. Brown, *et al*., "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications,* vol. 4, pp. 58-64, 1997.

[12] N. Ryan, *et al*., "Enhanced reality fieldwork: the context-aware archaeological assistant," *Computer Applications in Archaeology*, vol. 4, pp. 269-274, 1999.

[13] M. Miraoui, *et al*., "Dynamic Context-Aware Service Adaptation in a Pervasive Computing System," *Proceedings of the third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 77-82, 2009.

[14] G. D. Abowd, *et al*., "Towards a Better Understanding of Context and Context-Awareness," *Proceedings of the International Symposium on Handheld and Ubiquitous Computing*, pp. 304-307, 2001.

[15] S. Belhadi and R. Merzougui, "A Framework Survey for Context Aware Adaptation," *International Journal of Computer Science and Information Security*, vol. 14, pp. 732-738, 2016.

[16] D. Popovici, *et al*., "A framework for mobile and context-aware applications applied to vehicular social networks," *Social Network Analysis and Mining*, vol. 3, pp. 329-340, 2013.

[17] R. Mizouni, *et al*., "A framework for context-aware self-adaptive mobile applications SPL," *Expert Systems with Applications*, vol. 41, pp. 7549-7564, 2014.

[18] F. G. Marinho, *et al*., "MobiLine: A Nested Software Product Line for the domain of mobile and context-aware applications," *Science of Computer Programming*, vol. 78, pp. 2381-2398, 2013.

[19] B. Yazid, "Context discovery for automatic adaptation of services in ambient intelligence," Thesis of PhD, Department of comuting for Open and Decentralized Cooperative Systems, Henri Fayol Institute, 2011.

[20] M. Aksit and Z. Choukair, "Dynamic, Adaptive and Reconfigurable Systems Overview and Prospective Vision," *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, pp. 84-92, 2003.

[21] M. Yarvis, *et al*., "Conductor: A framework for distributed adaptation," *Proceedings of the Seventh Workshop on Hot Topics in Operating Systems*, pp. 44-49, 1999.

[22] C. Escoffier, *et al*., "iPOJO: an Extensible Service-Oriented Component Framework," *Proceedings of International Conference on Services Computing,* IEEE, Salt Lake City, USA, pp. 474-481.

[23] P. Dana, "Context management for mobile applications dedicated to transport," University of Valenciennes and Hainaut-Cambresi, French, 2012.

[24] K. Prableen, *et al*., "Big Data and Machine Learning Based Secure Healthcare Framework," *Procedia Computer Science*, vol. 132, pp. 1049-1059, 2018.