# Smart Emergency Alert System Using Internet of Things and Linked Open Data for Chronic Disease Patients

Hajar Khallouki, Mohamed Bahaj, Philippe Roose

## ▶ To cite this version:

**HAL Id: hal-02465085**

**https://hal-univ-pau.archives-ouvertes.fr/hal-02465085**

Submitted on 5 Feb 2020

# Smart Emergency Alert System using Internet of Things and Linked Open Data for Chronic Disease Patients

Hajar Khallouki[1*], Mohamed Bahaj[1] and Philippe Roose[2]

[1] Mathematics and computer science department
Hassan I University, Faculty of Sciences and Techniques
Settat, Morocco
[2] LIUPPA – T2I, 2 Allée du Parc Montaury
64600 Anglet, France
*Hajar.khallouki@gmail.com

**Abstract.** Nowadays, the widespread deployment of more powerful devices (sensors, smartphones, tablets, etc.) has provided us with great number sources of sensing data that are exploited in several domains namely the healthcare domain. Chronic diseases are the most common causes of death and disability worldwide. These types of diseases require more and more studies to help patients and notify cases of crises that lead to death. Representing knowledge through building an ontology for emergency alert system is important to achieve semantic interoperability among health information, predict the patient real-time context and to better execute decision notification. Linked Open Data services are used in our paper in order to provide with the semantic description of collected data from different sources (wearable sensors, environmental sensors, etc.).

**Keywords:** devices, sensing data, chronic diseases, ontology, emergency alert, real-time context, Linked Open Data.

## 1 Introduction

Since the last decade, the amount of data, the one on the Web as well as the one provided by Sensor Networks, is increasingly growing. Sensor Networks are the main parts of the Internet of Things (IoT). The term "Internet of things" was initially proposed by Kevin Ashton [1] in 1999 to describe things which are equipped with radio frequency identification chips (RFID chips). However, the concept has been developed during the last years and generalized towards an approach connecting a very large number of things to the internet, enabling them to provide services and collect context information independently.

The context awareness is a key element for creating smart applications. The collected contextual data is generally raw information from different distributed sources, which needs to be interpreted. Based on ontologies, it is possible to build semantic models which will be fed by this raw data and thus not only to increase their level of semantic representation but especially of being able to use them to make automatic decisions of adaptation of applications based on the context to the runtime.

The idea is to profit from the context-awareness functions to enable the chronically ill to live a normal life and to benefit from flexible and automated medical surveillance. In fact, several cases appear among which we can mention the patients with chronic kidney disease who suffer from an irreversible decrease in glomerular filtration rate, the best indicator of renal function.

The objectives of our work include:

- To model a smart system that enables predicting the patient's real-time context, detecting their critical situation and send an emergency alert when needed.
- A real-time data modeling and knowledge representation through ontology with examples of inference rules.
- A mechanism for collecting raw data from different sources.
- A smart emergency mechanism that locates the patient in emergency conditions along with the closest hospital then sends an emergency notification.

In the rest of this paper, we first introduce some related work on smart healthcare paradigm and existing approaches. Next, we present our proposed system architecture and its components. Finally, we conclude our paper.

## 2  Related work

Over the last decade, interest in well-being has increased among the population. Several approaches have been proposed in order to improve the healthcare systems. Solanas A. et al. [1] introduced the new concept of smart health, which is the context-aware complement of mobile health within smart cities. They provided an overview of the main fields of knowledge that are involved in the process of building this new concept. Moreover, they presented the main challenges and opportunities that s-Health would imply and provide a common ground for further research.

In [2] Liu M. et al. Proposed the concept of IoT-enabled intelligent assembly system for mechanical products (IIASMP). They described the IIASMP framework, which is based on advanced techniques such as information and communication technology, sensor network, and radio-frequency identification.

Venkatesh J. et al. [3] proposed an approach which integrates smart health applications with the connectivity of the Internet of Things (IoT), facilitating the use of machine learning and improving the complexity of context-aware designs. Their proposed approach allows designing more complex health applications with limited resources, using sensors both on the user and in the surrounding spaces.

In their work [4] Zamanifar A. et al. introduced a novel hierarchical scalable network scheme, DSHMP-IOT, for mobility management of IP-based sensors, which get the benefit of predicting the movement direction of mobile nodes in health-care applications to reduce the hand-off cost. Venkatraman S. et al. [5] developed user ontology for a Smart Health Information Portal (SHIP) to provide collaborative health terms in holistic medicine. The proposed approach integrates conventional medical terms related to cardiac conditions along with Homeopathy and Ayurvedic terms

towards providing meaningful complementary heart health information within an existing SHIP.

In [6] Azimi I. et al. started with studying the state-of-the-art IoT-based elderly monitoring approaches to investigate their advantages and shortcomings from a different viewpoint by considering the elderly requirements at the center of attention. Thus, they presented a modernized classification and proposed a hierarchical model for elderly-centered monitoring to investigate the current approaches, objectives and challenges in a top-down fashion.

Muhammad G. et al. proposed in [7] a solution for voice pathology monitoring of people using IoT–cloud. More specifically, a voice pathology detection system was proposed inside the monitoring framework using a local binary pattern on a Mel-spectrum representation of the voice signal, and an extreme learning machine classifier to detect the pathology. The proposed monitoring framework can achieve high accuracy of detection.

An ontology model is a framework for describing domain knowledge, considering relevant concepts and their connections [8].Kim J. et al. presented in [9] an ontology-driven interactive healthcare (OdIH_WS) to apply real-time information acquired from wearable sensors to services.

In [10]Ko E. J. et al. proposed context modeling and context reasoning in a context-aware framework which is executed on an embedded wearable system in a ubiquitous computing environment for U-Healthcare.

## 3   Proposed system architecture

Our main objective is to design an innovative architecture able of predicting the patient's real-time context, detecting their critical health situation and send an emergency alert when needed. Our architecture is composed of three main components: 1) the context aware manager, 2) interaction manager and 3) emergency manager
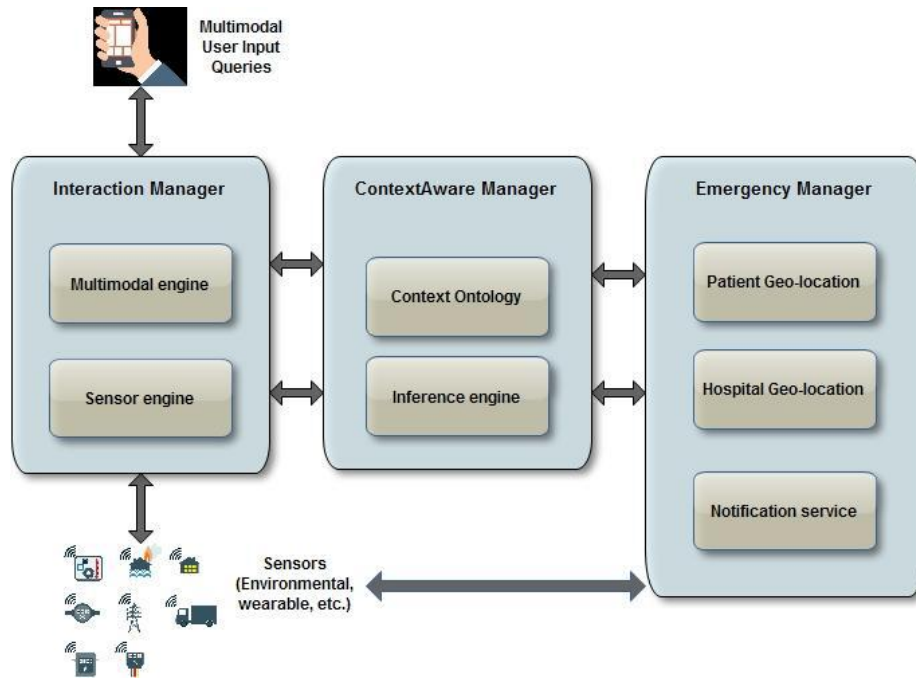
**Fig.1.** System architecture

The sensing devices measure information using a skin temperature sensor, a respiration sensor, a geolocation sensor, etc. The sensing devices sense the raw data of the patient and his context and then transfer the raw data to the sensor engine. The sensor engine receives raw data from the sensing devices then matches them to the context ontology.

A patient can interact with the system using different modalities (e.g. speech); these modalities will be captured by the multimodal engine and used in the context ontology.

To explore the potential capabilities of context-awareness, we opt for ontologies. The context ontology models the physical and environmental context of the patient; it classifies entities into lined graph that captures relationships between them. The context ontology is fed by the data provided by the interaction manager component.

The Inference Engine infers the patient context ontology, on the basis of context information collected through the interaction manager, then generates an emergency status if needed.

The last step in our architecture is the emergency notification. When an emergency status is declared as true by the inference engine, the emergency manager will first locate the patient and find the nearest hospital, thus, sends an emergency notification along with the patient's location in order to reach them.

### 3.1 Context Aware Manager

In order to utilize the information generated from different sources, context information models are needed, and context information models are defined by using the most effective method among the model representation methods. We propose semantic knowledge representation trough ontology. To apply this model to the chronic disease emergency alerting service, inference rules are required to infer designed Ontology.

*A. Context ontology*

Many knowledge modeling techniques exist, namely graphical modeling, object-oriented modeling and ontology-based modeling. The ontology-based modeling is very powerful and applicable in the context aware solutions. It has the advantage of sharing knowledge, logic inference and the reuse of knowledge.

The ontology is a specification of a conceptualization that describes the concepts and relationships existing in a domain. The ontology enables applications without the overhead of traditional data integration. Figure 2 illustrates a snapshot of our proposed context ontology for chronic disease patient.
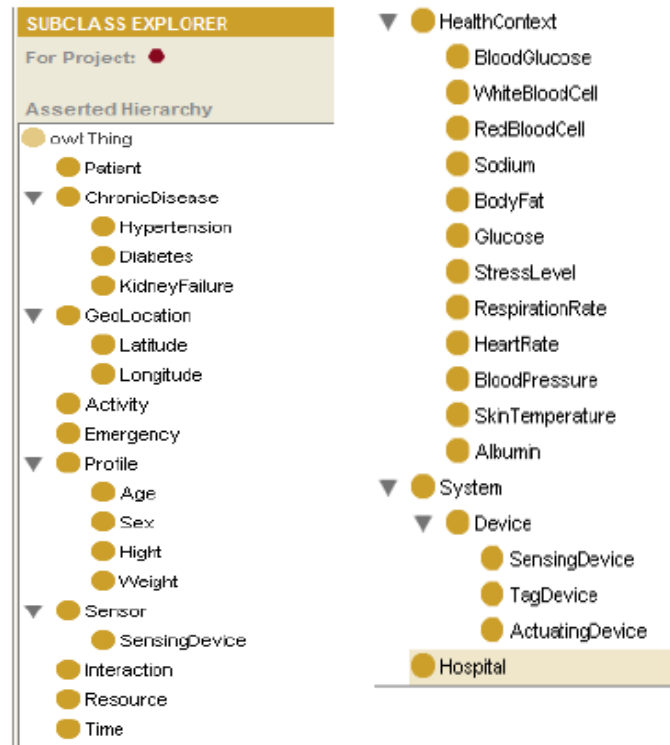


**Fig.2.** A snapshot of the proposed context ontology

In the Patient context ontology, relevant context items include patient health context (i.e. heart and respiration rate values, etc.), the patient's geolocation and activity, etc. These data can be used by the system to automatically infer the patient critical health status and detect possible emergency alert situations.

The *profile* class describes the personal information of the patient including age, sex, weight and height. The *ChronicDisease* class some of chronic diseases a patient may face (e.g. hypertension, diabetes, etc.). The *HealthContext* defines the specific details of the health status of the patients. The *GeoLocation* class specifies the geo-location of the patient; it is defined by the latitude and longitude values. The *System* class represents parts of sensing infrastructure. The *Device* class is a subclass of system, it represents three types of devices; *SensingDevice*, *TagDevice* and *ActuatingDevice*.

To retrieve the data stored in an ontology there are several query languages, the most common is called SPARQL [11].In the following figure, an example of a SPARQL query asking for the patient's geo-location.



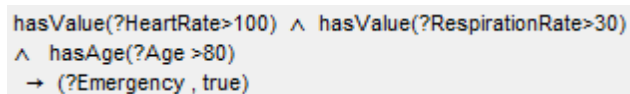**Fig.3.** Geo-location SPARQL query

### B. Inference Engine

Context information necessary for the emergency service is derived by inferring the context ontology. In order to improve the expressiveness of the proposed ontology and enable the definition of if-then statements; we used the rule based formalism *Semantic Web Rule Language* (SWRL) [12]. Figure 4 shows a rule example used in our system.



**Fig.4.** SWRL rule example

This rule indicates if the patient has a heart rate upper to 100, a respiration rate upper to 30 and is age is more than 80years old, then the emergency status takes the value true.

## 3.2 Interaction Manager

The interaction manager is a main component of our architecture; it enables the collection of raw data from different sources. The interaction manager is composed of two modules; multimodal engine and sensor engine.

### A. Multimodal Engine

There are many multimodal services integrated in devices such as; the HandWriting Recognition (HWR), Speech Recognition (SR) and EmotionML for user emotions recognition. The patient input modalities are represented with EMMA language. EMMA is an XML markup language which provides mechanisms for capturing and annotating the various stages of processing of users inputs [13].

The following figure shows an EMMA interpretation of a spoken word "help".

```
<emma:medium="acoustic"
emma:mode="voice"
emma:function="dialog"
emma:verbal="true"
emma:start="1241035886246"
emma:end="1241035889306"
emma:source="smm:platform=iPhone-2.2.1-5H11"
emma:signal="smm:file=audio-416120.amr"
emma:signal-size="4902"
emma:process="smm:type=asr&version=asr_eng2.4"
emma:media-type="audio/amr; rate=8000"
emma:lang="en-FR"
emma:grammar-ref="gram1"
emma:model-ref="model1">
 <emma:interpretation id="int1"
 emma:confidence="0.75"
 emma:tokens="help">
<flt><interaction>help</interaction>
</emma:interpretation> <emma:info>
<session>E50DAE19-79B5-44BA-892D</session>
</emma:info>
<emma:grammar id="gram1" ref="smm:grammar=help"/>
</emma:emma>
```

**Fig.5.** EMMA interpretation example

In order to integrate data provided by the multimodal engine to our knowledge model, we proposed algorithm (figure 6) that aims at mapping the EMMA description which contains the user inputs modalities along with the OWL description which describes the patient context.

| Algorithm : EmmaToOwlMappingData() | |
|---|---|
| **Inputs** | Owl : OWL file |
| | Emma : EMMA file |
| **output** | Owl file |
| 1. | *Ems*= Emma.getAllElements() ; // get all nodesfromemma file |
| 2. | *owls*= Owl.getAllElements() ; // get all nodesfromOWL file |
| | |
| 3. | *if*Ems.containtsLocaltion*then* |
| 4. | loc$_1$=Ems.getLocation; |
| 5. | *if* !owls.isLocationDifined*then* |
| 6. | defineLocationClass(Owl) |
| 7. | *end if* |
| | |
| 8. | loc$_2$=owls.getNodeLocation(); |
| 9. | loc$_2$.setLocation(loc$_1$) |
| 10. | owls.setNodeLocation(loc$_2$) |
| 11. | *en if* |
| | |
| 12. | lng$_1$=Ems.getLang(); |
| 13. | *if* !owls.isLangDifined*then* |
| 14. | defineLangClass(Owls) |
| 15. | *end if* |
| | |
| 16. | lng$_2$=owls.getNodeLang(); |
| 17. | lng$_2$.setLang(lng$_1$) |
| 18. | owls.setNodeLang(lng$_2$) |
| | |
| 19. | updateOWL(owls) |
| 20. | return *owl* |

**Fig.6.** Multimodal algorithm

The algorithm takes as inputs the EMMA annotation and the OWL description; it gets and compares all nodes from both EMMA and OWL descriptions. When similarities get identified, it takes the data from EMMA nodes and fills it into OWL nodes. Finally, the result is returned in the OWL file.

### B.  Sensor Engine

The sensor engine is an important component in our architecture; it receives the raw data from sensing devices and transforms them to semantic descriptions.

Sensor data needs to be shared, processed and interpreted. Delivering sensor data as Linked Data enables discovery, access, query and interpretation of it.

The sensed and observed data are encoded through JSON annotation, thus, transformed to semantic descriptions through ThingSpeak[1] platform. The semantic annotations become semantic descriptions encoded through the JSON-LD notation. The following example describes a sensor of geo-location.

```
{
"@id":"gps_shield",
"@type": "KDNDisease: sensor",
"observes": {
"@id":"geo-location",
"@type":"KDNDisease: geo-location"
          }
}
```

**Fig.7.** Geo-location sensor description

```
{
"@id":"geo-location",
"@type": "KDNDisease: geo-location",
"observedData":{
"latitude":{
"@id":" KDNDisease: latitude",
"@type":"xsd: double",
"@value":"33.030908"
        }
"longitude":{
"@id":" KDNDisease: longitude",
"@type":"xsd: double",
"@value":"-7.616505"
        }
                        }
}
```

**Fig.8.** Geo-location observation description

In order to collect raw data from sensors, process them and fill them to our proposed ontology, we proposed the following algorithm (figure 9).

| Algorithm JASONLD_to_OWL() | |
|---|---|
| **input** | jld: JSON-LD file, owl : owl file |
| **output** | owl : final owl file |

```
1.    sensor=jld.getSensor();  // get sensed data from JSON-LD
2.    lat=jld.getLatitude();
      long=jld.getLongitude();

      // testing if json-ld contains values (data) collected from sensors
3.    if   lat != "" and long != "" and sensor != "" then
4.        owl.setSensor(sensor); // fill sensed values into Owl
5.        owl.setLatitude(lat);
6.        owl.setLongitude(long);
7.    end if

8.    return owl;
```

**Fig.9.** Sensor algorithm

The algorithm takes as inputs the JSON-LD and the OWL descriptions. It starts by getting data from JSON-LD (e.g. sensor name, longitude and latitude values, etc.). Thus, checks if JSON-LD description contains the required sensed values, if yes, the OWL description will be filled with the collected values.

### 3.3 Emergency Manager

Once a fall is detected, an alert should be sent to the nearest hospital containing the location of the patient. The emergency manager is composed of three modules:1) the patient geo-location which specifies the patient real-time location, 2) the hospital geo-location in order to find the nearest hospital to the patient location and 3) the notification service for emergency alert. The following algorithm describes the process of the emergency alert.

```
Algorithm : EmergencyAlert()
inputs        em: emergency manager
              owl: ontology
  1.          emergency=false;
  2.          geoP=null;
  3.          hos=null;

  4.          emergency=em.getEmergencyStatus();
              //gets the emergency status from the emergency manager

  5.          if emergency=true then
  6.          geoP=owl.getPatientGeolocation();
              //gets the geolocation of the patient
  7.            if geoP!=null then
  8.               hos=owl.getNearbyHospital(geoP);
              //gets the closest hospital to the patient
  9.                  if hos!=null then
 10.                  em.sendAlertToHospital(hos,geoP);
              //sends an alert to the closest hospital along with the geolocation
              of the patient
 11.                  end if
 12.            end if
 13.          end if
```

**Fig.10.** Emergency alert algorithm

The algorithm takes as inputs the emergency manager and the ontology description. It starts by getting the emergency status from the emergency manager. If the returned status is true, it gets the patient geo-location and check the closest hospital to the patient location through *getNearbyHospital()* service. Thus, an emergency alert has to be sent along with the location of the patient in order to reach them.

## 4   Conclusion

In this paper, we proposed a knowledge representation through an ontology for emergency notification system. Our system allows the prediction of the patient real-time context, analyzing and processing their health conditions to better execute decision notification.

There remains more work to be carried out for covering all the aspects of our proposed approach and for improving our emergency alert system by integrating cloud computing for service continuity. Another work aims to build a context-based agent architecture in which the various components are implemented as autonomous agents to make our proposed framework more generic and distributed.

# References

1. Solanas, A., Patsakis, C., Conti, M., Vlachos, I. S., Ramos, V., Falcone, F. & Martinez-Balleste, A. (2014). Smart health: a context-aware health paradigm within smart cities. IEEE Communications Magazine, 52(8), 74-81.
2. Liu, M., Ma, J., Lin, L., Ge, M., Wang, Q., & Liu, C. (2017). Intelligent assembly system for mechanical products and key technology based on internet of things. Journal of Intelligent Manufacturing, 28(2), 271-299.
3. Venkatesh, J., Aksanli, B., Chan, C. S., Akyurek, A. S., & Rosing, T. S. (2017). Modular and Personalized Smart Health Application Design in a Smart City Environment. IEEE Internet of Things Journal.
4. Zamanifar A., Nazemi, E., & Vahidi-Asl, M. (2017). DSHMP-IOT: A distributed self healing movement prediction scheme for internet of things applications. Applied Intelligence, 46(3), 569-589.
5. Venkatraman S. (2017). User-centric Ontology for Smart Holistic Health Information Systems. International Journal Series in Multidisciplinary Research (IJSMR)(ISSN: 2455-2461), 23-37.
6. Azimi I., Rahmani, A. M., Liljeberg, P., & Tenhunen, H. (2017). Internet of things for remote elderly monitoring: a study from user-centered perspective. Journal of Ambient Intelligence and Humanized Computing, 8(2), 273-289.
7. Muhammad G., Rahman, S. M. M., Alelaiwi, A., & Alamri, A. (2017). Smart health solution integrating IoT and cloud: a case study of voice pathology monitoring. IEEE Communications Magazine, 55(1), 69-73.
8. Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies, 43(5), 907–928.
9. Kim, J., Kim, J., Lee, D., & Chung, K. Y. (2014). Ontology driven interactive healthcare with wearable sensors. Multimedia Tools and Applications, 71(2), 827-841.
10. Ko, E. J., Lee, H. J., & Lee, J. W. (2007). Ontology-based context modeling and reasoning for u-healthcare. IEICE TRANSACTIONS on Information and Systems, 90(8), 1262-1270.
11. W3C, "SPARQL 1.1 Query Language," W3C, 2013. [Online].Available:https://www.w3.org/TR/2013/REC-sparql11-query-20130321/. [Accessed: 19-Oct-2016].
12. Horrocks I., Patel-schneider P. F., Boley H., Tabet S., Grosof B., and Dean M., "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Memb. Submiss. 21, no. May 2004, pp. 1–20, 2004.
13. Johnston, M. (2009, November). Building multimodal applications with EMMA. In Proceedings of the 2009 international conference on Multimodal interfaces (pp. 47-54). ACM.