



An IoT Architecture to Provide Hybrid Context Reasoning

Roger Machado¹(✉), Ricardo Almeida¹, Rogério Albandes²,
Ana Marilza Pernas¹, and Adenauer Yamin^{1,2}

¹ Federal University of Pelotas, Pelotas, Brazil

{`rdsmachado,rbalmeida,marilza,adenauer`}@inf.ufpel.edu.br

² Catholic University of Pelotas, Pelotas, Brazil

`rogerio.albendes@sou.ucpel.edu.br`

Abstract. Considering the dynamic nature of the modern computational infrastructures provided by IoT, applications need to be aware of the contextual data that interest them, to be able to operate with as little human intervention as possible. Thus, context awareness becomes a key concept to provide adaptive services in IoT environments. Context reasoning is one of the more critical steps to obtain context awareness. However, a context reasoning strategy that can be applied satisfactorily in different application domains has not yet been found. Because of this, hybrid strategies for context reasoning are gaining prominence. In the literature, some researchers explore hybrid proposals, but these proposals do not offer flexibility on the use of the reasoning strategies. In this research, we conceive hybrid reasoning based on compositional rules, enabling a dynamic composition of different strategies. Thus, the context-aware applications can choose among different reasoning strategies, those that are most appropriate depending on the contexts that will be treated. To validate our architecture, we design and test it on a scenario based on healthcare. The obtained results showed that our architecture allows the utilization of hybrid strategies for context reasoning, improving situations identification, and decision-making.

Keywords: Internet of Things · Context awareness · Hybrid context reasoning

1 Introduction

In the last twenty years, technology has advanced considerably, with the proliferation of connected devices of different computational capacities and on a

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS) - PqG. Roger da Silva Machado is a FAPERGS/CAPES - BRAZIL Scholarship holder.

© IFIP International Federation for Information Processing 2020

Published by Springer Nature Switzerland AG 2020

A. Casaca et al. (Eds.): IFIP IoT 2019, IFIP AICT 574, pp. 86–102, 2020.

https://doi.org/10.1007/978-3-030-43605-6_6

scale that will reach a total of 50 billion connected devices in 2020 [5]. IoT systems have become an important and even essential part of our daily lives. Smart homes are good examples of where such systems can be found. However, the development of IoT systems is a difficult task because it involves multiple areas of computing, such as software engineering, artificial intelligence, and distributed systems [21].

Due to the fast growth of IoT computing, an increasing number of contextual data is continuously generated from different sources, formats, or semantics which is needed to be evaluated together to identify situations of interest to context-aware applications [12].

In this aspect, context awareness becomes an essential approach for providing adaptive and autonomous services. It can be used, for example, to select the most appropriate services according to the context information that is relevant to the user or to change operational parameters of services in execution. Thus, context-aware applications must be able to adapt their changing behaviors reducing human intervention, introducing to this, several challenges to developers [10].

In order to build and execute context-aware applications, some features need to be provided, ranging from the acquisition of contextual information from heterogeneous and distributed sources, representation of this information, storage, processing, and reasoning for their use on decision-making [2].

Given the importance of the context reasoning to obtain the context awareness, diverse strategies for that have recently been proposed. However, they are not versatile enough to individually meet the reasoning requirements for different IoT applications. Thus, the need for a combination of different methods for context reasoning is identified by means the hybrid strategies [3, 12].

Although several proposals on hybrid context reasoning are innovative, they are not able to deal with the dynamic needs from actual IoT demands. Thus, the goal of this work is the conception of an IoT software architecture, providing an approach to hybrid context reasoning. For this, we propose a compositional approach for context reasoning that enables the composition of different strategies. Consequently, the context-aware applications can choose among different reasoning strategy, and if they will be used individually or in combination, considering the contexts that will be treated. Considering this, we provide flexibility for the application to customize the context awareness provided, according to their demands.

The proposed compositional approach for context reasoning is an extension of the Context Recognition and Adaptation Subsystem of the EXEHDA middleware [14]. EXEHDA consists of a situation-aware middleware to IoT, which aims to create and manage a widely distributed computing system environment, as well as to promote the implementation of applications on it.

The remainder of this paper is organized as follows. Section 2 reviews some related works. We describe the compositional reasoning proposed in Sect. 3. In the fourth Section, we discuss the evaluation of the proposal, exploring a use case with the use of a dynamic strategy for context reasoning. Finally, the Fifth Section presents our conclusions and future work.

2 Related Work

In the literature, several papers explore hybrid strategies for context reasoning in IoT. To identify these papers, we performed a Systematic Literature Review (SLR) analyzing papers published between 2012 and 2017, more details can be found in [16]. SLR is a research methodology that foresees the execution of some procedures to generate a literature review in an area, to identify a set of papers that bring a precise prospection of the addressed state of the art [11]. With the execution of the SLR, we selected seven papers. In the following, we present aspects regarding reasoning strategy, as well as main functionalities about the seven works.

The paper [15] proposes a software architecture for context awareness called DynamiCC. The proposal focuses on the conception of a context model and a software architecture that allow the interpretation and the dynamic composition of the information acquired by sensors positioned in different locations of the ubiquitous environment. A hybrid strategy for reasoning was proposed to perform the processing of contextual information. DynamiCC uses semantic reasoning provided by the ontology, through axioms and rules to infer new knowledge based on the ontology instances. In addition to this, DynamiCC uses reasoning based on rules such as ECA (Event-Condition-Action), which performs the processing of contextual information provided by a relational model.

In [24] is proposed a software architecture called CASP (Context-Aware Service Platform). This platform has as main characteristics to be context-aware and deal with the aggregation and abstraction of context information using ontologies to represent them. CASP platform uses two reasoning strategies in the context processing stage, one based on ontological rules and the other in cases. The ontological rules represent general knowledge of the domain, while the cases capture specific knowledge. Ontological reasoning evaluates the rules provided and adds the inferred information to the knowledge base. Case-based reasoning recovers similar situations and uses the corresponding solutions to update the data in the knowledge base represented by the ontological model.

The paper [25] presents the architecture of a framework for hybrid reasoning developed for the CARA (Context-Aware Real-time Assistant) system. The authors use hybrid reasoning based on cases and fuzzy rules, making it more robust and adaptable to an environment subject to change. Case-based is used to detect conditional anomalies for residential automation, and the fuzzy rules to deal with exceptions, as well as case retrieval and adaptation of query-sensitive cases.

In [23] is proposed the software conception of a group recommendation system for concerts for groups of users. The prototype is context-aware and takes into account the user's location and timing when giving recommendations. In order to perform the reasoning of contextual data, the authors propose a hybrid strategy that implements two algorithms, which take advantage of the historical information of the users: a collaborative lineage algorithm (K-Nearest Neighbor) and a matrix-factorization algorithm. The algorithms can be used individually or in combination.

The paper [1] proposes software that individually classifies the retail products on a shop shelf. The authors propose two different hybrid strategies. The first combines SVMs (Support Vector Machines) with HMMs (Hidden Markov Models). In the second, the SVMs are combined with CRF (Conditional Random Fields) aiming to form a new context. Probabilistic models are trained by learning from context-free classifier errors (SVMs) and neighboring relationships between retail products.

In [19] is proposed a framework for a hybrid strategy for context reasoning for the Mining Minds, which offers personalized support for health and well-being. In order to perform the context reasoning, a new conjunctive approach is proposed, based on ontology and machine learning. The machine learning strategy supports the inference by classifying high-level contexts based on the data set instantiated in the ontology.

The paper [13] proposes a framework that uses a hybrid strategy for context reasoning to middleware SWARMs (Smart and Networking Underwater Robots in Cooperation Meshes). The approach consists of three reasoning: (i) ontology axioms to infer knowledge related to the classes contained in ontology; (ii) ontology rule that process the contextual data contained in the ontology, using SWRL (Semantic Web Rule Language) for the design of the rules; and (iii) Multi-Entity Bayesian Network, which consists of a logical system that integrates first-order logic with Bayesian probability theory.

Table 1 presents a comparative analysis of the seven papers selected. This analysis was based on the steps followed to obtain context awareness [12]: (i) types of sensors used in the acquisition step; (ii) models used to perform the context representation; (iii) model used to store the contextual data; and (iv) strategies to implement the context reasoning.

Based on Table 1, we can observe that, in the acquisition step, only two papers do not present how they collect contextual data. The remaining works have support to deal with different types of sensors, allowing the treatment of different applications.

Regarding the modeling step, we can see that four papers use the ontology-based model, mainly due to its semantic representation characteristic. It is worth mentioning that [15] proposed the use of hybrid modeling, combining ontological and relational models.

Issues related to contextual information storage are not discussed by the majority of the papers, although it is essential for context-aware applications. The work of [25] uses the markup scheme model XML (eXtensible Markup Language), mainly because it can be utilized to transmit information between different components. The work of [15] uses a relational model of storage, which is not considered a satisfactory model for the storage of data provided by the ontologies [4]. Besides, the work of [19] proposes the use of the triple model along with Jena TDB, being a more indicated model to perform the persistence of information provided by ontologies.

We believe that mechanisms with different characteristics are required for context storing since the current context-aware applications tend to deal with

Table 1. The comparison performed among related work.

Paper	Acquisition	Representation model	Storage model	Reasoning strategy
[15]	Sensor: Physical and Logical	Ontological and Relational	Ontological Rules and Rules	Relational
[24]	Sensor: Physical and Logical	Ontological	Information Not Available	Case-based and Ontological Rules
[25]	Sensor: Physical and Logical	Fuzzy	XML	Case-based and Fuzzy Rules
[23]	Information Not Available	Information Not Available	Information Not Available	Machine Learning (K-Nearest Neighbor and Matrix Factorization)
[1]	Information Not Available	Information Not Available	Information Not Available	Machine Learning and Probabilistic Logic
[19]	Sensor: Physical and Logical	Ontological	Triple	Machine Learning and Ontological Axioms
[13]	Sensor: Physical and Logical	Ontological	Information Not Available	Ontological Axioms, Ontological Rules, and Probabilistic Logic

diverse types of data, semantic levels, among other aspects. In turn, the efficiency of these mechanisms is a significant aspect, considering the scalability of the volume of contextual data involved, and its constant updating or search.

Regarding context reasoning, despite using hybrid strategies, the solutions do not allow the application to choose a strategy that better fits their needs. Besides, we note the absence of an approach that enables the combination of the different strategies for context reasoning. In this research, we propose the creation of compositional reasoning that could increase the flexibility for the use of contextual data, facilitating the identification of interest situations. Moreover, with the development of this strategy, applications can choose whether these different reasoning strategies will be utilized individually or combined, depending on application demand and the contextual data treated.

3 Software Architecture Proposed

The differential of our proposal is to allow the applications to choose among the strategies offered to context reasoning. Thus, providing greater flexibility for the application to customize the context awareness provided, according to their demands. Figure 1 gives an overview of the proposal, where we can see the software components necessary to offer a dynamic strategy for context reasoning. The main features of these components are discussed below, highlighting the Reasoning Component focused on this paper.

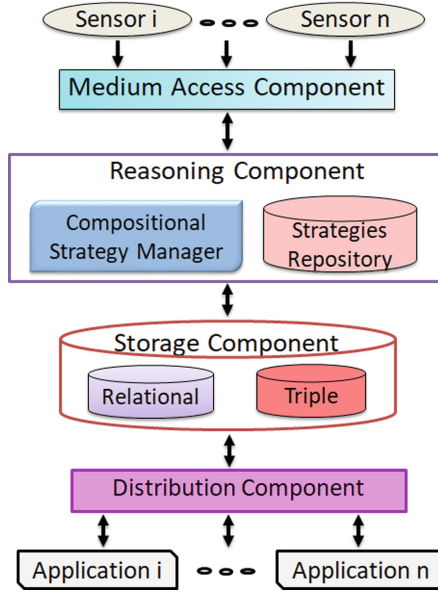


Fig. 1. Overview of proposal

In Fig. 1, Medium Access Component is responsible for sending and receiving information from the applications. It performs the context acquisition from distinct types of sensors, such as events about the use of operating system resources and log files internal to the system, and can receive contextual data from different devices. Moreover, it performs the configured actions based on the identified situations. These actions can be alerts by email, instant messaging service, or commands execution, which results in the nonfunctional adaptation of the environment at runtime.

Storage Component is formed by two storage models offering: (i) a relational model, which has a satisfactory behavior in many situations, being used to store the configuration necessary to perform compositional approach for context reasoning, providing an easy data access and facilitating its modification if required; (ii) a triple model, which allows the manipulation of ontological data more efficiently compared to its manipulation in memory or using a relational model [20].

Distribution Component communicates with context-aware applications, being responsible for the configuration of the proposal, for receiving the requests made by the applications, and for the visualization of the stored contextual information.

In the middle of Fig. 1, the Reasoning Component has a module called Compositional Strategy Manager, which provides a dynamic strategy for context reasoning. Reasoning Component allows the applications to choose which strategy or strategies they want to use, being the Compositional Strategy Manager

responsible for making the composition of the strategies that will be used to context reasoning. This component receives data collected from sensors of Medium Access Component, transmits the data and situations detected to Storage Component, and sends a notification to the Medium Access Component for the execution of determined actions.

In order to offer multiple reasoning strategies, Reasoning Component provides the Strategies Repository. Each reasoning strategy is cataloged in the repository with a unique Uniform Resource Identifier (URI). Thus, allowing access to a specific strategy that a context-aware application wishes to use.

Among the strategies available to perform the context reasoning, we can mention the use of three of the main strategies [18]:

- rules-based following the event-condition-action pattern. This strategy is used to allows the conception of the compositional approach for context reasoning;
- ontology-based strategy, which can be used in two forms, in the first, the internal axioms in the ontology are used to infer knowledge about the classes represented in the ontology. In the second form, external rules written in the SWRL language are applied, allowing the system to infer new context information exploring the ontology instances;
- supervised learning provides several techniques to be used. In the first step, this strategy uses a training set, where, in this set, the data are categorized [9]. After training, in the second step, exploring the learning acquired is possible to classify new context data.

The dynamic strategy is conceived based on compositional rules, allowing the use of different strategies for reasoning in a combined way. Each rule can contain three different methods: compositional rules, learning techniques, or ontology-based strategy. Each method is represented by marking tags, in which is used a specific tag for each one.

Marking tags begin with the “#” symbol followed by an identifier and a number, thus allowing to recognize the desired composite rule. When the tag has to be replaced by another compositional rule, the “M” identifier must be used. The identifier “A” will be replaced by a supervised learning strategy. Also, identifier “O” can be used to represent an ontology-based strategy.

In order to support the use of the dynamic reasoning strategy, we include specific tables in the relational database in the Storage Component. Figure 2 presents the ER (Entity Relationship) diagram that represents the compositional approach concepts and how it can be modeled to enables their conception.

As presented in Fig. 2, we create an entity to model the strategy used called “Reasoning”. This entity contains the “id” attribute, responsible for identifying the rules, this attribute being the primary key of the table. Applications can identify which rule will be used by searching for “id” and “rule” attribute referring to the compositional rule that will be applied. Also, the “category” attribute is used to find out what kind of strategy will be used, whether it is learning or ontology-based.

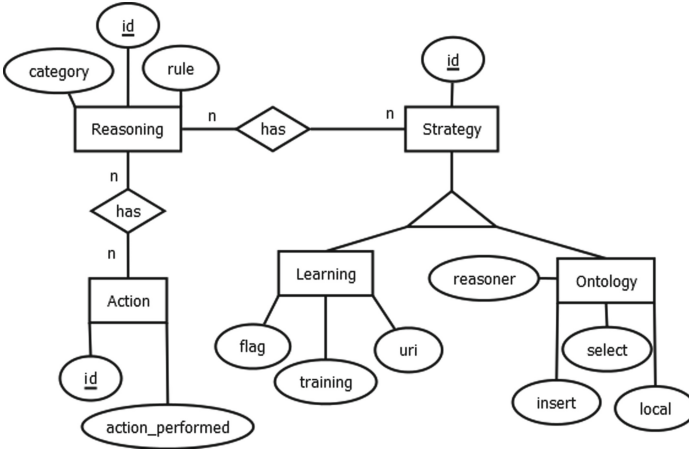


Fig. 2. ER diagram of compositional rules

Considering that, context reasoning can be composed of several strategies. Thus, we define a “Strategy” entity with the “id” attribute responsible for identifying strategies. In Fig. 2, the strategies based on supervised learning and ontologies are modeled in independent entities, being a specialization from “Strategy”.

“Learning” entity has the “uri” attribute, which is responsible for identifying which learning technique should be used. Besides, this entity has the “training” attribute, which specifies which training set should be used to train the chosen learning technique. Also, “flag” attribute is designed to determine the need to perform the training of the selected technique, so when the training step is necessary, its value will be “1”, otherwise “0”.

“Ontology” entity has the “reasoner” attribute, which specifies which statement should be executed to perform the ontology’s reasoning. The “select” and “insert” attributes refer to the SPARQL queries that must be executed to search the data present in the ontology and to insert the data into the triple model of Storage Component. Also, this entity has the “local” attribute, which specifies the location on the disk of the OWL description file of the ontology used.

In Fig. 2 the entity “Action” can also be visualized, which has the “id” attribute that is responsible for the identification of the actions, the value of this attribute is used in the compositional rule to identify which action will be performed. Besides, the entity has the “action_performed” attribute, which specifies the actions to be performed. These actions are transferred to Medium Access Component, which is responsible for carrying out the necessary action. The commands are represented in the compositional rules by the “#C” tag followed by an identifier, which is the “id” attribute of the “Action” table.

Compositional Strategy Manager performs the execution of the compositional rule according to a specific execution flow, as we can see in Fig. 3. Initially, the desired compositional rule is obtained in the Storage Component, and Compositional Strategy Manager performs the identification of other rules present in the initial compositional rule. After, strategy categories explored are recognized. If learning techniques are used, it is necessary to search the “uri” attribute. Also, it is identified which training set should be used for training the technique, and the value of the “flag” attribute is verified to confirm the need to execute the training step.

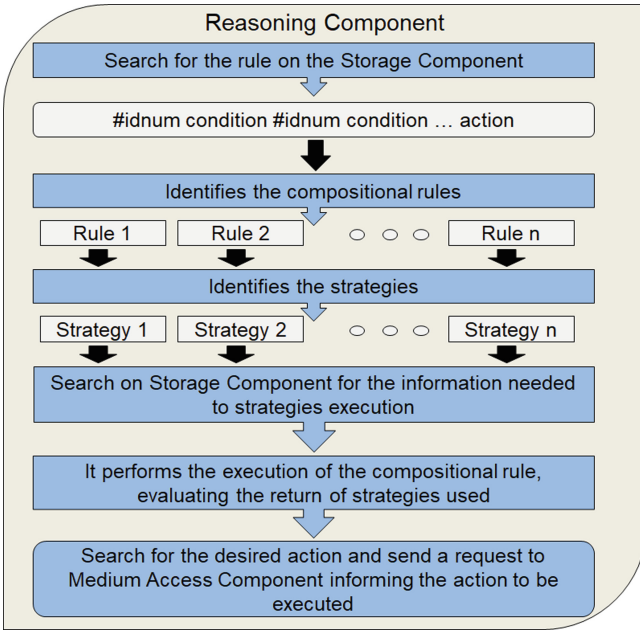


Fig. 3. Execution flow of compositional rule

If the use of ontology-based reasoning was detected, it is necessary to search for the “local” attribute, which contains the location on disk of the ontology, allowing it to be loaded. After that, the Compositional Strategy Manager identifies the mechanism of reasoner chosen, being represented by the “reasoner” attribute. Also, the Compositional Strategy Manager searches the “select” attribute representing the SPARQL query that must be run to fetch the instances that were inferred during the reasoner process. Lastly, the “insert” attribute representing the SPARQL query that will insert the new instances into the triple model of Storage Component is identified.

After the identification of the strategy used and their processing, the Compositional Strategy Manager identifies the command represented in the compositional rule and sends a notification of the need for action to the Medium Access Component.

4 Evaluation

Several evaluation methods are present in the literature, among them, this work highlights the scenario-based assessment, which is considered a mature alternative to be used in context-aware environments [17]. Scenario-based methods evaluate the ability of the software architecture to deal with the demands of a set of interest scenarios [22]. Thus, to evaluate our proposed architecture, we use a scenario-based method.

This scenario is based on work [6], where our architecture monitors the vital signs of patients in cardiac rehabilitation and generates early alerts with the use of scores. For this, we proposed a compositional rule combining a rule-based and supervised learning strategy with the use of regression techniques. Thus, making it possible to predict possible situations of a collapse of a patient and anticipate the activation of the rapid response teams to the action.

With the use of compositional rules, unlike other studies, it is possible to combine the scores generated by different vital signs. Moreover, even the rules used to produce the scores are customizable and can be adapted to distinct patterns of patient behavior. Table 2 shows the values analyzed, in this scenario, to generate the early alerts based on the vital signs obtained.

Table 2. Early alerts score.

Vital sign	1	2	3	0	1	2	3
Heart rate	–	≤40	41–50	51–100	101–110	111–129	≥130
O ₂ saturation	<85	85–89	90–92	≥130	–	–	–

Figure 4 shows the compositional rule used in this scenario. We can see that the compositional rule is composed of two other rules, which are represented by #M1 and #M2, and they have conditionals to be evaluated, resulting in an action to be performed.

First, the Compositional Strategy Manager performs the identification of rules contained in the compositional rule. In this way, the rules M1 and M2 are identified, making it necessary to search for the rules described by the id with values 1 and 2 in the reasoning table, and the substitution of the tags that represent the rules in the compositional rule. After, Compositional Strategy Manager verifies if the rules have other rules represented by #M or if they have some tag #A referring to the learning technique or #O for ontology-based.

```
#M1
#M2
if((heart==2 and saturation==2) or (heart==3 or saturation==3)):
  #C1(name, "critical")
```

Fig. 4. Compositional rule used in evaluation scenario.

With the verification performed, the tags #A1 and #A2 are identified, making necessary find which techniques are present in the Learning table that have the ids with values 1 and 2. After the search, the Compositional Strategy Manager substitutes the tags by the URIs referring to the techniques. As learning strategies are used, the concatenation of the URI with the string “predict” is done, thus allowing to predict the data passed by parameter.

In Fig. 5 is shown the rule referring to tag #M1, which has the id field with value 1, which is used to treat the vital sign heart rate. Figure 6 shows the rule referring to tag #M2, with the id with value 2, which processes the data provided by the vital sign O₂ saturation.

```
if(#A1<130):
  heart=3
elif((#A1<=40) or (#A1>=111 and #A1<=129)):
  heart=2
elif(#A1>=41 and #A1<=50) or (#A1>=101 and #A1<=110):
  heart=1
elif(#A1>=51 and A1<=100):
  heart=0
```

Fig. 5. Rule that replaces the M1 tag.

```
if(#A2<85):
  saturation=3
elif(#A2>=85 and #A2<=89):
  saturation=2
elif(#A2>=90 and #A2<=92):
  saturation=1
elif(#A2>=93):
  saturation=0
```

Fig. 6. Rule that replaces the M2 tag.

Figure 7 shows the rule resulting from the substitution of tag #A1 by the learning technique chosen to perform the reasoning. Where can be visualized the choice of linear learning technique Lars¹. This rule refers to the generation of alerts based on the data collected from the heart rate, creating the values of the score to be analyzed in the compositional rule.

```

if(lars.predict(d)<130):
    heart=3
elif((lars.predict(d)<=40) or (lars.predict(d)>=111 and lars.predict(d)<=129)):
    heart=2
elif((lars.predict(d)>=41 and lars.predict(d)<=50) or (lars.predict(d)>=101 and lars.predict(d)<=110)):
    heart=1
elif(lars.predict(d)>=51 and lars.predict(d)<=100):
    heart=0

```

Fig. 7. Rule with A1 tag replaced.

In Fig. 8 is shown the rule resulting from the substitution of the tag #A2 by the learning technique chosen to carry out the reasoning. Where can be visualized the choice of linear learning technique Lasso². This rule refers to the generation of alerts based on the collected data of the O₂ saturation, creating the values of the score to be analyzed in the compositional rule.

```

If(lasso.predict(d)<85):
    saturation=3
elif(lasso.predict(d)>=85 and lasso.predict(d)<=89):
    saturation=2
elif(lasso.predict(d)>=90 and lasso.predict(d)<=92):
    saturation=1
elif(lasso.predict(d)>=93):
    saturation=0

```

Fig. 8. Rule with A2 tag replaced.

It is important to note that the same parameter is passed for the two reasoning techniques because, in this scenario is always predicted the next expected value for that signal. The values of vital signs are passed to each technique during the training phase, which in this scenario occurs with each new vital sign collection.

¹ http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lars.html.

² http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html.

Figure 9 presents the compositional rule that is applied in this scenario, where all markup tags have already been replaced, both by other rules and by learning techniques. Also, the action that should be executed, if the rule is evaluated as true, has already been inserted in the place of the #C1 tag. This action refers to sending a message by the instant message application to the responsible doctor, where the patient's name and the situation of critical prediction are passed on in the message.

```

If(lars.predict(d)<130):
  heart=3
elif((lars.predict(d)<=40) or (elif lars.predict(d)>=111 and lars.predict(d) <=129)):
  heart=2
elif((lars.predict(d)>=41 and lars.predict(d)<=50) or (elif lars.predict(d)>=101 and lars.predict(d)<=110)):
  heart=1
elif(lars.predict(d)>=51 and lars.predict(d)<=100):
  heart=0
if(lasso.predict(d)<85):
  saturation=3
elif(lasso.predict(d)>=85 and lasso.predict(d)<=89):
  saturation=2
elif(lasso.predict(d)>=90 and lasso.predict(d)<=92):
  saturation=1
elif(lasso.predict(d)>=93):
  saturation=0
if((heart==2 and saturation==2) or (heart==3 or saturation==3)):
  sender.send_msg(str(name), str('Critical'))

```

Fig. 9. Rule applied for alert generation.

In order to analyze the performance of learning techniques when predicting the next value of vital sign reading, it was decided to use measurements made available by the MIMIC database³ (Multiparameter Intelligent Monitoring in Intensive Care). The MIMIC database includes data from more than 90 patients registered in an Intensive Care Unit (ICU), collected between 1992 and 1999, containing an average of 40 hours of physiological signal records [7].

Performance verification was based on the accuracy metric that is the proportion of instances predicted correctly. The use of this metric is justified by the fact that it is widely used to evaluate the quality of the results in several areas of artificial intelligence research with an emphasis on machine learning [8].

In this scenario, it is treated as a correctly classified instance, when the technique accurately predicts which score the future value of vital sign reading will have. In order to analyze the performance of the learning strategy, the prediction of the categories of the next vital sign value was performed using ten patients from the MIMIC database.

³ <https://www.physionet.org/physiobank/database/mimicdb/>.

Table 3. Accuracy of strategy for heart rate.

Patient	0	1	2	3	Total
055	0.996	0.993	0.979	0.9	0.993
211	0.999	0.875	0.836	–	0.997
221	0.998	0.945	0.956	–	0.995
226	0.966	0.977	0.985	0.995	0.979
230	0.999	0.5	0.696	0.571	0.999
248	0.982	0.778	0.832	0.476	0.956
252	1	–	–	–	1
253	0.999	0.92	–	–	0.999
401	0.999	0.5	–	–	0.999
403	1	–	–	–	1

Table 4. Accuracy of strategy for O₂ saturation.

Patient	0	1	2	3	Total
055	0.999	0.848	0.0.907	0.972	0.999
211	0.998	0.989	0.936	0.971	0.996
221	1	–	–	–	1
226	0.998	0.988	0.989	0.993	0.996
230	0.997	0.98	0.989	0.992	0.995
248	0.999	0.950	0.666	0.957	0.999
252	0.998	0.987	0.985	0.991	0.995
253	0.998	0.993	0.978	0.984	0.996
401	0.997	0.989	0.99	0.868	0.993
403	0.996	0.995	0.99	0.951	0.995

Tables 3 and 4 shows the accuracy achieved by the learning techniques employed, both for heart rate and SO₂ saturation. Note that in some cells of the table is contained the character “–”, which represents that for that patient no instance of that category was analyzed.

As we can see in Tables 3 and 4, the performance of learning strategies was satisfactory, reaching more than 0.95 % of hits in the total number instances of each patient. Note that in some cases all the predictions were correct, and these cases happened for the patients that have all the instances of the same score category. The worst accuracy rates obtained were 0.5 %, where it can be analyzed that they are of categories that had few instances, and still, it is of cases that the vital signs change rapidly among the score categories.

5 Conclusion

In this paper, we considered the challenges faced by context-aware applications in IoT, dealing with an increasing number of contextual data from different sources, formats, or semantics. In order to process these contextual data, works present in literature have explored hybrid reasoning strategies. Although many papers propose a hybrid reasoning strategy, we can note the absence of an approach that enables the processing of context information from different reasoning strategies stands out.

Considering these challenges, the following contributions have been achieved with the development of this work: (i) the design of an IoT architecture that supports hybrid context reasoning; (ii) the proposal of a Reasoning Component that provides multiple reasoning strategies; (iii) the conception of a compositional approach for context reasoning that enables the use of different reasoning strategies in a dynamic form, allowing the strategies utilization both individually or in a combined way.

With the conception of the compositional approach for context reasoning, the context-awareness services of the EXEHDA middleware have been updated, and now the middleware can provide different reasoning strategies. Thus, increasing the flexibility of acquired contexts reasoning, improving the identification of situations of interest, and facilitating decision-making.

The proposal was evaluated through one scenario that explores the use of the compositional strategy for context reasoning on the scope of current relevance, which is context awareness in healthcare. In this scenario, the compositional strategy was applied to process the vital signs of heart rate and SO₂ saturation. We used two supervised learning techniques, and combine the results into a single compositional rule to identify risk situations with anticipation. With the use of compositional rules, we possibility the combination of scores of different vital signs, and the creation of rules customizable and adaptable to distinct patterns of patient behavior.

As future work, we highlight: (i) the interfaces developing using data visualization techniques to facilitate the interpretation of the stored contextual information and identified situations; (ii) explore the compositional strategy in different use cases, which can benefit from the features offers for context reasoning.

References

1. Baz, I., Yoruk, E., Cetin, M.: Context-aware hybrid classification system for fine-grained retail product recognition. In: 2016 IEEE 12th Image, Video, and Multi-dimensional Signal Processing Workshop, IVMSWP 2016 (2016). <https://doi.org/10.1109/IVMSWPW.2016.7528213>
2. Bellavista, P., Corradi, A., Fanelli, M., Foschini, L.: A survey of context data distribution for mobile ubiquitous systems. *ACM Comput. Surv.* **44**(4), 24:1–24:45 (2012). <https://doi.org/10.1145/2333112.2333119>
3. Bettini, C., et al.: A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.* **6**(2), 161–180 (2010). <https://doi.org/10.1016/j.pmcj.2009.06.002>

4. Can, O., Sezer, E., Bursa, O., Unalir, M.O.: Comparing relational and ontological triple stores in healthcare domain. *Entropy* **19**(1), 30 (2017)
5. United States. Federal Trade Commission: The Internet of Things: Privacy and Security in a Connected World. Federal Trade Commission staff reports, DIANE Publishing Company, USA (2015). <https://books.google.com.br/books?id=alQ7rgEACAAJ>
6. Dias, F.O.: Um Modelo Proativo de Antecipação de Ações de Times de Resposta Rápida Baseado em Análise Preditiva. Dissertação de mestrado em computação aplicada, Unisinos, São Leopoldo-RS (2017)
7. Goldberger, A.L., et al.: Physiobank, physiotookit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation* **101**(23), e215–e220 (2000). *Circulation Electronic Pages*
8. Japkowicz, N., Shah, M.: *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York (2011)
9. Khan, N., Alegre, U., Kramer, D., Augusto, J.C.: Is ‘context-aware reasoning = case-based reasoning’? In: Brézillon, P., Turner, R., Penco, C. (eds.) *CONTEXT 2017*. LNCS (LNAI), vol. 10257, pp. 418–431. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57837-8_35
10. Khattak, A.M., et al.: Context representation and fusion: advancements and opportunities. *Sensors* **14**(6), 9628–9668 (2014). <https://doi.org/10.3390/s140609628>
11. Kitchenham, B., Charters, S.: *Guidelines for performing systematic literature reviews in software engineering*. Joint report, Keele University and Durham University, Durham (2007)
12. Li, X., Eckert, M., Martínez, J.F., Rubio, G.: Context aware middleware architectures: survey and challenges. *Sensors* **15**(8), 20570 (2015). <https://doi.org/10.3390/s150820570>. <http://www.mdpi.com/1424-8220/15/8/20570>
13. Li, X., Martínez, J.F., Rubio, G.: Towards a hybrid approach to context reasoning for underwater robots. *Appl. Sci.* **7**(2) (2017). <https://doi.org/10.3390/app7020183>
14. Lopes, J., et al.: A middleware architecture for dynamic adaptation in ubiquitous computing. *J. Univers. Comput. Sci.* **20**(9), 1327–1351 (2014). http://www.jucs.org/jucs_20_9/a_middleware_architecture_for
15. Lopes, J.L., et al.: A distributed architecture for dynamic contexts composition in ubicomp. In: *38th Latin America Conference on Informatics - CLEI 2012 - Conference Proceedings* (2012). <https://doi.org/10.1109/CLEI.2012.6427253>
16. Machado, R., Almeida, R., Lopes, J., Pernas, A., Yamin, A.: Construção do estado da arte em estratégias híbridas para raciocínio de contexto: uma abordagem explorando revisão sistemática da literatura. In: *SBCUP - 10º Simpósio Brasileiro de Computação Ubíqua e Pervasiva* (2018)
17. Patidar, A., Suman, U.: A survey on software architecture evaluation methods. In: *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 967–972, March 2015
18. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: a survey. *IEEE Commun. Surv. Tutor.* **16**(1), 414–454 (2014). <https://doi.org/10.1109/SURV.2013.042313.00197>
19. Razzaq, M.A., Amin, M.B., Lee, S.: An ontology-based hybrid approach for accurate context reasoning. In: *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 403–406, September 2017. <https://doi.org/10.1109/APNOMS.2017.8094159>
20. Rosa, F.L., Machado, R.S., Cavalheiro, G.G.H., Yamin, A.C., Pernas, A.M.: Análise de desempenho de sistemas de gerenciamento de dados em triplas com base no benchmark watdiv. *WSCAD-WIC* (2017)

21. Sánchez Guinea, A., Nain, G., Le Traon, Y.: A systematic review on the engineering of software for ubiquitous systems. *J. Syst. Softw.* **118**(C), 251–276 (2016). <https://doi.org/10.1016/j.jss.2016.05.024>
22. Shanmugapriya, P., Suresh, R.M.: Software architecture evaluation methods - a survey. *Int. J. Comput. Appl.* **49**(16), 19–26 (2012)
23. Smaaberg, S., Shabib, N., Krogstie, J.: A user-study on context-aware group recommendation for concerts. In: *CEUR Workshop Proceedings - SP 2014: Social Personalisation Workshop* (2014)
24. Strobbe, M., Van Laere, O., Dhoedt, B., De Turck, F., Demeester, P.: Hybrid reasoning technique for improving context-aware applications. *Knowl. Inf. Syst.* **31**(3), 581–616 (2012). <https://doi.org/10.1007/s10115-011-0411-7>
25. Yuan, B., Herbert, J.: Context-aware hybrid reasoning framework for pervasive healthcare. *Pers. Ubiquit. Comput.* **18**(4), 865–881 (2014). <https://doi.org/10.1007/s00779-013-0696-5>