Contextual Defeasible Reasoning Framework for Heterogeneous Systems

Salwa Muhammad Akhtar[1] and Hafiz Mahfooz Ul Haque[2(✉)]

[1] Department of Computer Science, University of Lahore, Lahore, Pakistan
salwaakhtar0728@gmail.com
[2] Department of Software Engineering, University of Lahore, Lahore, Pakistan
mahfoozul.haque@se.uol.edu.pk

Abstract. This paper presents a contextual defeasible reasoning based multi-agent formalism to model heterogeneous systems using the notion of a multi-context system. This framework relies on the semantic knowledge sources which allow us to model context-aware non-monotonic reasoning agents to infer the desired goals using the extracted rules from the ontologies and handles inconsistencies using conflicting contextual information. We illustrate the validity and correctness of the proposed formalism using a simple case study of a smart healthcare system with the prototypal implementation of the system.

Keywords: Multi-context systems · Context-awareness · Defeasible reasoning · Semantic knowledge representation

1 Introduction

Context-aware systems and applications have gained significant attention in mobile and pervasive computing paradigms to model heterogeneous knowledge sources. These systems facilitate users by providing customized information or services at any time and anywhere using personalized smart devices. The customized information can be provided according to the environment in a specified contextual format. The smart devices acquire contextual information autonomously, perform reasoning, interact with other users or devices, and adapt their behavior in the rapidly changing environment. Literature has revealed different notions of the context so far. In this work, we follow the approach proposed by Dey et al. [9] for modeling contextual information to be used by context-aware reasoning agents. Context modeling and reasoning has been considered to be the most promising and fundamental research area in context-aware computing. When these context-aware smart devices interact with each other to fulfill the user's desires, they usually perform context modeling and reasoning [12]. In the literature, numerous context-aware frameworks have been proposed incorporating different context modeling and reasoning approaches including ontology-based systems, rule-based reasoning, etc. [5,21,22]. Formal modeling of such systems are mostly based on client-server architecture or centralized in nature but

P. C. Vinh and A. Rakib (Eds.): ICCASA 2020/ICTCC 2020, LNICST 343, pp. 16–30, 2021.
https://doi.org/10.1007/978-3-030-67101-3_2

perform distributed reasoning using different smart devices and/or agents. However, the imperfect nature of contextualized information in a heterogeneous environment is often become a very challenging task due to the inconsistent behavior of the system. As a result, these systems often become useless and unable to draw a plausible conclusion due to inconsistent and incomplete information to be shared among the heterogeneous environment. To resolve this issue in a real context-aware deployment setting, the coalition of heterogeneous domain modeling could be useful to exchange contextual information. In this connection, the multi-context system can be used for interlinking different knowledge sources using a predefined communication mechanism to preserve the identity and independence of each of the knowledge sources. A multi-context system (MCS) consists of a set of contexts where each context is a self-contained knowledge source having a set of contextual information (facts) and a set of inference rules and allows the flow of information among different contexts [11].

In the literature, several approaches have been proposed using multi-context systems [6,8,13,16,17]. In [8], Brewka et al. have proposed a multi-context reasoning framework to combine arbitrary monotonic and nonmonotonic logics. They model the flow of information among different contexts using bridge rules. In [6], the authors have focused on a distributed reasoning approach for modeling ambient agents to form a peer-to-peer system. In this framework, distributed algorithm based preference ordering has been used to handle inconsistencies. The authors in [13] have used mapping rules to govern the flow of information among different contexts in an MCS. They have used defeasible rules to manage the inconsistencies in the MCS. In [17], the authors have proposed an MCS based framework to handle inconsistencies for modeling the heterogeneous system. The communication flow among different contexts is modeled using bridge rules for semantic knowledge sharing. However, authors in [16] used mapping rules to model communication among different contexts. In contrast with the previous work, this approach is novel in a sense that context-aware agents use contextual defeasible reasoning based formalism to model the flow of information among different contexts and handle inconsistencies using a defeasible reasoning approach. This formalism relies on semantic knowledge sources where domains are modeled using ontologies. The rules used by agents are extracted from corresponding ontologies to model the system to infer the desired goals. We illustrate the use of the proposed framework using a simple example and its prototypal implementation using Python programming.

The rest of the paper is organized as follows. In Sect. 2, we briefly review the core notions of the multi-context system and contextual defeasible reasoning. Section 3 presents the semantic knowledge formulation to contextualize heterogeneous knowledge sources. In Sect. 4, we present contextual defeasible reasoning based multi-agent formalism for heterogeneous systems. In Sect. 5, the prototypal implementation of the proposed framework is provided using a simple case study of the smart healthcare system. Section 6 briefly presents the related work, and finally concludes in Sect. 7.

2 Preliminaries

2.1 Multi-Context System

A multi-context system consists of a set of contexts (knowledge sources) where each context has its own knowledge base, a set of inference rules, and a logic used to perform reasoning. Multi-context systems have been considered as one of the most suitable approaches for interlinking different contexts where each context may be formalized using formal knowledge representation languages. The multi-context system has applications in different areas such as data integration, argumentation, multi-agent systems. The knowledge representation model based on the multi-context system paradigm uses inference rules for local and distributed reasoning to exchange information among different contexts. Non-monotonic features can be included in the multi-context system to resolve potential inconsistencies in distributed contextual knowledge sources [6,8,10,13]. MCS is mainly used to express the individualized domain-specific knowledge as well as the inter-contextual flow of information among different contexts [10]. Formally speaking, a multi-context system $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ is a collection of contexts where each context in MCS is a triple; $\mathbb{C}_i = \{L_i,\ KB_i,\ R_i\ \}$ where L_i is a logic, KB_i is a knowledge base and R_i is a set of local and mapping rules. There are two types of rules in MCS: local rules and mapping rules. The body of the local rules is a conjunction of contextualized information from the local context and these rules can either be strict rules or defeasible rules or a combination of both. Mapping rules perform reasoning by conjoining local contextual information with the foreign contextual information and infer the derived goals considering local contextual information.

Inconsistency is one of the major issues in the multi-context systems, as an inconsistent MCS produces no results and thus inconsistency renders the whole system useless. Another issue being dealt with in multi-context systems is the availability of incomplete information. This type of issue arises when the system is unable to generate the required information due to some software or hardware malfunction. When such incomplete information arrives in the system, due to the unavailability of complete information the system is unable to infer the desired results and thus the system becomes useless. In this work, we intend to solve the inconsistency issue due to which the system losses its state of Equilibria, i.e., the system as a whole does not have a belief state. This happens when the contextual information received by the system is contradictory or inconsistent. In such situations, the multi-context system becomes incapable of deriving plausible results which renders it useless. This becomes a serious issue, especially in the safety-critical system such as in healthcare systems.

2.2 Contextual Defeasible Logic (CDL)

Contextual defeasible logic has emerged from multi-context systems and defeasible reasoning. CDL is based on the rule-based reasoning technique following the principles of defeasible logic to handle incomplete and inconsistent information.

Due to its low computational complexity, defeasible logic has been considered as one of the most promising techniques in non-monotonic reasoning. It is a simple and efficient reasoning technique to perform reasoning monotonically as well as non-monotonically when developing expert decision making systems [2]. This reasoning is useful for deriving plausible conclusions even with partial or conflicting information. The conclusions drawn from this reasoning are tentative, therefore a conclusion can be withdrawn when more authentic information is obtained. CDL essentially supports two kinds of reasoning, local reasoning, and global reasoning. Local reasoning is performed by strict rules or defeasible rules or both, while global reasoning is performed by mapping rules. Strict rules state that if the premises of the rule are true, so is the conclusion whereas defeasible rules can be defeated by contrary evidence. Both the strict and defeasible rules define the local contextual knowledge acquired from a single ontology [13]. Strict rules are of the form:

$$r_i^s : a_1^i, a_2^i, \ldots, a_{n-1}^i \rightarrow a_n^i.$$

whereas defeasible rules are of the form:

$$r_i^d : a_1^i, a_2^i, \ldots, a_{n-1}^i \Rightarrow a_n^i.$$

On the other hand, mapping rules are constructed from different knowledge sources (contexts). These rules coalition the local contextual information ($a_1^i \in \mathbb{C}_i$) and the foreign contextual information ($a_3^j \in \mathbb{C}_j$) to model the system. Local contextual knowledge consists of facts and rules of the same ontology, while foreign contextual knowledge consists of the facts and rules of other ontology. Mapping rules are interpreted as defeasible rules, and are of the form:

$$r_i^m : a_3^j, a_1^i, \ldots, a_{n-1}^i \Rightarrow a_n^i.$$

The above rule $\Re^m$ is a set of mapping rules and $\mathcal{T}_i$ is a preference ordering on the $\mathbb{C}$. Modeling of local rules can be done using a single ontology however modeling of mapping rules requires multiple ontologies.

3 Contextualizing Semantic Knowledge Sources

In recent years, literature has revealed significant contributions in modeling and reasoning heterogeneous systems using semantic knowledge representation formalisms. To model the domains of real-time environment, knowledge engineers opt for different knowledge representation formalisms according to the requirements and define a set of facts and relationships to perform reasoning in order to model a knowledge-based system. Knowledge-based systems use knowledge representation techniques to resolve human-like assisted decision making using inference rules. Literature has revealed several languages to develop knowledge-based systems [23]. Among others[4], the ontology-based approach has been advocated as the most promising ones in the pervasive computing environment due to its simplistic and efficient reasoning capability, flexibility, and expressiveness. In addition, ontology-based systems provide semantic knowledge interoperability

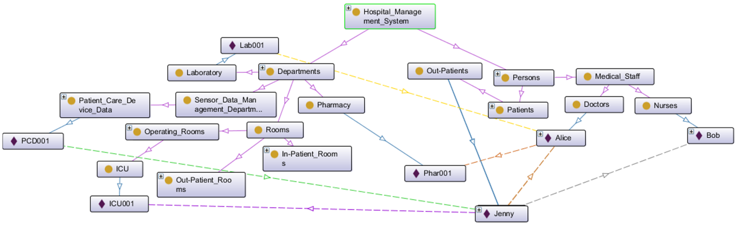

Fig. 1. A fragment of smart hospital ontology

with the pervasive systems to perform distributed reasoning using contextual modeling techniques [12,18]. Gruber et al. [14] define ontology as "*an ontology is a formal specification of a conceptualization.*" In the semantic web, ontology is an explicit conceptualization of the domain that can be used to model the system using language constructors in the form of classes (concepts) with their predefined relationships, axioms and rules to support inferencing, store the conceptualized information in a structured manner, and query the information. In the literature, ontology has been classified in two different versions by World Wide Web Consortium (W3C) as OWL 1 (Web Ontology Language) and OWL 2. OWL 1 has three sub-languages named as OWL Lite, OWL DL and OWL Full. OWL 2 has also three sub-languages named as OWL 2 EL, OWL 2 QL and OWL 2 RL. Each of the sub-languages in OWL 1 and OWL 2 can be used according to reasoning capability, expressivity, and scalability of the system.

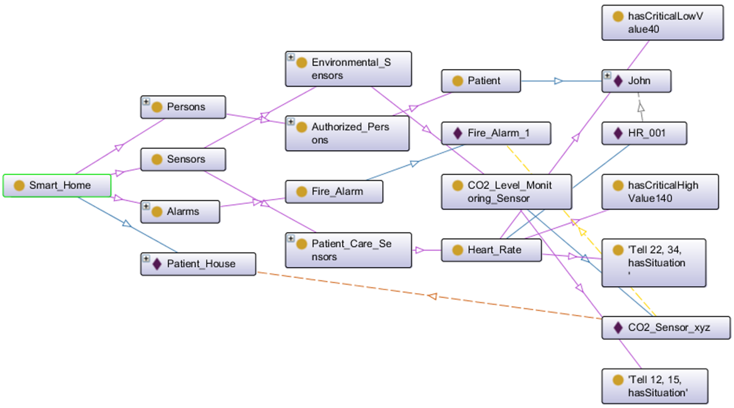

Fig. 2. A fragment of smart home ontology

In [15,21,25], it has been shown how the semantic knowledge sources (ontologies) can be used for context modeling and reasoning using context-aware rule-based reasoning agents. According to Brewka et al. [8], a single context in MCS can be represented in the form of ontology which is an independent domain itself.

Benslimane et al. [5] describe ontology as an independent knowledge source having a set of axioms and inference rules. In contrast, the multi-context system consists of a set of contexts having a set of rules used in the ontology. The formalism proposed in this work empowers the distributed knowledge sources to be interpreted in a heterogeneous fashion and enables the information flow among different contexts. In this work, we consider the semantic knowledge-based context modeling approach. As the proposed framework is heterogeneous in nature using the notion of multi-context system and domains are modeled in ontologies. For this, we develop two ontologies of smart home and smart hospital systems. These are very comprehensive ontologies and it is very hard to show these in this paper. However, fragments of these ontologies can be seen in Fig. 1, Fig. 2 and class hierarchies of both ontologies are shown in Fig. 3. To model the system, we extract contextual information independently form both ontologies with the intention of preserving the identity and independence of each specialized domain. We choose OWL 2 RL and Semantic Web Ontology Language (SWRL) due to its reasoning capability, genericity, extensibility, and expressiveness. Using OWL 2 RL and SWRL, we develop complex rules for each specialized domain that can be transformed into horn-clause rules format and is suitable for the design and development of a rule-based reasoning system [16,17].

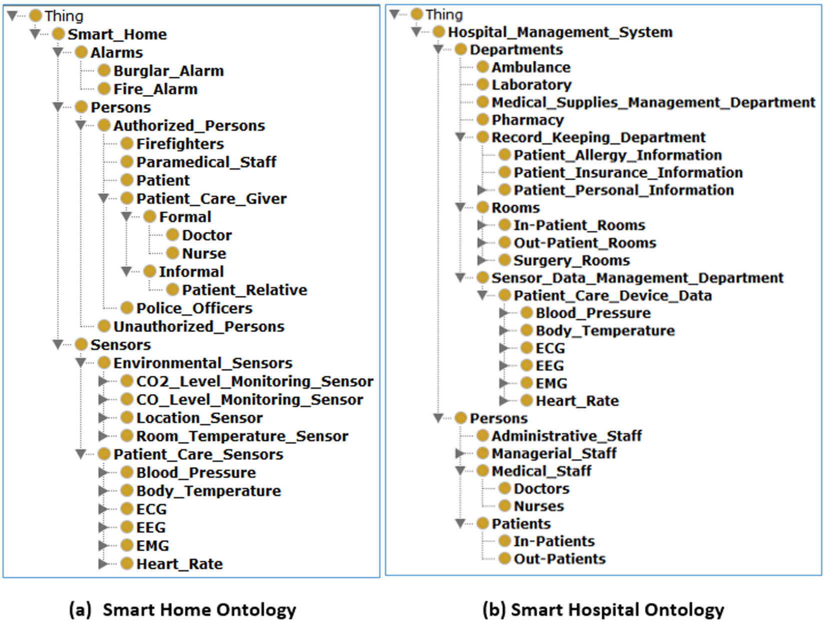

Fig. 3. Class hierarchies of smart healthcare system

4 Contextual Defeasible Reasoning Based Multi-agent Formalism

In this section, we present a contextual defeasible reasoning based multi-agent formalism to model heterogeneous systems using the notion of a multi-context system. In a context-aware multi-agent system (MAS), each agent in the system consists of a knowledge-base having a set of strict and defeasible rules along with a set of contextualized information and a reasoning strategy. These context-aware agents acquire contextualized information from their corresponding knowledge sources. The contextualized information (the set of facts and the set of rules) may either be obtained from a single ontology or multiple ontologies. These agents perform reasoning based on the set of rules. If the reasoning is being performed on the rules which are obtained from a single ontology, then it is known as local reasoning. If the agents perform reasoning on the set of rules and set of facts which are extracted multiple ontologies, then it is known as distributed reasoning [7,20]. To model a centralized system, agents may use individualized knowledge source whereas, in case of a decentralized heterogeneous system, agents acquire information from different knowledge sources using a set of mapping rules along with the set of strict and defeasible rules to perform reasoning, share information among agents and then adapt their behavior accordingly (Fig. 4).

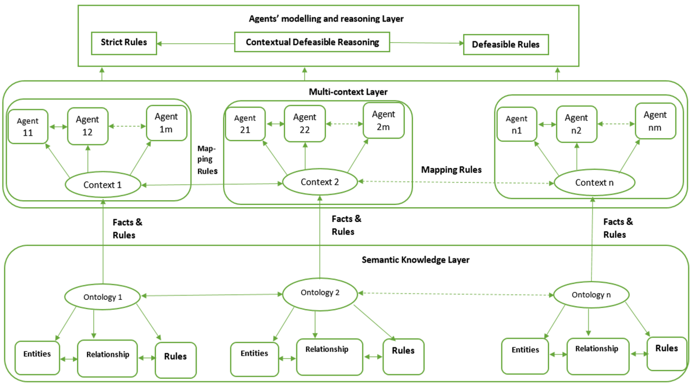

Fig. 4. Proposed framework of the system

In a multi-context setting, every context is composed of a group of agents where each agent has the capability to perform its specified tasks as designated by its corresponding knowledge source (context). To suitably model a heterogeneous system, we define MCS as $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ where $\mathbb{C}$ represents the heterogeneous system in which each context $\mathbb{C}_i$ is considered as a sub-domain of $\mathbb{C}$, for all $i \in \mathbb{C}$. Each context in MCS includes a triple (V_i, R_i, T_i), where contextual knowledge in C_i is represented by vocabulary V_i, R_i represents a set of rules and T_i is a preference ordering on C. In each context in MCS, we develop multi-agent system consisting of $n_{Ag}(\geq 1)$ agents, i.e., $A_g = \{1, 2, ..., n_{Ag}\}$, where each agent is expressed as a triple $(\Re, \mathcal{F}, \succ)$. In the set, $\Re$ represents the set of all rules derived from multiple ontologies, $\mathcal{F}$ is a finite set of facts that are stored in the working memory of the agents and $\succ$ is the superiority relation on $\Re$. As the proposed framework runs in a highly decentralized environment, therefore there is a need of different kinds of rules for modeling and reasoning.

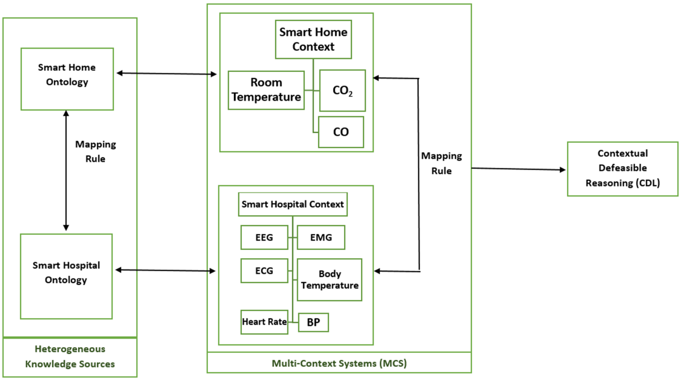

Fig. 5. Framework reasoning process

In the system, the set $\Re = \{\Re^s, \Re^d, \Re^m, \Re^c\}$ is a finite set of rules including strict, defeasible, mapping rules and communication rules respectively. Strict rules $(\Re^s)$ follow the straightforward reasoning approach and these rules are non-contradictory in nature whereas defeasible rules $(\Re^d)$ can be defeated based on contrary evidence. Mapping rules $(\Re^m)$ associate contextual information from different contexts and perform reasoning to derive the results for the local context. The set of facts from local or foreign context are formed to make the mapping rules and these rules are interpreted defeasibly. Mapping rules can change the perception based on conflicting information in order to reduce the inconsistency and suitable mechanism for the coalition of different contextual data from different contexts, to model heterogeneous systems. Hence the set $\Re$ which is $\langle \Re^s \cup \Re^d \cup \Re^m \cup \Re^c \rangle$ may be also contradictory. The superiority relation $\succ$ in the rules can be used to prioritize the set of rules in case if conflicting information arrives in the working of agents at the same time interval. Rules are of the form $P_1, P_2, \ldots, P_n \hookrightarrow P$. In a rule instance, the antecedents $P_1, P_2, \ldots, P_n$ and the consequent P are contextual data. In the rest of this logic, we assume $\rightarrow$ arrow for strict rules and communication rules, and $\Rightarrow$ arrow for defeasible and mapping rules respectively. To model the communication, agents usually exchange information via message passing and coordinate among themselves in order to infer the desired goals. In our model, agents use special communication primitive; i.e., "*Tell(i, j, C)*" which means that agent i shares a context C with the agent j. As the proposed formalism follows the rule-based reasoning strategy, so argument "*Tell(i, j, C)*" may appear either in the antecedent or in the consequent of the rule. Whenever "*Tell(i, j, C)*" appears in the consequent, then the rule is said to be a communication rule. All other rules including strict, defeasible, and mapping rules are known as deduction rules. Even if the argument "*Tell(i, j, C)*" appears in the antecedent, then it is also known as deduction rules. Due to the heterogeneity of the proposed model, mapping rules can also be used for the exchange of information among different agents in different contexts. In the system, a multi-context system is composed of a set of rule-based agents that perform three core actions: *(a) Rule*, *(b) Copy* and *(c) Idle*. *Rule* instances are of different types such as strict rules, defeasible rules, mapping rules, and communication rules. *Copy* action is performed by firing the rule instances of communication rules whenever agents exchange the contextual information. *Idle* action allows agents to remain in an idle state but transit to the next state is triggered by the system. The system performs defeasible reasoning non-deterministically, and the rule priorities are static and set by the system designer at the design time of the system in order to avoid inconsistent behavior of the system.

5 Case Study: Prototypal Implementation of the System

To illustrate the use of proposed formalism, we model a comprehensive smart healthcare system considering two independent domain ontologies, namely Smart Hospital System and Smart Home. This case study specifically focuses on Parkinson's disease patients who mostly stay at home and can't escape themselves immediately in hazardous situations. The core purpose of developing this case study is to model the heterogeneous system using context-aware reasoning agents for sharing knowledge across different domains. The system can be installed in the patient's house to detect abnormalities in the patient's condition and to allow the continuous monitoring of the patient's physiological situation, and generate alerts in the form of messages, emails and notifications, and send them automatically to the doctor or caregiver. After considerable deliberation, two different domains (contexts) are developed to implement the multi-context system based formalism. We assume the proposed system consists of CO_2, CO and room temperature monitoring sensors embedded in the patient's home, and a smart device is attached with the patients to monitor the vital signs using sensors such as heart rate, blood pressure, body temperature, EEG, ECG, and EMG and provide the required information to the agents modeled in the system. A set of facts and rules are extracted from different domain ontologies and assigned to each agent in the system correspondingly. As the system is heterogeneous in nature and runs a highly decentralized environment, therefore it is assumed that the agents in this system acquire and share contextual information in an autonomous manner. Each of these agents has its own knowledge-base consisting of a set of rules along with the set of facts, and a reasoning mechanism. These agents perform three main actions: *(a) Rule*, *(b) Copy* and *(c) Idle*. *Copy* actions are triggered by firing communication rules instances to exchange the information among agents and *Idle* actions are triggered for the transition to the next states while leaving agents configuration unchanged. For *Rules* actions, there are three kinds of rules which are triggered by firing the rule instances. However, their interpretations are different for strict rules and defeasible rules. Mapping rules are interpreted defeasibly by associating the facts from different contexts (domains). Agents use these actions for modeling and reasoning to represent the overall behavior of the system as shown in Fig. 5.

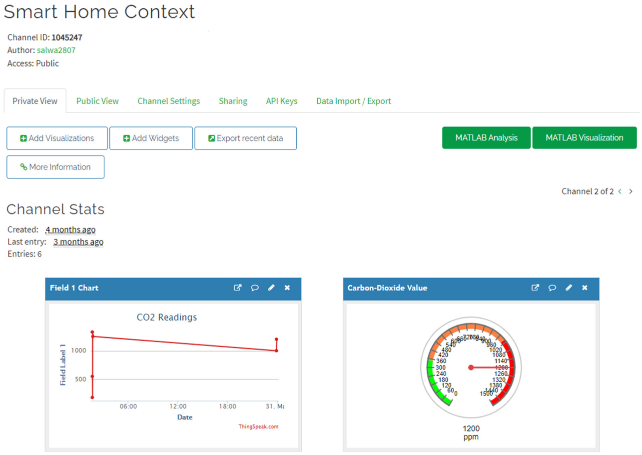

Fig. 6. Thingspeak channels with their access

In a multi-context setting, three agents (room temperature agent, carbon dioxide monitoring agent, and carbon monoxide monitoring agent) are developed for smart home and six agents (heart rate monitoring agent, body temperature agent, blood pressure agent, EEG monitoring agent, ECG monitoring agent, and EMG monitoring agent) are developed for the smart hospital context. These agents are programmed in Python language using Notepad++ editor. In both contexts, context-aware agents are developed for acquiring contextual information, processing data and perform distributed reasoning accordingly. These agents generate values based on the acquired information using the Thingspeak channel. As the system starts execution, agents start performing the reasoning process no-deterministically in order to achieve the desired goals. A total of nine agents are developed which are sending data to the Thingspeak platform. In order to receive data on the Thingspeak platform, an account was first made for the Thingspeak website. After signing into Thingspeak, two channels are developed for each context namely, smart home and smart hospital. These channels received the data generated by the virtual agents. These channels have the ability to store the received data in the Thingspeak cloud until the user deletes the data. Figure 6 shows the channels created on the Thingspeak platform for implementing the prototype of the proposed framework. Thingspeak react is a Thingspeak application that allows us to trigger actions in the form of sending emails, notifications, SMS, etc. when the data received by a Thingspeak channel. For the sake of this work, three reacts were developed which sent email, notification, and SMS respectively whenever the data met certain conditions. Figure 7 shows the Thingspeak reacts created on Thingspeak platform for implementing the prototype of the proposed framework.

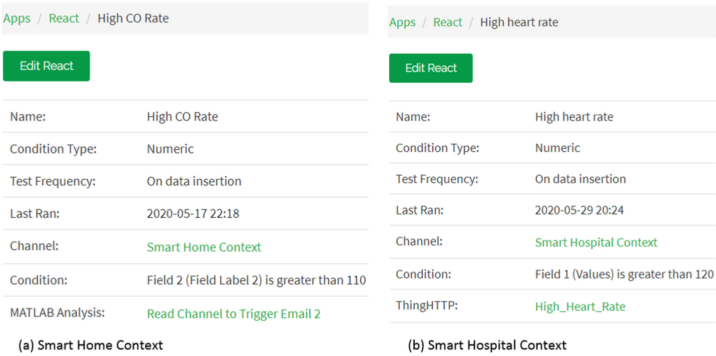

Fig. 7. Thingspeak reacts

Thingspeak HTTP is a Thingspeak application that allows us to connect services and applications to a Thingspeak channel using the HTTP protocol. We use two services; Twilio and IFTTT for SMS and notification respectively. MATLAB software has been integrated with Thingspeak and can be used to write the source code for email generation [24]. Figure 9 shows the source code used to send the email. An Account on the IFTTT platform is required to access its services. Once an account has been created it can be used on multiple devices to interconnect them. IFTTT has access to many services including Webhook, a technique using a callback function to access and alter an application's behavior. This framework uses IFTTT's webhook access to send notifications. An account on the Twilio website is required to utilize its services. This framework uses Twilio to send an SMS. Twilio uses the SMS gateway and the SMPP protocol to send an SMS to the specified number.

After developing agents, connecting them to Thingspeak platform, and then connecting to other services for the alert generation, the prototype of the proposed framework is started to perform actions to check the validity of the system. As a result, it is noticed that the agents successfully transfer data to Thingspeak channels. For determining whether the system will generate an SMS when required, agents have been programmed for this. Whenever the heart rate monitoring agent is activated and it starts generating random values between 0 bpm and 200 bpm. When the system detects the values below the 120 bpm threshold the system behaves normally. However, as the system detects the value above the normal range, for example; 120 bpm, the system automatically generates a notification and sends the notification to the user's specified device. Figure 8 shows

Table 1. Example rules for smart healthcare system

Agent 1: Smart Home Care Agent
Initial facts: Location('Home), CO2Level('1300), currentCO2Level('Home, '1300), CO2GreaterThan ('1300, '1000), COLevel('17), currentCOLevel('Home, '17), COGreaterThan ('17, '15)
R11: Location(?l), currentCO2Level(?l, ?co2), CO2Level(?co2), CO2GreaterThan (?co2, '1000) → CarbonDioxideSituation (?l, "Dangerous") R12: Location(?l), currentCOLevel(?l, ?co), COLevel(?co), COGreaterThan (?co, '15) → CarbonMonoxideSituation (?l, "Dangerous") R13: CarbonDioxideSituation (?l, "Dangerous") → Tell (1, 4, (CarbonDioxideSituation (?l, "Dangerous")) R14: CarbonMonoxideSituation (?l, "Dangerous") → Tell (1, 4, (CarbonMonoxideSituation (?l, "Dangerous"))
Agent 2: Heart Rate Monitoring Agent
Initial facts : Person ('Alice), HeartRate('127), hasHeartRate('Alice, '127), GreaterThanEqual('127, '100)
R21: person(?p), hasHeartRate(?p, ?hr), HeartRate(?hr), GreaterThanEqual(?hr, '100) → hasHeartBeat (?p, "Abnormal") R22: person(?p), hasHeartRate(?p, ?hr), HeartRate(?hr), LessThanEqual(?hr, '60) → hasHeartBeat (?p, "Abnormal") R23: hasHeartBeat (?p, "Abnormal") → Tell (2, 4, (hasHeartBeat (?p, "Abnormal"))
Agent 3: EEG Monitoring Agent
Initial facts : Person ('Alice), EEG('5), hasEEG('Alice, '5), LessThanEqual('5, '8)
R31: person(?p), hasEEG(?p, ?eeg), EEG(?eeg), GreaterThanEqual(?eeg, '8) → PatientEEG(?p, "Abnormal") R32: person(?p), hasEEG(?p, ?eeg), EEG(?eeg), LessThanEqual(?eeg, '8) → PatientEEG(?p, "Abnormal") R33: PatientEEG(?p, "Abnormal") → Tell (3, 4, (PatientEEG(?p, "Abnormal"))
Agent 4 : Smart Hospital Control Agent
Initial facts : Person ('Alice), PatientID('P1), hasPatientID ('Alice, 'P1), PatientCondition ('critical), Location('Home)
R41: Person(?p), hasPatientID(?p, ?PID), PatientID(?PID) → Patient(?p) R42:'Tell(1,4,(CarbonMonoxideSituation(?l,"Dangerous")) → CarbonMonoxideSituation(?l,"Dangerous") R43: 'Tell (1,4,(CarbonDioxideSituation(?l, "Dangerous")) → CarbonDioxideSituation(?l, "Dangerous") R44: 'Tell (2, 4, (hasHeartBeat(?p, "Abnormal")) → hasHeartBeat (?p, "Abnormal") R45: 'Tell (3, 4, PatientEEG(?p, "Abnormal")) → PatientEEG(?p, "Abnormal") R46: hasHeartBeat (?p, "Abnormal") ⇒ PatientCondition(?p, Critical) R47: PatientEEG(?p, "Abnormal") ⇒ PatientCondition(?p, Critical) R48: hasHeartBeat (?p, "Abnormal"), PatientEEG(?p, "Abnormal") ⇒ PatientCondition(?p, Critical) R49: CarbonMonoxideSituation (?l, "Dangerous") ⇒ LocationCondition(?l, "Critical") R410: CarbonDioxideSituation (?l, "Dangerous") ⇒ LocationCondition(?l, "Critical") R411: CarbonMonoxideSituation (?l, "Dangerous"), CarbonDioxideSituation (?l, "Dangerous") ⇒ LocationCondition(?l, "Critical")
Superiority Relation: R48 ≻ R46, R411 ≻ R410, R411 ≻ R49

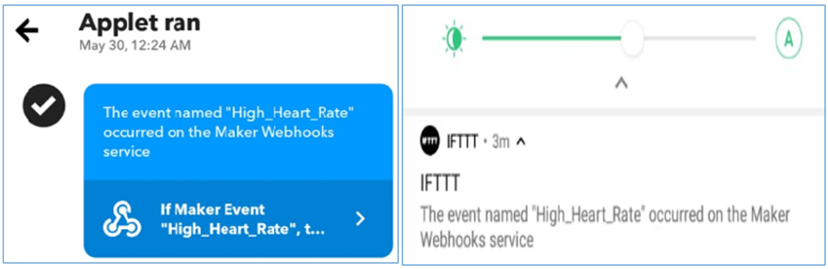

Fig. 8. Thingspeak channel field receiving data from corresponding agents

```
alert_body = 'CO Level is not in normal range.';
alert_subject = 'CO Level Danger';
alert_api_key = 'TAK3H92UP1852CRXX';
alert_url= "https://api.thingspeak.com/alerts/send";
jsonmessage = sprintf (['{"subject": "%s", "body": "%s"}'], alert_subject, alert_body);
options = weboptions ("HeaderFields", {'Thingspeak-Alerts-API-Key', alert_api_key; 'Content-Type',
'application/json'});
result = webwrite (alert_url, jsonmessage, options);
```

Fig. 9. Matlab code for email generation

the notification received in the notification panel and int the IFTTT mobile application respectively. Some example rules of the smart healthcare system can be seen in Table 1 to show the reasoning process of the system. These rules are extracted from ontologies and can be used by the agents in the development of a preference model to support customized preferences in order to defeat the contrary evidence.

6 Related Work

In the literature, a significant effort has been made to resolve inconsistencies in heterogeneous systems using defeasible reasoning with the incorporation of multi-context systems to leverage human lively situations. In [6], the authors proposed a multi-context systems based argumentation framework incorporating the non-monotonic features that use the preference information to resolve conflicts, and a distributed algorithm for query evaluation. They have proposed a framework to resolve the potential conflicts caused by the interaction of different contexts through mappings. To resolve potential inconsistencies, they have used a mobile phone-based application to evaluate the contextual information received from various sources and determined the preferences based decision support mechanism in order to avoid conflicts. Antoniou et al. in [3] proposed a semantic knowledge-based context-aware meeting alert application. This application acquires different types of contexts such as location, environment, time, calendar information, and whether services. The user's calendar uses the contextual information stored in the server. They use DR-Prolog defeasible reasoning engine to perform reasoning based on the user's rules to infer the appropriate decisions and notifies the user by showing an alert message about the upcoming scheduled event. In [1], the authors proposed a defeasible logic-based framework for multi-context distributed systems and distributed reasoning processes. The authors, in this work, extended the defeasible logic theory using the notion of meta-rules to perform reasoning over theories for checking the validity and correctness of the system. The inconsistencies occurring in the framework were dealt with using defeasible logic. In [19], the authors have presented a rule-based contextual reasoning framework for ambient intelligence and demonstrated its use in ambient assisted living environment, by developing an application using the Kevoree software development platform. It uses the information from a sensor embedded smart home and a smartwatch worn by the patient to determine his/her current situation. In case of an emergency, the application sends the alert message on the caretaker's mobile phones.

The proposed framework and application differ from other approaches in the sense that it captures the effects of handling the conflicts in the form of inconsistencies and incompleteness using contextual defeasible logic. It provides a semantic knowledge-based sound inferencing system to stipulate distributed reasoning in multi-agent systems using mapping rules and the consistency of the system is attained in order to achieve the desired goals correctly and efficiently. The prototypical implementation using Thingspeak, IFTTT, Twilio, and MATLAB platforms reveals the validity and correctness of the proposed formalism.

7 Conclusion

In this paper, we proposed ontology-driven contextual defeasible reasoning based multi-agent formalism to handle inconsistent information in a highly decentralized environment. We develop a simple case study considering two different domains (contexts) ontologies with the prototypal implementation of the system and show the validity, correctness, and verify the non-conflicting contextual information. In future work, we plan to develop a state-of-the-art mobile application for smartphone users to tailor the specific need and verify the correctness properties of the system suitable for the smart spaces.

References

1. Al-Anbaki, N.S., Obeid, N., Sabri, K.E.: A defeasible logic-based framework for contextualizing deployed applications. Work **10**(9) (2019)
2. Antoniou, G.: A nonmonotonic rule system using ontologies. In: RuleML, vol. 60 (2002)
3. Antoniou, G., Bikakis, A., Karamolegou, A., Papachristodoulou, N.: A context-aware meeting alert using semantic web and rule technology-preliminary report. In: Semantic Web Technology for Ubiquitous and Mobile Applications (SWUMA 2006), vol. 23 (2006)
4. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. **2**(4), 263–277 (2007)
5. Benslimane, D., Arara, A., Falquet, G., Maamar, Z., Thiran, P., Gargouri, F.: Contextual ontologies. In: Yakhno, T., Neuhold, E.J. (eds.) ADVIS 2006. LNCS, vol. 4243, pp. 168–176. Springer, Heidelberg (2006). https://doi.org/10.1007/11890393_18
6. Bikakis, A., Antoniou, G.: Defeasible contextual reasoning with arguments in ambient intelligence. IEEE Trans. Knowl. Data Eng. **22**(11), 1492–1506 (2010)
7. Borgida, A., Serafini, L.: Distributed description logics: directed domain correspondences in federated information sources. In: Meersman, R., Tari, Z. (eds.) OTM 2002. LNCS, vol. 2519, pp. 36–53. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36124-3_3
8. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, vol. 7, pp. 385–390. AAAI Press (2007)

9. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001). https://doi.org/10.1007/s007790170019
10. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Towards diagnosing inconsistency in nonmonotonic multi-context systems. Logic-Based Interpretation, p. 9 (2009)
11. Eiter, T., Fink, M., Schüller, P., Weinzierl, A.: Finding explanations of inconsistency in multi-context systems. Artif. Intell. **216**, 233–274 (2014)
12. Esposito, A., Tarricone, L., Zappatore, M., Catarinucci, L., Colella, R.: A framework for context-aware home-health monitoring. Int. J. Auton. Adapt. Commun. Syst. **3**(1), 75–91 (2010)
13. Grau, B.C., Parsia, B., Sirin, E.: Working with multiple ontologies on the semantic web. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 620–634. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30475-3_43
14. Gruber, T.R., et al.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993)
15. Haque, H.M.U., Khan, S.U.: A context-aware reasoning framework for heterogeneous systems. In: 2018 International Conference on Advancements in Computational Sciences (ICACS), pp. 1–9. IEEE (2018)
16. Haque, H.M.U., Khan, S.U., Hussain, I.: Semantic knowledge transformation for context-aware heterogeneous formalisms. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **10**(12), 664–670 (2019)
17. Mahfooz Ul Haque, H., Rakib, A., Uddin, I.: Modelling and reasoning about context-aware agents over heterogeneous knowledge sources. In: Cong Vinh, P., Tuan Anh, L., Loan, N.T.T., Vongdoiwang Siricharoen, W. (eds.) ICCASA 2016. LNICST, vol. 193, pp. 1–11. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56357-2_1
18. Hong, M.W., Cho, D.J.: Ontology context model for context-aware learning service in ubiquitous learning environments. Int. J. Comput. **2**(3), 172–178 (2008)
19. Moawad, A., Bikakis, A., Caire, P., Nain, G., Le Traon, Y.: R-core: a rule-based contextual reasoning platform for AmI. In: Joint Proceedings of the 7th International Rule Challenge, the Special Track on Human Language Technology and the 3rd RuleML Doctoral Consortium Hosted at the 8th International Symposium on Rules (RuleML2013) (2013)
20. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology (2001)
21. Rakib, A., Haque, H.M.U.: A logic for context-aware non-monotonic reasoning agents. In: Gelbukh, A., Espinoza, F.C., Galicia-Haro, S.N. (eds.) MICAI 2014, Part I. LNCS (LNAI), vol. 8856, pp. 453–471. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13647-9_41
22. Rakib, A., Ul Haque, H.M., Faruqui, R.U.: A temporal description logic for resource-bounded rule-based context-aware agents. In: Vinh, P.C., Alagar, V., Vassev, E., Khare, A. (eds.) ICCASA 2013. LNICST, vol. 128, pp. 3–14. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05939-6_1
23. Sowa, J.F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Cole Publishing Co., Pacific Grove (2000)
24. Sun, D., Toh, K.C., Yuan, Y., Zhao, X.Y.: SDPNAL+: a Matlab software for semidefinite programming with bound constraints (version 1.0). Optim. Methods Softw. **35**(1), 87–115 (2020)
25. Uddin, I., Rakib, A., Haque, H.M.U., Vinh, P.C.: Modeling and reasoning about preference-based context-aware agents over heterogeneous knowledge sources. Mob. Netw. Appl. **23**(1), 13–26 (2018). https://doi.org/10.1007/s11036-017-0899-5