

Modelling and Verification of Context-Aware Intelligent Assistive Formalism

Shahid Yousaf^{1,*}, Hafiz Mahfooz Ul Haque², Abbas Khalid¹, Muhammad Adnan Hashmi³ and Eraj Khan¹

¹Department of Computer Science & IT, The University of Lahore, Lahore, 54000, Pakistan

²Department of Software Engineering, The University of Lahore, Lahore, 54000, Pakistan

³Higher Colleges of Technology, Abu Dhabi, 25026, UAE

*Corresponding Author: Shahid Yousaf. Email: shahid.yousaf@cs.uol.edu.pk

Received: 25 August 2021; Accepted: 14 October 2021

Abstract: Recent years have witnessed the expeditious evolution of intelligent smart devices and autonomous software technologies with the expanded domains of computing from workplaces to smart computing in everyday routine life activities. This trend has been rapidly advancing towards the new generation of systems where smart devices play vital roles in acting intelligently on behalf of the users. Context-awareness has emerged from the pervasive computing paradigm. Context-aware systems have the ability to acquire contextual information from the surrounding environment autonomously, perform reasoning on it, and then adapt their behaviors accordingly. With the proliferation of context-aware systems and smart sensors, real-time monitoring of environmental situations (context) has become quite trivial. However, it is often challenging because the imperfect nature of context can cause the inconsistent behavior of the system. In this paper, we propose a context-aware intelligent decision support formalism to assist cognitively impaired people in managing their routine life activities. For this, we present a semantic knowledge-based framework to contextualize the information from the environment using the protégé ontology editor and Semantic Web Rule Language (SWRL) rules. The set of contextualized information and the set of rules acquired from the ontology can be used to model Context-aware Multi-Agent Systems (CMAS) in order to autonomously plan all activities of the users and notify users to act accordingly. To illustrate the use of the proposed formalism, we model a case study of Mild Cognitive Impaired (MCI) patients using Colored Petri Nets (CPN) to show the reasoning process on how the context-aware agents collaboratively plan activities on the user's behalf and validate the correctness properties of the system.

Keywords: Context-awareness; multi-agents; colored petri net; ontology



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

In recent years, interest and demand for smart systems and applications have been rapidly evolving. With their remarkable upsurge, smart devices and autonomous applications are becoming more complex, optimized, sophisticated, and efficient. In this arena, numerous devices are impeccably integrated via portable or embedded devices by providing readily available services to facilitate users at anytime and anywhere. Using these devices and applications, everyday users spend much time and effort to accomplish their tasks and autonomously exchange information among different devices and/or platforms for complex problem solving and make their lives much more comfortable, relaxed, and reliable, but device-dependent. Context-aware computing has been considered as one of the most emerging paradigms which provide freedom from the bondage of traditional computing systems to adaptive and highly dynamic computing. Context-aware systems often run in a highly decentralized environment and exhibit complex adaptive behaviors without or with very limited human interaction.

Literature has revealed a significant amount of work in intelligent assistive formalisms and applications with the incorporation of context-awareness in different domains such as safety-critical systems [1,2], disaster recovery systems [3,4], traffic management systems [5,6], different healthcare solutions [7,8], etc. Among others, Mild Cognitive Impairment (MCI) is one of the most commonly spread diseases in elderly age people. These people become helpless and are bound to stay at home due to MCI disease. In mild cognitive impairment, people gradually start facing a feeling of slight and unnoticeable decline in memory. Cognitive impairment mostly starts around the age of 65 or above. According to United Nations (UN) Population Division, the population of elderly people is feared to increase by more than 2 billion by 2050 [9]. Commonly, these people face difficulties in remembering, decision-making, learning new things, and planning & scheduling daily life activities. They require the assistance of a person/device to drive their regular activities like taking meals, medication, attending social events (wedding ceremony, birthday party, etc.), as well as important events like official meetings and doctor's appointments, etc. These people can perform their routine life activities easily with little help like notification and alarm. They feel more comfortable and live like normal people if their smart devices such as smartphones or Personal Digital Assistants (PDA) assist them to plan and schedule their routine life activities. Intelligent assisted applications have the great potential to improve the quality of life by rendering a wide range of personalized services and providing users with easier and more efficient ways to exchange information and help interact with other people. For example; in [10], authors have developed a cognitive assistant to help cognitively impaired people. For this, they proposed a system called COACH to assist dementia patients by reminding them about different daily life activities. Such types of assistive application systems are used to manage routine life activities of a person where context is the primary component since it provides the requisite information about the current status/activities of the person under consideration. The authors in [11], presented an Autominder system, which is a personal robotic assistant for elderly people to help them in dealing with memory impairment. This system issues reminders to notify cognitively impaired people in managing and planning their daily life activities such as taking medicines on time and engaging them in social and family activities. Although a significant effort has been made by the researchers to build intelligent assistive applications in different domains, however, the semantic knowledge-based context-aware intelligent assistive formalism for cognitive impairment is still in its infancy and needs to be rigorously studied for the exploitation of its innate potential. In this paper, we propose a context-aware intelligent assistive formalism for cognitively impaired people to schedule and plan their daily life activities without or with minimal human assistance. This formalism consists of three layers named as sensors layer, semantic layer, and contextual reasoning layer. In the sensors layer, we assume that the raw data is fetched through the context-aware agents from the environment and this data is contextualized

using the ontology. In the semantic layer, the desired contextualized information is obtained from a semantic knowledge source using SPARQL queries and the set of facts and the set of rules are extracted using semantic knowledge transformation mechanism. In the reasoning layer, a context-aware multi-agent system collaboratively plans the activities to infer the desired results and notify patients to take appropriate actions accordingly. To illustrate the use of the proposed formalism, we model a case study of cognitively impaired patients in Colored Petri Nets to analyze the behavior of the system and validate the correctness properties.

The rest of the paper is structured as follows: Section 2 describes preliminaries and related work. Section 3 presents contextualizing semantic knowledge ontology to represent the context in the machine-processable form. In Section 4, we propose a context-aware multi-agent reasoning formalism for cognitive impairments. Section 5 presents a comprehensive case study for MCI patients along with its execution strategies and evaluates the correctness properties and we finally conclude in Section 6.

2 Preliminaries and Related Work

2.1 Contextual Modelling and Reasoning

Literature has revealed several definitions of contexts so far. Among others, Dey et al. [12] define context as “*Any information that can be used to characterize the situation of entity is called context information*”. Context information has been classified in different forms along with its different properties such as (a) User context (identity, preference, activity, location) (b) Device context (processor speed, screen size, location) (c) Application context (version, availability) (d) Physical environment context (illumination, humidity) (e) Resource context (availability, size, type) (f) Network context (minimum speed, maximum speed) (g) Location context (where it is subsumed) (h) Activity context (start time, end time, actor) to name a few. In general, there are four major categories to acquire contextual information from different sources such as applying sensors, network information, device status, and browsing user profiles. The contextual instances can be classified further in two dimensions, i.e., external (physical) and internal (logical). In an external context, data is manipulated by the hardware sensor, while in an internal context, data is manipulated by the interaction of the user (user goals, tasks, emotions, etc.). Context-aware systems and applications are typically designed using smart and intelligent devices with embedded sensors such as smartphones, smartwatches, intelligent assistants, etc. Intelligent assistants have usually customized devices embedded with sensors and/or components, which acquire contextual information from the environment through the physical sensors or logical sensors, perform reasoning based on the existing knowledge, and assist people in their respective activities.

In context-aware computing, context modelling has been considered to be the most promising approach for contextualizing information. As a context is captured from the different perspectives, a uniform context representation approach is very essential. Context Modelling refers to defining and storing context data in a machine-processable form [13]. Literature has revealed various context modelling approaches including Key-Value Models [14,15], Markup Scheme Models [16,17], Graphical Model [18,19], Logic-Based Modelling [20,21], Object-Oriented Models [22–24] and Ontology-Based Models [25–27]. Among others, the ontology-based context-modelling approach has been advocated as the most promising approach due to its modular structuring mechanism, reusability, independence of its own identity [28–32]. In this paper, we choose the ontology-driven context modelling approach for the proposed intelligent assistive formalism.

Contextual reasoning plays a vital role in the domain of context-aware systems to infer the contextualized information from the existing knowledge sources and lets the system to take decisions

intelligently and interactively in a real-time environment. Contextual reasoning mechanism is used to make decisions dynamically whenever there is a change in the user's context. The use of contextual reasoning using Multi-Agent Systems (MAS) increases the cognitive ability of the context-aware agents to sense the environment, perform reasoning and take actions accordingly in order to achieve the desired goals. Literature has revealed various reasoning techniques such as BDI (belief-desire-intention) [33], case-based reasoning [34], rule-based reasoning [35], etc. Among others, rule-based reasoning has been considered as one of the most simple and efficient reasoning techniques to perform reasoning when developing expert decision-making systems.

2.2 Related Work

Numerous efforts have been made in the development of context-aware intelligent assistive systems with the incorporation of context modelling and reasoning. Intelligent assistants facilitate the users by assisting them in the achievement of their short-term as well as long-term goals as described earlier. An intelligent assistant is a software agent that helps the user by performing tasks on its behalf [36]. There has been a growing interest in intelligent assistants for a variety of systems and applications. In [37], authors have presented a framework called DynaMoL for managing the follow-up of a colorectal cancer patient. This is a dynamic decision analysis framework that used the Bayesian learning model to learn from the large medical database. Besides, in [38], yet another intelligent assistant called Intelligent Therapy Assistant (ITA) has been developed to assist the therapists in configuring the patient's treatments in the Guttman Neuro Personal Trainer (GNPT) platform. This ITA used data mining techniques on large stored data to treat every patient depending on his/her profile. Similarly in [39], the authors presented an automated cognitive assistant system about transportation for mentally retarded people suffering from traumatic brain injuries so that they can easily use public transport. This system is implemented with a GPRS-enabled cell phone, Bluetooth sensor beacon (separate from the cell phone and placed in user's pocket or purse), and inference engine (a software that learns from the user's history and behavior to reason about users' transportation routines). Primarily, this system is developed to provide a route starting point to destination with minimum intervention of the user. Moreover, a prototype of a smart home for cognitively impaired people has been presented in [40] which assists patients in their initiatives and helps them in performing their daily life activities. Apart from that, in [41], the authors have shown a large number of assistive living tools, and systems for the assistance of elderly people suffering from various mental impairments. A variety of theoretical as well as practical systems have been presented in this paper to help people suffering from dementia and other cognitive impairments like OutCare [42], wireless health care system [43], etc.

3 Contextualizing Semantic Knowledge Ontology

Semantic web technologies [44] are considered to be the most optimal choice for context modelling and knowledge representation in intelligent and dynamic environments due to their uniform representation and reasoning capability. Ontology is the explicit formal specifications of different terms/concepts, which are used to share information in a particular domain. Ontology has different generations such as Web Ontology Language (OWL) namely OWL 1 and OWL 2 [45]. OWL-1 has three variants [46] namely OWL-Lite, OWL-DL, and OWL-Full. Similarly, OWL 2 has three sub-languages [47] such as OWL 2 EL, OWL 2 QL, and OWL 2 RL. We have chosen OWL 2 RL since it provides scalable reasoning without compromising the expressive power. Further, it enables implementation in polynomial time with rule-based technologies. Semantic Web Rule Language (SWRL) can be used to write complex rules. It enhances the expressivity, completeness, and allows users to write user-defined

rules for characterizing the complex systems. To model the domain, we have opted the ontology-based context modelling approach to contextualize the context-aware intelligent assistive formalism. For this purpose, we develop the ontology and construct OWL 2 RL and SWRL rules. The ontology consists of 66 classes in total, where *Person*, *UserAgent*, *PlannerAgent* and *Contextual_Agents* are the superclasses; the rest are their child classes. A few example of object and data properties are: *HasRegID*, *ActivityLocation*, *HasPlannedActivity*, *ActivityDate*, *ActivityStartTime* and *ActivityEndTime*. Rules are formulated from ontology classes, data and object properties. The proposed context model provides the structure of contextualized information to all nominated agents. All types of contexts which can be possibly used by intelligent assistant/agents to assist MCI Patients are defined by categorizing them accordingly, i.e., we have divided context into the following categories: social context, entertainment context, environmental context, exercise-related context, food-related context, in-home everyday context, medical context, office-related context, to-do thing context, saloon related context, location context and traveling context. A fragment of the ontology is shown in Fig. 1. The Fig. 2 depicts the class hierarchies of the MCI Patient’s Context.

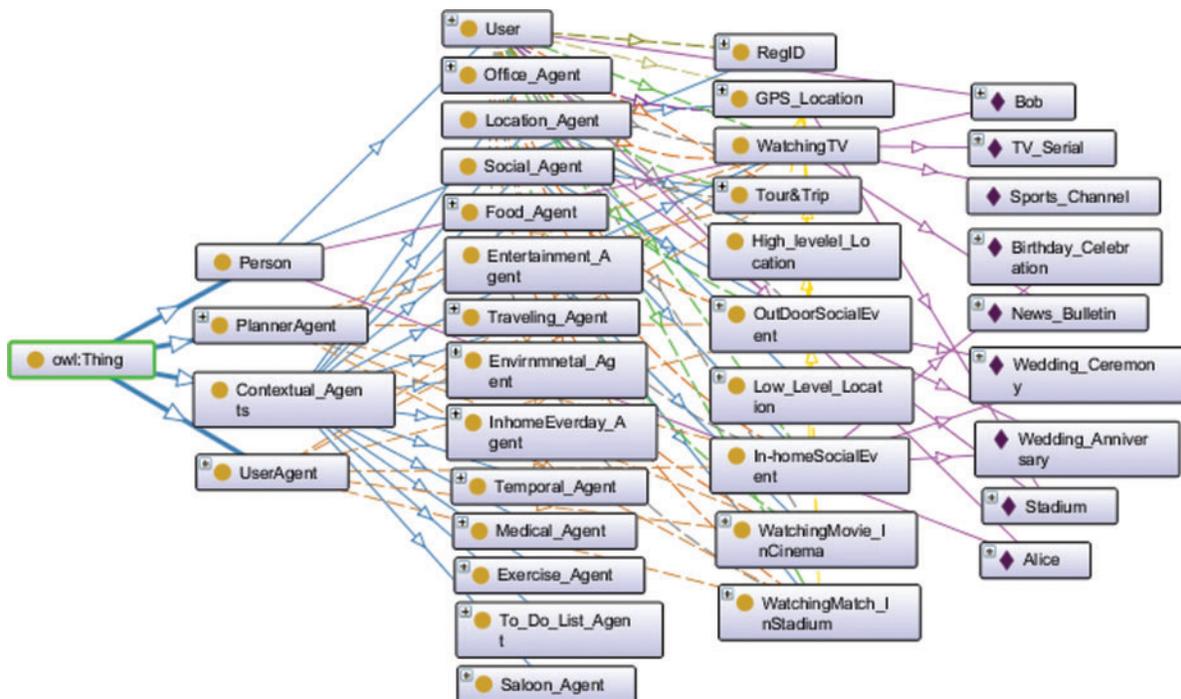


Figure 1: A fragment of the ontology of intelligent assistant for MCI patients

To execute the MCI domain in the Protégé ontology editor, the SPARQL query language is used to check and analyze the results, and it is specifically designed for the semantic web ontologies to evaluate extensible value testing [48]. For this, we executed a number of queries using the SPARQL query plugin in the Protégé ontology editor and some of the example queries are given in Tab. 1.

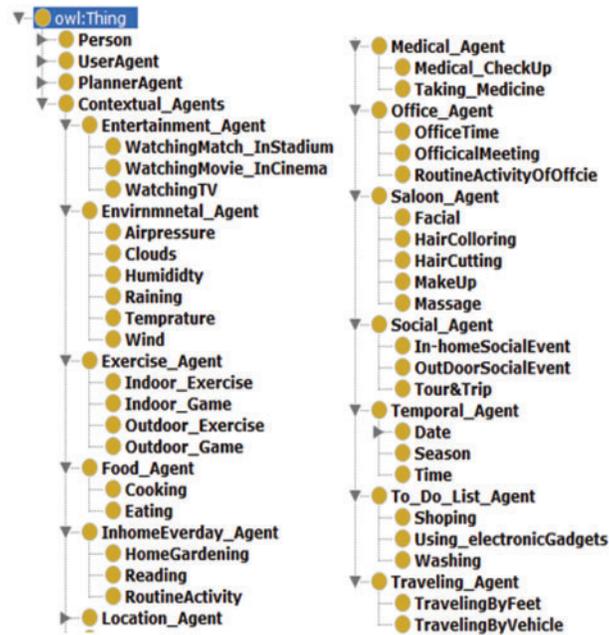


Figure 2: Class hierarchies of the MCI patients' context

Table 1: Some example SPARQL queries

Sr. No	Queries
1	<pre> SELECT ?UserName ?Reg_ID WHERE { ?UserName mci:hasRegID ?Reg_ID. } </pre>
2	<pre> SELECT ?UserName ?Activity WHERE { ?UserName mci: HasWriteActivity ?Activity.} </pre>
3	<pre> SELECT ?UserName ?Activity ?Date WHERE { ?UserName mci: HasWriteActivity ?Activity. ?Activity mci:ActivityDate ?Date. } </pre>
4	<pre> SELECT ?UserName ?WrittenActivity ?Date ?StartTime ?EndTime ?ActivityLocation WHERE { ?UserName mci:HasWriteActivity ? WrittenActivity. ? WrittenActivity mci:ActivityDate ?Date. ? WrittenActivity mci:ActivityStartTime ?StartTime. ? WrittenActivity mci:ActivityEndTime ?EndTime. ? WrittenActivity mci:ActivityLocation ? ActivityLocation. } </pre>

(Continued)

Table 1: Continued

Sr. No	Queries
5	<pre> SELECT ?UserName ?PlannedActivity ?Date ?StartTime ?EndTime ?ActivityLocation WHERE { ?UserName mci:HasPlannedActivity ?PlannedActivity. ?PlannedActivity mci:ActivityDate ?Date. ?PlannedActivity mci:ActivityStartTime ?StartTime. ?PlannedActivity mci:ActivityEndTime ?EndTime. ?PlannedActivity mci:ActivityLocation ?ActivityLocation } </pre>

4 Context-Aware Multi-Agent Reasoning Formalism

In this section, we present a semantic knowledge-based intelligent assistive formalism using a context-aware Multi-Agent System. This formalism consists of a set of agents where each agent in the system has a set of facts (contextualized information), a set of rules, and a reasoning strategy. These agents acquire the contextualized information from the corresponding environment which is obtained from the ontology. Further, they perform reasoning using a set of rules; share the contextualized information among different agents in order to infer the desired results, and adapt behaviors accordingly. In context-aware multi-agent settings, knowledge is shared among different agents. Moreover, ontology provides declarative semantics to assist intelligent agents in contextual reasoning. As shown in Fig. 3, the architecture of context-aware intelligent assistive formalism consists of three different layers. The lower layer (sensors layer) interacts with the environment of the user. Further, we assume that the user is carrying the smartphone with him/her, and raw contextual information (user's activities) is acquired with the help of different sensors of the smartphone from the physical environment. This raw context information is interpreted on the intermediate layer (semantic layer). This interpretation process of context is called context modelling and OWL ontologies are developed to represent the context in the machine-processable form in a standardized format. No doubt, context modelling is an integral part of context-aware assistive applications since the non-intrusive behavior of the assistive application is not possible without the proper understanding of the user's context. The upper layer is the contextual reasoning layer consisting of two modules, i.e., contextual agents and context-aware rule-based reasoning.

The proposed system consists of a set of context-aware agents, i.e., Social Agent, Food Agent, Medical Agent, etc. These context-aware agents receive machine-processable contextualized information from the semantic layer and provide the contextualized information so obtained, to the User Agent and Planner Agent. A user agent interacts with the user to receive and send the information (activities and their schedule). A Planner agent plans all activities of the user with the coordination of other contextual agents (depending on the user's activities) and then sends it to the user agent. Then, the user agent shows this plan to the user. As the user acknowledges, the Planner agent saves this plan and sends an updated plan to the corresponding agents. The contextual agents continuously synchronize themselves with the environmental context through sensors/agents and monitor the activities of the user continuously and update the Planner Agent after certain intervals of time. If the user deviates from his/her scheduled plan, the Planner agent sends alerts to the user agent. Besides, the second component of the third layer is context-aware rule-based reasoning. A set of generic rules is designed to reason about the user's context with the help of contextual agents. The rules are defined

in the SWRL format, its general representation is as following

$$\textit{Antecedent} \Rightarrow \textit{Consequent} \quad (1)$$

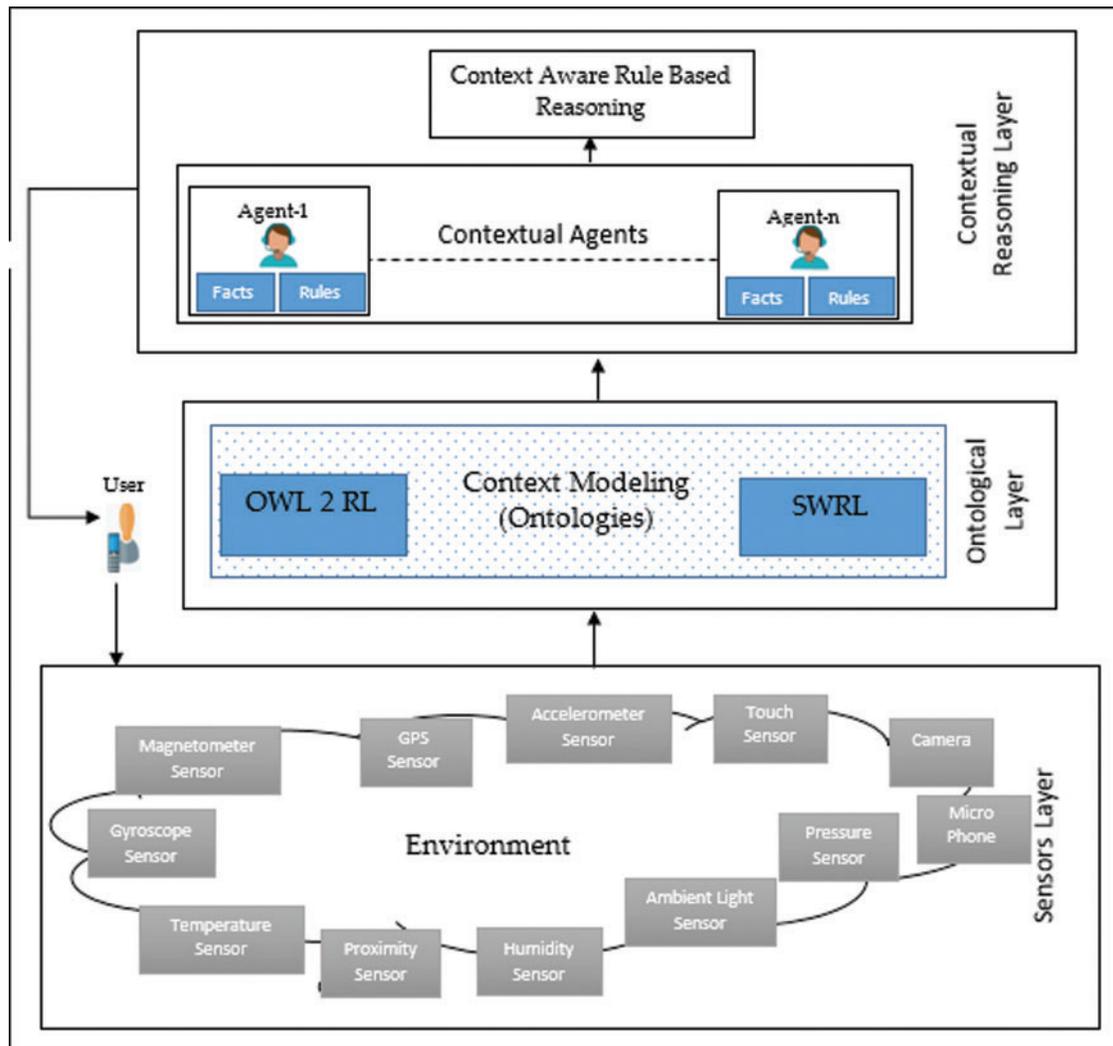


Figure 3: Architecture of context-aware intelligent assistive formalism

An antecedent is a body and the consequent is the head of the rule. Whenever the specified condition holds true in the antecedent part then ultimately the condition in the consequent part must be held true. The antecedent and consequent both are the conjunctions of atoms which are written in the form of a_1, a_2, \dots, a_n . The question mark ($?X$), followed a variable is used to denote the variables, for example:

$$\textit{Person} (?u), \textit{RegID} (?id), \textit{hasRegID} (?u, ?id) \rightarrow \textit{User} (?u) \quad (2)$$

Here *Person*, *RegID*, *hasRegID* and *User* are the rule atoms, $(?u)$ and $(?id)$, are the variable and after assigning values to variable this rule becomes in the following form:

$$Person(Bob), RegID(R01), hasRegID(Bob, R01) \rightarrow User(Bob) \quad (3)$$

The above rule will be interpreted as “there is a person “Bob” and registration ID “R01”, and Bob has the registration ID “R01” therefore he is a registered user”.

In this work, a context-aware rule-based reasoning system collaboratively plans activities and makes decisions dynamically whenever there is the faintest change in the user’s context. Usage of contextual reasoning increases the cognitive ability of the context-aware agents in achieving their desired goals, of course.

5 Modelling and Reasoning MCI Patient’s Case Study Using CMAS

To demonstrate the usability and practical effectiveness of the proposed formalism, we present a case study of MCI patients. This case study has been designed for the people who normally face difficulty in recalling their activities on daily basis. This system would assist the patients in planning and managing their daily routine life activities. We develop a scenario for a cognitively impaired person, namely Bob, who usually forgets to attend important office meetings, family gatherings, doctor’s appointments, taking medicines, etc. He frequently forgets to execute any plan he has scheduled earlier. If someone reminds him, however, he can recall the plan and handle the situation accordingly. Using the proposed system, he can use his intelligent assistant to schedule and recall all the activities without any human intervention. The functionality of the proposed system is illustrated with the help of a case scenario from the MCI example system. For instance, every Sunday morning, Bob usually schedules his activities for a whole week and he can query the planned schedule on his intelligent assistant anytime by writing activity name or by checking the activity date, time, and venue. Bob’s planned activities for the whole week are shown in [Tab. 2](#). Bob writes all these activities on the intelligent assistant to schedule these activities. The intelligent agent requests confirmation about Bob’s schedule and Bob confirms this schedule. When this schedule is executed, Bob follows the planned schedule. However, sometimes Bob forgets to plan any activity which was supposed to be planned earlier or activity might be changed due to some reason. In this case, if Bob fails to do the activity in time, the assistant will remind him to perform this activity as the changes were not made. For example, on Monday, Bob must have to leave for the hotel Holiday Inn at 18:20 as he has to join the wedding ceremony of his friend at 19:00. The hotel is at the drive of 40 min from his home. At 18:10, the intelligent assistant will remind Bob to get ready as he has to leave for the hotel at 18:20 to attend the wedding. If Bob does not leave the office till 18:20, he will be notified again or he can cancel this scheduled activity if not interested to attend this event. Similarly, if some other tasks occur during the week, the intelligent assistant will merge the task with the existing activities; on Wednesday, Bob’s wife said to him that she wants to go for shopping, for example. He asks his intelligent assistant to set the time of shopping on Saturday at 12:00, the intelligent assistant replied to him you have already planned an outing activity on Saturday at 10:00, and you will not be able to come back home before 12:00. Therefore, there is a clash between these two activities. The intelligent assistant suggests to him different schedules and Bob selects the option of shopping to go on Saturday at 19:00. Similarly, for next week’s plan, Bob again writes the whole week’s activities and asks his intelligent assistant to schedule them. Such activities are set by Planner agents, and the agents can revise their beliefs based on the existing set of facts and the set of inferred rules. The activities are shown in [Tab. 2](#).

Table 2: Tentative schedule of Bob's weekly activities

Activity/ Context	Date	Start time	End time	Expected location	Expected travel time
Wedding ceremony	13-09-2021	19:00	21:00	Holiday_Inn_Hotel	40
TV Serial	14-09-2021	20:00	20:50	Home	N/A
Birthday party	15-09-2021	19:30	21:30	Home	N/A
Cricket match	17-09-2021	18:30	24:30	Stadium	15
Tour	19-09-2021	10:00	20:00	Hills View Park	120

To model the case study with the proposed framework, a total of 15 different agents are used. They work in an orchestrated way to serve the user in a better way. We used distributed rule-based reasoning method for context reasoning. A multi-agent framework with contextual reasoning has been proposed, where a program of each agent consists of a set of rules, contextual information and reasoning mechanism. These context-aware agents acquire contextual data and set of extracted rules from ontologies. We have written the SWRL rules for all 15 agents but due to space constraints, we present a few rules for only 5 agents to depict the working flow of the system in [Tab. 3](#). These rules strengthen the reasoning ability of OWL ontologies and make them more expressive. These SWRL rules are merged with OWL ontologies by using SWRLTab which is available in the Protégé tool. This tool is most commonly used for ontology development and a variety of plugins are available in this tool. Further, it also allows to import new plugins. The SWRLTab of Protégé-OWL allows the creation of new SWRL rules and manipulations on these by reading and writing existing SWRL rules. Some chosen example rules are shown in [Tab. 3](#). Here R_{ij} refers to the j^{th} rule of the i^{th} agent.

Table 3: A few examples rules**Agent 1: User Agent**

Initial Facts: Person(Bob), RegID(R01), hasRegID(Bob, R01), User(Bob), UsersActivity(Bob, Wedding_Ceremony), ActivityDate(Wedding_Ceremony, 30-06-2021), ActivityStartTime(Wedding_Ceremony, 19:00), ActivityEndTime(Wedding_Ceremony, 21:00) ActivityLocation(Wedding_Ceremony, Holiday_Inn_Hotel)

(Continued)

Table 3: Continued

R_{11} : Person(?u), RegID(?id), hasRegID(?u, ?id) \rightarrow User(?u)
 R_{12} : User(?u) \rightarrow Ask(1,5, SelectActivity(?act))
 R_{13} : Tell(5, 1, OutDoorSocialEvent(?act)) \rightarrow , OutDoorSocialEvent(?act)
 R_{14} : User(?u), OutDoorSocialEvent(?act) \rightarrow UsersActivity(?u, ?act)
 R_{15} : Tell(3, 1, ActivityDate(?act, ?date)) \rightarrow ActivityDate(?act, ?date)
 R_{16} : Tell(3, 1, ActivityStartTime (?act, ?Stime)) \rightarrow ActivityStartTime (?act, ?Stime)
 R_{17} : Tell(3, 1, ActivityEndTime(?act, ?Etime)) \rightarrow ActivityEndTime(?act, ?Etime)
 R_{18} : Tell(4, 1, ActivityLocation(?act, ?loc)) \rightarrow ActivityLocation(?act, ?loc)
 R_{19} : User(?u), In-homeSocialEvent(?act), ActivityDate(?act, ?date), ActivityStartTime(?act, ?Stime), ActivityEndTime(?act, ?Etime), ActivityLocation(?act, ?loc) \rightarrow hasScheduledActiviy(?u, ?act)
 R_{110} : User(?u), hasScheduledActiviy(?u, ?act) \rightarrow Ask(1, 2, PlanActivityFor(?u, ?act))
 R_{111} : Tell(2, 1, HasPlannedActivity(?u, ?act)) \rightarrow HasPlannedActivity(?u, ?act)

Agent 2: Planner Agent

Initial Facts: PlanActivityFor(Bob, Wedding_Ceremony), PlanActivity(Wedding_Ceremony)
 HasPlannedActivity(Bob, Wedding_Ceremony)

R_{21} : Ask(1, 2, PlanActivityFor(?u, ?act)) \rightarrow PlanActivityFor(?u, ?act)
 R_{22} : Tell(3, 2, ActivityDate(?act, ?date)) \rightarrow ActivityDate(?act, ?date)
 R_{23} : Tell(3, 2, ActivityStartTime(?act, ?Stime)) \rightarrow ActivityStartTime(?act, ?Stime)
 R_{24} : Tell(3, 2, ActivityEndTime(?act, ?Etime)) \rightarrow ActivityEndTime(?act, ?Etime)
 R_{25} : Tell(4, 2, ActivityLocation(?act, ?loc)) \rightarrow ActivityLocation(?act, ?loc)
 R_{26} : Tell(5, 2, In-homeSocialEvent(?act)) \rightarrow In-homeSocialEvent(?act)
 R_{27} : In-homeSocialEvent(?act), ActivityDate(?act, ?date), ActivityStartTime(?act, ?Stime), ActivityEndTime(?act, ?Etime), ActivityLocation(?act, ?loc), PlanActivity(?act) \rightarrow HasPlannedActivity(?u, ?act)
 R_{28} : HasPlannedActivity(?u, ?act) \rightarrow Tell(2, 1, HasPlannedActivity(?u, ?act))

Agent 3: Temporal Agent

Initial Facts: ActivityDate(Wedding_Ceremony, 15-03-2021),
 ActivityStartTime(Wedding_Ceremony, 19:00), ActivityEndTime (Wedding_Ceremony, 21:00),
 CurrentTime(17:33)

R_{31} : Tell(1, 3, UsersActivity(?u, ?act)) \rightarrow UsersActivity(?u, ?act)
 R_{32} : UsersActivity(?u, ?Act), SetDate(?act, ?date), Calendar(?date) \rightarrow ActivityDate(?act, ?date)
 R_{33} : ActivityDate(?act, ?date) \rightarrow Tell(3, 1, ActivityDate(?act, ?date))
 R_{34} : UsersActivity(?u, ?Act), SetActivityStartTime(?act, ?Stime), Time(?Stime) \rightarrow ActivityStartTime(?act, ?Stime)
 R_{35} : ActivityStartTime(?act, ?Stime) \rightarrow Tell(3, 1, ActivityStartTime(?act, ?Stime))
 R_{36} : UsersActivity(?u, ?Act), SetActivityEndTime(?act, ?Etime), Time(?Etime) \rightarrow ActivityEndTime(?act, ?Etime)
 R_{37} : ActivityEndTime(?act, ?Etime) \rightarrow Tell(3, 1, ActivityEndTime(?act, ?Etime))
 R_{38} : ActivityEndTime(?act, ?Etime) \rightarrow Tell(3, 2, ActivityEndTime(?act, ?Etime))

(Continued)

Table 3: Continued**Agent 4: Location Agent**

Initial Facts: ActivityLocation(Wedding_Ceremony, Holiday_Inn_Hotel),
CurrentLocationOfUser(Bob, Home)

R_{41} : Tell(1, 4, UsersActivity(?u, ?act)) \rightarrow UsersActivity(?u, ?act)
 R_{42} : Tell(2, 4, HasPlannedActivity(?u, ?act)) \rightarrow HasPlannedActivity(?u, ?act)
 R_{43} : UsersActivity(?u, ?Act), SetLocation(?act, ?loc), GPS(?loc) \rightarrow ActivityLocation(?act, ?loc)
 R_{44} : ActivityLocation(?act, ?loc) \rightarrow Tell(4, 1, ActivityLocation(?act, ?loc))
 R_{45} : ActivityLocation(?act, ?loc) \rightarrow Tell(4, 2, ActivityLocation(?act, ?loc))
 R_{46} : Ask(2, 4, SendCurentLocationofUser(?u, ?C_Loc)) \rightarrow SendCurentLocationofUser(?u, ?C_Loc)
 R_{47} : User(?u) GPS(?C_Loc), GetCurrentLocationOfUser(?u, ?C_Loc) \rightarrow CurrentLocationOfUser(?u, ?C_Loc)
 R_{48} : CurrentLocationOfUser(?u, ?C_Loc) \rightarrow Tell(4, 2, CurrentLocationOfUser(?u, ?C_Loc))

Agent 5: Social Agent

Initial Facts: OutDoorSocialEvent(Wedding_Ceremony),
SelectedActivityName(Wedding_Ceremony, "OutDoorSocialEvent"),
In-homeSocialEvent(Birthday_Party)

R_{51} : Ask(1, 5, SelectActivity(?act)) \rightarrow SelectActivity(?act)
 R_{52} : Activity(?act), SelectActivity(?act), SelectedActivityName(?act, "In-homeSocialEvent") \rightarrow In-homeSocialEvent(?act)
 R_{53} : In-homeSocialEvent(?act) \rightarrow Tell(5, 1, In-homeSocialEvent(?act))

We assume that data is gathered with the help of the Smartphone sensors and then OWL ontologies are developed to present the context in uniform and standard format. But OWL 2 does not express all types of relations and some complex rules are also not expressible in OWL 2. Therefore, we write user-defined complex rules using SWRL to address this issue with the combination of OWL 2 RL.

Tab. 4 illustrates the total number of queries that are executed on the system. These queries consist of the *correctly answered queries* (successfully executed) on the systems, *incorrect queries* which are executed but give no output, and *not working queries* (rejected) that contain errors. The proposed system obtains 82% *Precision*, 78% *Recall*, and *F-Measure* 80%.

6 Formal Modelling and Validation of MCI Case Study

Colored Petri Nets (CPN) is a graphical modelling language that is the union of Petri nets [49] and functional programming languages [50]. This union makes the CPN models suitable for modelling complex systems. Further, the thrust of CPN language is on the modelling and validation of software systems having concurrency, communication, and synchronization [51]. CPN models are capable to be simulated or verified with formal methods (i.e., state-space analysis and invariant analysis). CPNs allow for state space-based exploration and analysis. State-space analysis can be used to detect system properties such as the absence of deadlocks. CPN model consists of Colors set, Places, Transitions, Arcs, Color functions, Guard functions, Arc Inscriptions, and Initializing function. Places, transitions,

and arcs are the basic Petri net components. We have created an untimed CPN model (as shown in Fig. 4) for the proposed model and performed simulation-based analysis which vividly demonstrated that the proposed model is logically correct.

Table 4: Results of the queries on the model

Total number of executed queries	50
Number of correctly answered queries	39
Number of not working queries	8
Number of incorrect queries	3
<i>Precision</i>	82%
<i>Recall</i>	78%
<i>F-Measure</i>	80%

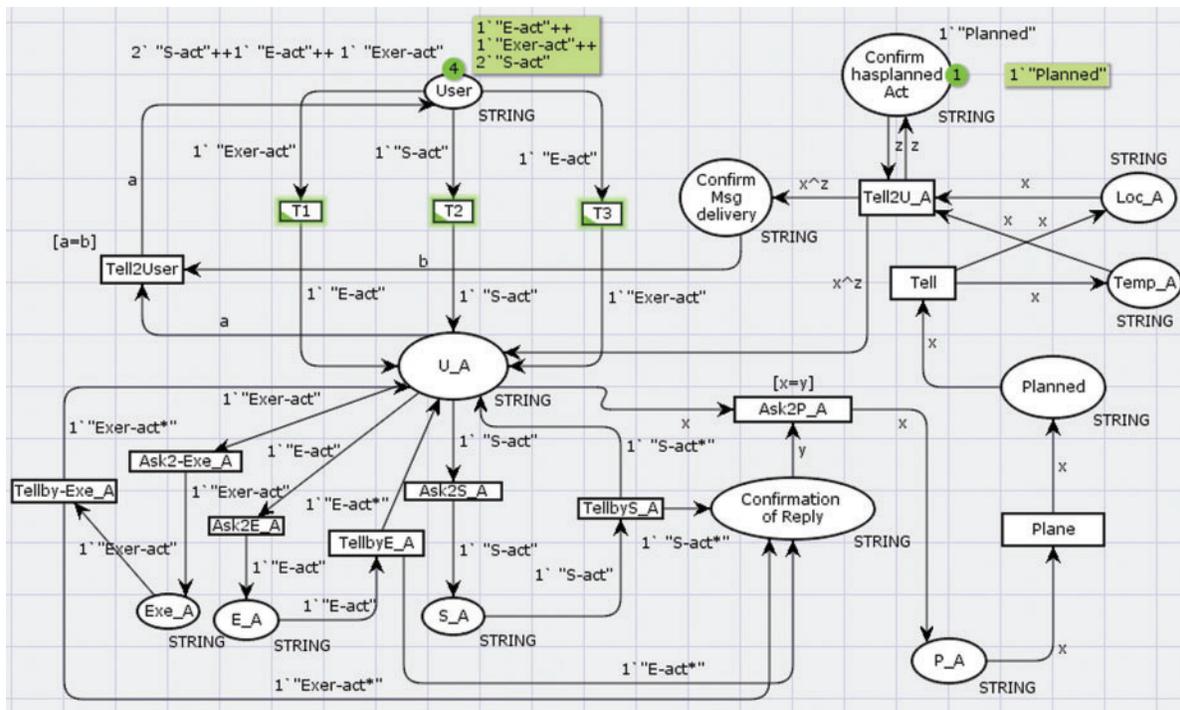


Figure 4: CPN model for coordination process of multi-agents

We illustrate the use of the proposed system using the CPN model to formally model check and verify the correctness properties. For this, we initially *model the* user’s activities along with the plans. The syntactic representation of the model shows the number of activities along with each token. In the system, *User* place has a total of 4 tokens which corresponds to their four activities. Two of them are social activities which are represented as 2 “*S-act*”. The rest two activities are (a) exercise-related activity 1 “*Exer-act*” and (b) entertainment activity 1 “*E-act*”. According to the structure of the proposed multi-agent coordination process, the user agent (*U_A*) in the CPN model receives the activity request from the user. As the system starts execution, *U_A* chooses the corresponding

contextual agent to plan the activities. For instance, when a user requests the exercise agent (E_A) to plan the exercise activity, the U_A uses special communication primitives $Ask2-Exe-A$ and $Tellby-Exe_A$ to exchange the contextual information among different agents in term firing rules instances and systems transitions. $Ask2-Exe-A$ means U_A asking Exe_A to plan the exercise activity in the gymnasium whereas $Tellby-Exe_A$ means Exe_A acknowledges to U_A . As a result of a successful transition, the token $I` "Exer-act$ is passed to U_A to get acknowledgment and $ConfirmationOfReply$. This transition has a guard value $[x = b]$ which ensures that the right activity is selected according to the U_A . The planner agent P_A plans the activity considering the location agent (Loc_A) along with the temporal agent ($Temp_A$), as a result, a tag ("*planned*") will be added to every confirmed token which shows that this activity has been planned successfully. In the same fashion, the rest of the activities will be planned concurrently.

To validate the CPN-based multi-agent communication model, state-space tool can be used to analyze behavioral properties such as boundedness, liveness, and fairness property. Fig. 5 shows the experimental results of the proposed model, as it can be seen that state space generated a total of 3645 nodes and 12312 arcs. This simulation took 1 s and its status is full. These states show that the given model will terminate in a given amount of time and there are no such conditions that it will execute infinitely. Furthermore, it indicates that there is a Strongly Connected Component (SCC) in the graph and there is no restriction on the movement of nodes.

```

Statistics
-----

State Space
Nodes: 3645
Arcs: 12312
Secs: 1
Status: Full

Scc Graph
Nodes: 3645
Arcs: 12312
Secs: 0

```

Figure 5: Statistics of O-graph

Boundedness in model checking is an efficient technique that can be used to find bugs in the system design. If the upper bound of any place in the proposed CPN model is beyond the expectation then it means that the system will not render the expected results. Fig. 6 shows that it prevents the overflow of the buffer. The highest upper bound of any state is 4 in the proposed model. Moreover, there are a total of four activities that have been executed on the CPN model. It means that no such condition exists in our system which puts it in the buffer overflow situation. In the CPN model, the home property is used to represent the scenario where the system gets back to the initial state. In the proposed model, the user sends a request to the system for the planning of different contextual activities. The system, in turn, will return these activities to the user after planning. Therefore, catering of home property in the proposed model is very natural. Fig. 7 shows that 3645 node is home marking, which is also the last node of state space. This node is against the *user* place which is the start state of the proposed model. It means the model is working according to the requirements and the user will receive the planned activities in return.

Boundedness Properties

Best Integer Bounds

	Upper	Lower
Agents_Communication_Model'Confirm_Msg_delivery 1	4	0
Agents_Communication_Model'Confirm_hasplanned_Act 1	1	1
Agents_Communication_Model'Confirmation_of_Reply 1	4	0
Agents_Communication_Model'E_A 1	1	0
Agents_Communication_Model'Exe_A 1	1	0
Agents_Communication_Model'Loc_A 1	4	0
Agents_Communication_Model'P_A 1	4	0
Agents_Communication_Model'Planned 1	4	0
Agents_Communication_Model'S_A 1	2	0
Agents_Communication_Model'Temp_A 1	4	0
Agents_Communication_Model'U_A 1	4	0
Agents_Communication_Model'User 1	4	0

Figure 6: Upper and lower integer bound of places in the model

Verification of the Liveness properties implies the absence of the deadlock in the system. Fig. 7 shows that the dead marking is only 3645, the last node of the state space. It means that there is no other dead state which would have occurred during the currency of the simulation. So, the system will perform all functionalities. Similarly, *dead transition instances* are none, which means that no such transition exists in the model which is dead and puts the system in a deadlock situation. The fairness property deals with the “halting” of the system. A firing sequence is said to be unconditionally (globally) fair if it is finite. Fig. 7 shows that there is no infinite occurrence of sequences in the proposed model and it will halt in the given timeframe. The result of home properties, liveness properties, and fairness properties is shown in Fig. 7.

```
Home Properties
-----
Home Markings
  [3645]

Liveness Properties
-----
Dead Markings
  [3645]

Dead Transition Instances
  None

Live Transition Instances
  None

Fairness Properties
-----
  No infinite occurrence sequences.
```

Figure 7: Result of home properties, liveness properties and fairness properties

7 Conclusion

In this paper, we have proposed a semantic knowledge-based context-aware intelligent assistive formalism for cognitively impaired people. This is a three-layered model where context-aware agents extract the contextualized information from the ontology, perform reasoning on it and automatically schedule the user activities based on his convenience and feedback. We used HermiT reasoner and SPARQL queries to check the consistency and correctness of the ontology domain. We model the case study using Colored Petri Nets to analyze the execution behavior and validate the correctness properties of the system. In future work, we will implement the system physically in a real-time environment using a comprehensive case study and develop a smartphone application. The state-of-the-art system would assist the user to autonomously plan the whole week's activities at their own ease and convenience.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. E. Bardram and N. Nørskov, "A context-aware patient safety system for the operating room," in *Proc. of the 10th Int. Conf. on Ubiquitous Computing*, New York, USA, pp. 272–281, 2008.

- [2] B. Frada, A. Zaslavsky and N. Arora, "Context-aware mobile agents for decision-making support in healthcare emergency applications," in *Int. Workshop on Context Modeling and Decision Support, CEUR Workshop Proc.*, Fribourg, Switzerland, 2005.
- [3] D. I. Inan, G. Beydoun and B. Pradhan, "Developing a decision support system for disaster management: Case study of an Indonesia volcano eruption," *International Journal of Disaster Risk Reduction*, vol. 31, pp. 711–721, 2018.
- [4] F. Burstein, P. D. Haghighi and A. Zaslavsky, "Context-aware mobile medical emergency management decision support system for safe transportation," in *Decision Support: An Examination of the DSS Discipline*, New York, USA, pp. 163–181, 2011.
- [5] Z. Rehena and M. Janssen, "Towards a framework for context-aware intelligent traffic management system in smart cities," in *Companion Proc. of the Web Conf. 2018*, Republic and Canton of Geneva, CHE, pp. 893–898, 2018.
- [6] D. Goel, S. Chaudhury and H. Ghosh, "An IoT approach for context-aware smart traffic management using ontology," in *Proc. of the Int. Conf. on Web Intelligence*, New York, USA, pp. 42–49, 2017.
- [7] B. Manate, V. I. Munteanu, T. F. Fortis and P. T. Moore, "An intelligent context-aware decision-support system oriented towards healthcare support," in *2014 Eighth Int. Conf. on Complex, Intelligent and Software Intensive Systems*, Birmingham, UK, pp. 386–391, 2014.
- [8] M. Oliveira, C. Hairon, O. Andrade, R. Moura, C. Sicotte *et al.*, "A context-aware framework for health care governance decision-making systems: A model based on the Brazilian digital TV," in *2010 IEEE Int. Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Montreal, QC, Canada, pp. 1–6, 2010.
- [9] V. Fuster, "Changing demographics: A new approach to global health care due to the aging population," *Journal of the American College of Cardiology*, vol. 69, no. 24, pp. 3002–3005, 2017.
- [10] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie *et al.*, "A decision-theoretic approach to task assistance for persons with dementia," in *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence*, San Francisco, CA, USA, pp. 1293–1299, 2005.
- [11] M. E. Pollack, L. Brown, D. Colbry, C. E. McCarthy, C. Orosz *et al.*, "Autominder: An intelligent cognitive orthotic system for people with memory impairment," *Robotics and Autonomous Systems*, vol. 44, no. 3, pp. 273–282, 2003.
- [12] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *Int. Symposium on Handheld and Ubiquitous Computing*, Berlin, Heidelberg, pp. 304–307, 1999.
- [13] M. Baldauf, S. Dustdar and F. Rosenberg, "A survey on context-aware systems," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007.
- [14] H. Li and J. Wang, "Application architecture for ambient intelligence systems based on context ontology modeling," in *2011 Int. Conf. on Internet Technology and Applications*, Wuhan, China, pp. 1–4, 2011.
- [15] T. Stang, "Service Interoperability in Ubiquitous Computing Environments," *Ph.D. dissertation*, Ludwig-Maximilians-University of Munich, Germany, 2003.
- [16] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications. *Ph.D. dissertation*, Georgia Institute of Technology, USA, 2000.
- [17] L. Capra, W. Emmerich and C. Mascolo, "Reflective middleware solutions for context-aware applications," in *Proc. of the Third Int. Conf. on Metalevel Architectures and Separation of Crosscutting Concerns*, Berlin, Heidelberg, pp. 126–133, 2001.
- [18] S. Boutemedjet and D. Ziou, "A graphical model for context-aware visual content recommendation," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 52–62, 2007.
- [19] K. Henriksen, J. Indulska and A. Rakotonirainy, "Modeling context information in pervasive computing systems," in *Int. Conf. on Pervasive Computing*, Berlin, Heidelberg, pp. 167–180, 2002.
- [20] J. McCarthy, "Notes on formalizing contexts," in *Proc. of the Thirteenth Int. Joint Conf. on Artificial Intelligence*, San Mateo, California, pp. 555–560, 1993.

- [21] M. Miraoui, S. El-etriby, A. Z. Abed and C. Tadj, "A logic based context modeling and context-aware services adaptation for a smart office," *International Journal Of Advanced Studies in Computers, Science And Engineering*, Gothenburg, vol. 5, no. 1, pp. 1–6, 2016.
- [22] F. Dobslaw, A. Larsson, T. Kanter and J. Walters, "An object-oriented model in support of context-aware mobile applications," in *Int. Conf. on Mobile Wireless Middleware, Operating Systems, and Applications*, Berlin, Heidelberg, pp. 205–220, 2010.
- [23] B. Bouzy and T. Cazenave, "Using the object oriented paradigm to model context in computer go," in *Proc. of the First Int. and Interdisciplinary Conf. on Modeling and Using Context*, Rio de Janeiro, Brazil, pp. 279–289, 1997.
- [24] J. Bhogal and P. Moore, "Towards object-oriented context modeling: Object-oriented relational database data storage," in *28th Int. Conf. on Advanced Information Networking and Applications Workshops*, Victoria, BC, Canada, pp. 542–547, 2014.
- [25] X. H. Wang, D. Q. Zhang, T. Gu and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Proc. of the Second IEEE Annual Conf. on Pervasive Computing and Communications Workshops*, USA, pp. 18–22, 2004.
- [26] P. Moore and B. Hu, "A context framework with ontology for personalised and cooperative mobile learning," in *Computer Supported Cooperative Work in Design III*, Berlin, Heidelberg, pp. 727–738, 2007.
- [27] N. Xu, W. Zhang, H. Yang and X. Xing, "CACOnt: A ontology-based model for context modeling and reasoning," *Applied Mechanics and Materials*, vol. 347, pp. 2304–2310, 2013.
- [28] I. Horrocks, P. F. Patel-Schneider and F. V. Harmelen, "From SHIQ and RDF to OWL: The making of a Web ontology language," *Journal of Web Semantics*, vol. 1, no. 1, pp. 7–26, 2003.
- [29] H. Chen, F. Perich, T. Finin and A. Joshi, "SOUPA: standard ontology for ubiquitous and pervasive applications," in *the First Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*, MOBIQUITOUS, pp. 258–267, 2004.
- [30] D. Zhang, T. Gu and X. Wang, "Enabling context-aware smart home with semantic technology," *International Journal of Human-Friendly Welfare Robotic Systems*, vol. 6, no. 4, pp. 12–20, 2005.
- [31] H. Chen, T. Finin and A. Joshi, "Semantic web in the context broker architecture," in *Proc. of the Second IEEE Annual Conf. on Pervasive Computing and Communications*, Orlando, Florida, USA, 2004.
- [32] T. Gu, X. H. Wang, H. K. Pung and D. Q. Zhang, "An ontology-based context model in intelligent environments," in *Proc. of Communication Networks and Distributed Systems Modeling and Simulation Conf.*, San Diego, California, USA, 2004.
- [33] L. Padgham, D. Scerri, G. Jayatilleke and S. Hickmott, "Integrating BDI reasoning into agent-based modeling and simulation," in *Proc. of the 2011 Winter Simulation Conf. (WSC)*, Phoenix, AZ, USA, pp. 345–356, 2011.
- [34] D. Guessoum, M. Miraoui and C. Tadj, "Contextual case-based reasoning applied to a mobile device," *International Journal of Pervasive Computing and Communications*, vol. 13, no. 3, pp. 282–299, 2017.
- [35] A. Bikakis and G. Antoniou, "Rule-based contextual reasoning in ambient intelligence," in *Int. Workshop on Rules and Rule Markup Languages for the Semantic Web*, Berlin, Heidelberg, Germany, pp. 74–88, 2010.
- [36] G. Czibula, A. M. Guran, I. G. Czibula and G. S. Cojocar, "IPA-An intelligent personal assistant agent for task performance support," in *2009 IEEE 5th Int. Conf. on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, pp. 31–34, 2009.
- [37] T. Y. Leong and C. Cao, "Modelling medical decisions in dynamol: A new general framework of dynamic decision analysis," in *Proc. of the Ninth Conf. on Medical Informatics*, Amsterdam: North Holland, pp. 483–487, 1998.
- [38] J. Solana, C. Cáceres, A. G. Molina, P. Chausa, E. Opisso *et al.*, "Intelligent therapy assistant (ITA) for cognitive rehabilitation in patients with acquired brain injury," *BMC Medical Informatics and Decision Making*, vol. 14, no. 1, pp. 1–13, 2014.
- [39] D. J. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic *et al.*, "Opportunity knocks: A system to provide cognitive assistance with transportation services," in *Int. Conf. on Ubiquitous Computing*, Berlin, Heidelberg, Germany, pp. 433–450, 2004.

- [40] J. Bauchet, D. Vergnes, S. Giroux and H. Pigot, "A pervasive cognitive assistant for smart homes," in *Proc. of the Int. Conf. on Aging, Disability and Independence*, St. Petersburg, Florida, USA, vol. 228, 2006.
- [41] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2012.
- [42] J. Wan, C. Byrne, G. M. O'Hare and M. J. Grady, "Orange alerts: lessons from an outdoor case study," in *2011 5th Int. Conf. on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, IEEE, pp. 446–451, 2011.
- [43] C. C. Lin, M. J. Chiu, C. C. Hsiao, R. G. Lee and Y. S. Tsai, "Wireless health care service system for elderly with dementia," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 696–704, 2006.
- [44] J. Domingue, D. Fensel and J. A. Hendler, "Introduction to the semantic web technologies," in *Handbook of Semantic Web Technologies*, Springer, Berlin, Heidelberg, 2011.
- [45] G. Antoniou and F. V. Harmelen, "Web ontology language: Owl," in *Handbook on Ontologies*, Springer, Berlin, Heidelberg, pp. 67–92, 2004.
- [46] D. L. McGuinness and F. V. Harmelen, "OWL web ontology language overview," *W3C Recommendation*, vol. 10, 2004.
- [47] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue *et al.*, "OWL 2 web ontology language profiles," *W3C Recommendation*, vol. 27, 2009.
- [48] E. Prud'hommeaux and A. Seaborne, "SPARQL query language for RDF," *W3C Recommendation*, 2008.
- [49] W. Reisig, "*Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*," 1st ed., Berlin, Germany, Springer, 2013.
- [50] R. Milner and M. Tofte, "*The Definition of Standard ML*," MIT Press, Cambridge, MA, USA, 1990.
- [51] W. M. P. VanderAalst, C. Stahl and M. Westergaard, "Strategies for modeling complex processes using colored petri nets," in *Transactions on Petri Nets and other Models of Concurrency vii*, Springer, Berlin, Heidelberg, pp. 6–55, 2013.