# Context-Aware Access Control with Imprecise Context Characterization for Cloud-Based Data Resources*

A. S. M. Kayes[a,*], Wenny Rahayu[a], Tharam Dillon[a], Elizabeth Chang[b], and Jun Han[c]

[a] *La Trobe University, Melbourne, Australia*

[b] *University of New South Wales, Canberra, Australia*

[c] *Swinburne University of Technology, Melbourne, Australia*

**Abstract**

Computing technologies are increasingly dynamic and ubiquitous in everyday life nowadays. Context information plays a crucial role in such dynamically changing environments and the different types of contextual conditions bring new challenges to context-sensitive access control. This information mostly can be derived from the crisp sets. For example, we can utilize a crisp set to derive a patient and nurse are co-located in the general ward of the hospital or not. Some of the context information characterizations cannot be made using crisp sets, however, they are equally important in order to make access control decisions. Towards this end, this article proposes an approach to *Context-Aware Access Control using Fuzzy logic (FCAAC)* for data and information resources. We introduce a *formal context model* to represent the fuzzy and other contextual conditions. We also introduce a *formal policy model* to specify the policies by utilizing these conditions. Using our formal approach, we combine the fuzzy model with an ontology-based approach that captures such contextual conditions and incorporates them into the policies, utilizing the ontology languages and the fuzzy logic-based reasoning. We introduce a *unified data ontology* and its associated *mapping ontology* in terms of facilitating access control to *cloud-based*

*An earlier version of this paper has been published in the Proceedings of the 25th International Conference on Cooperative Information Systems (CoopIS 2017) [1].

*Corresponding author - email address: **a.kayes@latrobe.edu.au** (A. S. M. Kayes)

*data resources*. We justify the feasibility of our approach by demonstrating the practicality through a *prototype implementation*, several healthcare *case studies* and a *usability study*. Finally, we demonstrate an *experimental evaluation* in terms of query response time. The experiment results demonstrate the satisfactory performance of our proposed FCAAC approach.

*Keywords:* Context-aware access control, Cloud-based data resources, Fuzzy contextual conditions, Context model, Fuzzy reasoning model, Policy model, Ontology

---

## 1. Introduction

Accessing data and information resources from multiple cloud environments has increasingly become challenging nowadays due to the homogeneous and heterogeneous nature of data sources. Efficiently controlling the users' access to such cloud-based data resources from multiple sources is one of the main challenges. Over the years, access control mechanisms have shifted from a fixed desktop environment to dynamic environments (e.g., pervasive, cloud and mobile computing environments) [2]. Due to this paradigm shift, the role of dynamically changing *context information* has gained great importance for *context-specific decision making*, where users need seamless access to data and information resources from anywhere and at anytime fashion, even when they are on the move. In terms of *context-aware access control* systems [3, 4], context means information about the state of a relevant entity or the state of a relevant relationship between entities, where an entity can be a user, resource or their environments.

The gathering of relevant context information as the major underlying mechanism in today's dynamic world is crucial and thus demanding for further studies on many aspects of access control to data and information resources. Among the significant factors, an access controller needs to be *context-aware* by incorporating the different types of dynamic context information. In particular, there is a need for an even seamless integration of *precise fuzzy conditions* and *other*

*relevant contextual conditions* subsequently with access control policies, in order to manage an access to information resources at different granularity levels. Consider a healthcare scenario where a doctor Jane is needed to access the medical records of a patient Bob, who is currently admitted to a hospital due to a severe heart attack. In general, only the emergency doctors have access to all of the medical records for patients who are admitted for emergency treatment, including their medical history and personal health records. However, Jane, while not being an emergency doctor, can play the *emergency doctor role* from the *emergency ward* of the hospital when Bob's health status is "*high critical*" and consequently can access *all of his medical records* from multiple sources to save his life. Therefore, an access controller needs to consider such kinds of fuzzy facts/conditions when making access control decisions. In particular, there is a need to quantify the fuzzy conditions more precisely (e.g., Bob's health status is "*high critical*" with "*criticality level 95%*"). On the one hand, context-specific access control to data and information resources together with such conditions can provide an extra level of safety for patients. On the other hand, accessing data and resources from multiple sources can provide an extra level of flexibility for healthcare users (e.g., emergency doctors, nurses, researchers) in such emergency medical situations.

The different access control solutions have been historically been applied to support emergency situations mentioned earlier. Among them the traditional, spatial and temporal Role-Based Access Control (RBAC) approaches [5, 6] are the fundamental and widely accepted solutions to support hospital users and patients. In RBAC, the roles (e.g., doctors, nurses and so on) are organized in static hierarchies and users (e.g., Jane) are authorized to play such roles for exercising organizational functions. However, some of these roles cannot be organized in the same way in static hierarchies. These roles can be called as dynamic or contextual roles (e.g., the emergency doctor). Users need to satisfy the relevant contextual conditions (e.g., a fuzzy condition that a patient Bob's current health status is "95% critical") to grant such dynamic roles and access necessary data resources accordingly. In order to manage the emergency

situations, these fuzzy contextual conditions can be effectively derived from the IoT devices (e.g., a patient's body sensors data) and the smart spaces. In such smart environments, all static and dynamic roles that are organized in static hierarchies might be associated with large processing overheads and administrative costs. Based on the RBAC models, the access control policies can either be too restrictive and deny Jane from accessing emergency health records of Bob, or allow Jane's access too liberal and potentially compromising security and privacy. Thus, the basic RBAC approaches where the user roles are organized in static role hierarchies are not adequate to address this problem. Instead of RBAC roles in static hierarchies, there is a growing need to exploit the dynamic contextual conditions (fuzzy and other), in order to reduce the burden of manual specification of all static and dynamic roles. However, without the benefits of modeling fuzzy and other contextual conditions, we have to manually model all possible roles (e.g., doctors, emergency doctors) in static hierarchies and specify the associated access control policies. In order to find applicable policies, subsequently, we have to search in large policy rule-base. Since we have to deal with emergency situations, it is really important to ensure the speed of the responses to access the relevant data resources.

### *1.1. Background*

Context-aware access control is a mechanism to determine whether a user's request to limit the access permissions to data and information resources based on the dynamically changing contextual conditions (e.g., the interpersonal relationship between patient and nurse is "assigned nurse", the patient's health status is "66% normal" with "criticality level 34%", etc.). In the literature, there has been a significant amount of research work in developing context-aware access control approaches.

A number of such access control approaches consider the *spatial information* (e.g., [5]), the *temporal information* (e.g., [6]), the *event-driven information* such as surgery in progress (e.g., [7]), and other *environment context information* such as the range of IP addresses (e.g., [8, 9, 10]), as contextual conditions when

making access control decisions. In this context, our group has a successful track record in developing context-aware access control systems by considering a wide variety of contextual conditions: the *general context information* about the state of the users, resources and their environments [3, 11], the *relationship context information* utilizing the process of inferring implicit knowledge [12], and the *purpose-oriented situation information* based on the currently available context information [4, 13]. We also propose a context-aware access control policy model in our earlier research [14], incorporating these relevant contextual conditions into the access control policies. These contextual conditions usually derive from the crisp sets (e.g., the doctor is located in the "emergency ward" of the hospital or "not"), and these traditional approaches are not adequate to deal with imprecise context characterization. However, there are other types of contextual conditions which only can be derived from the fuzzy sets by utilizing the low-level fuzzy facts, and they are equally important in order to make access control decisions at different granularity levels.

Other than the above-mentioned traditional context-sensitive access control approaches, several research works consider the use of fuzzy conditions (e.g., computing resource owners' trust degrees [15], quantifying risks [16], measuring trust levels [17], calculating user-permission strengths [18]) for making access control decisions. However, these approaches are not context-aware and robust enough to integrate both the fuzzy conditions and other dynamic contextual conditions with access control policies for context-specific decision making. Using successful experience from our group's earlier research on fuzzy linguistic representations for capturing the semantics of warehoused data [19], we develop our fuzzy model that is used in this article to deal with imprecise context characterization.

Looking at the existing context-sensitive access control approaches, these solutions extensively have been used to access data and information resources from centralized sources (e.g., [20, 21]). They do not provide adequate functionality to access data and resources from distributed environments (e.g., from multiple could sources). In the literature, different data integration techniques

have been developed over the last few decades to collate data from multiple sources (e.g., [22, 23, 24]). However, these techniques are still limited in order to provide the "granted" or "denied" access control decision to the users. Currently, the cloud-based Internet of things (IoTs) paradigm [25] seeks a new form of context-sensitive access control approach for understanding mechanisms of controlling data and information resources from different cloud and Big Data sources [26]. Our group has also established some cloud-based models to address several issues: user-side quality of service management [27], fuzzy inference for measuring trust values of the cloud providers [28], applications and clients' interaction with the cloud by lowering costs along with supporting high security [29]. Over the last few years, several fog-based access control approaches also have been proposed (e.g., [30, 31, 32, 33]) to reduce the processing overheads and administrative costs involved in managing and accessing cloud-based data and services (e.g., [34, 35, 36]). These fog nodes usually provide intermediary computation and networking services between the end-users and the traditional cloud data servers. However, these fog-based access control approaches are not adequate to facilitate context-sensitive access control to data and resources from distributed cloud environments.

### *1.2. Research Issues*

In order to achieve *context-awareness* and integrate the different types of fuzzy and other contextual conditions into the access control processes in the distributed environments, the following research problems need to be addressed.

(RP1) How to derive precise contextual conditions from imprecise fuzzy facts for context-specific decision making?

(RP2) How to integrate these derived fuzzy conditions and other relevant contextual conditions with access control policies to facilitate context-specific access to information resources at different granularity levels?

(RP3) How to interact between the fog and the cloud in building context-aware access control applications?

*1.3. Contributions*

The above-identified gap in the literature suggests that there is still a need for a new form of context-aware access control approach that can further limit the applicability of the available access permissions to cloud-based data resources, integrating both the fuzzy facts and other contextual conditions together with access control policies for context-specific decision making. A first version of our context-aware access control approach with imprecise context characterization was introduced in [1]. However, this earlier access control approach and its associated policy model is still limited in controlling context-sensitive access control to data and resources from multiple cloud sources. This article extends our initial approach to Context-Aware Access Control using Fuzzy logic (FCAAC) [1] to improve context-sensitive access control decisions in the cloud environments. The contributions and significant extensions are listed as follows.

(CE1) ***Research Motivation:*** We have extended our emergency healthcare scenario and demonstrated the CAAC requirements in developing applications for cloud and fog computing environments. We have now included the detailed analysis of the scenario and its associated research challenges. The analysis certainly helps to build a foundation for the development of software prototypes for cloud-based data resources.

(CE2) ***Formal FCAAC Approach:*** We have introduced a new form of context-sensitive access control approach, named Context-Aware Access Control using Fuzzy logic (FCAAC), addressing the following aspects (see (i) to (iii)). We have extended our initial FCAAC approach with respect to access control operations that are performed on multiple sources, specifically including a new aspect (see (iv)).

(i) ***Context Representation Model:*** We have presented a formal analysis of the fuzziness of (imprecise) context information. We have introduced a formal context model to represent the fuzzy and normal contextual conditions (low-level contextual conditions) from the raw context facts.

(ii) ***Context Reasoning Model:*** The context representation model is extended with user-defined reasoning rules to derive high-level contextual conditions. In particular, the context reasoning model uses the context representation model to infer richer contextual conditions at different abstraction levels based on the low-level contextual conditions.

(iii) ***Policy Model:*** We have presented a formal analysis of the context-specific access control decision making by taking into account the low-level and high-level contextual conditions (fuzzy and normal).

(iv) ***Unified Data and Mapping Models:*** We have presented a formal analysis of a unified data model and its associated policy model in terms of facilitating context-sensitive access control to data and information resources from multiple sources.

(CE3) ***Ontology-based FCAAC Approach:*** Using our formal approach, we have introduced an ontology-based approach to model the relevant fuzzy and normal contextual conditions, and consequently model the context-sensitive access control policies, incorporating the relevant conditions into the access control processes. We have made significant extension to our ontologies and presented important ontological definitions and examples. We have now introduced a general data ontology and its associated mapping ontology in relation to apply our FCAAC approach in the cloud and fog environments. Whereas our earlier approach is not applicable to access data and resources from multiple cloud sources.

(CE4) ***Evaluation of Our Approach:*** Other than the above two main contributions, we have justified the feasibility of our approach by demonstrating the following factors:

(i) ***Software Prototype:*** We have developed a software prototype of the FCAAC approach that can assist application developers in

rapid prototyping. We have now included the detailed components of our prototype.

(ii) ***Case Studies:*** We have presented several case studies from the healthcare domain which demonstrate the practicality of the proposed approach and provide the basis for developing context-sensitive access control applications in the fog and cloud environments.

(iii) ***Usability Study:*** We have now carried out a usability study by supporting a user interface and demonstrating a walkthrough of our FCAAC proposal in a real setup with real users.

(iv) ***Performance Evaluation:*** We have conducted two sets of experiment in a healthcare environment and evaluated the applicability of our FCAAC approach by means of response time. The experiment results have shown the satisfactory performance of our proposed context-sensitive access control approach.

(v) ***Comparative Analysis:*** In addition to the prototype, case studies, usability study and performance evaluation, we have now presented a comparative analysis of the existing access control approaches. We have also included the fog and cloud-based access control approaches. The comparative assessment has shown that our FCAAC approach offers a range of new benefits for context-sensitive access control in the fog and cloud computing environments.

### *1.4. Outline*

The rest of this article is organized as follows. Section 2 presents a healthcare scenario to motivate our work. Section 3 introduces our formal context-sensitive access control approach, named FCAAC, including the context representation and reasoning model and its associated policy model. It also includes a unified data model and its associated mapping model in terms of facilitating access control to cloud-based data resources from multiple sources. Section 4 introduces

an ontology-based development platform for our proposed FCAAC approach. Section 5 demonstrates the practicality of our approach, including a software prototype, several healthcare case studies, a usability study and an experimental evaluation in terms of query response time. Section 6 briefly presents the related work and a comparative analysis of our FCAAC approach with respect to existing access control approaches. Finally, Section 7 concludes the paper and outlines future research directions.

## 2. Research Motivation and General Requirements

This section presents an extended application scenario from our earlier research [3, 1]. We first analyse the need for the incorporation of fuzzy and normal contextual conditions in the access control process, illustrating an access control to multiple data sources for different types of users within the distributed systems. In addition, we identify the general requirements of developing a new access control approach for the cloud environments by integrating both the fuzzy conditions and other contextual conditions together with access control policies. We use suitable examples from this scenario throughout the article to explain the concepts of our approach.

### *2.1. Application Scenario*

In this section, we consider an extended healthcare scenario from our earlier research [3, 1].

- *A patient Bob who is currently admitted in the emergency department of the hospital due to a severe heart attack. Jane, who is a hospital doctor, is required to access the necessary medical records of Bob from multiple sources to treat him and save his life from such life-threatening situation. After getting emergency treatment, Bob is shifted to the general ward of the hospital and assigned a registered nurse Mary to monitor his health status.*

*2.2. Scenario Analysis*

In this section, we analyse the application scenario to capture the technical challenges to control access to data and resources from multiple sources.

In general, the emergency doctors, including a patient's treating physician, can access all the necessary health records of patients, such as the medical records, past medical history and private medical records. However, Jane, while not being an emergency doctor, is able to access the necessary medical records by playing the emergency doctor role from the emergency ward of the hospital when Bob's health status is "*high or 95% critical*". When the context changes (e.g., Bob's health status becomes "*66% normal*"), a decision on a further access request by Jane to Bob's emergency medical records may need to change accordingly (e.g., an access permission should be *denied*). That is, Jane is only authorized to play the hospital doctor role, and consequently can access Bob's normal medical records when his health condition is "*66% normal*".

Normally, a registered nurse, who is assigned to look after a patient (or a group of patients), is able to access the daily medical records during her ward duty time and when she is present in the general ward where the patient is located. However, in the mentioned emergency scenario, Mary is able to access Bob's medical records when she is co-located with Jane, who is currently treating Bob by playing the emergency doctor role, and only when his health status is "*high critical*". When the context changes (e.g., Mary *leaves the emergency department* or *outside of duty time*), a decision on a further access request by Mary to Bob's medical records may need to change accordingly (e.g., an access permission should be *denied*). That is, Mary, by playing the assigned nurse role, is only able to access Bob's daily medical records during her ward duty time and only when they both are co-located in the general ward of the hospital.

The different types of conditions are involved in this scenario, e.g., the location and request time of a nurse, the health status of a patient, etc. Therefore, an access controller needs to exploit such conditions directly or indirectly when making access control decisions. The normal conditions such as the location and request time can be obtained directly from the context sources. The health

status is not able to obtain directly but can be derived from the available low-level data such as the body temperature and pulse rate. As such, it is necessary to further process the retrieved low-level imprecise data or fuzzy facts automatically to precisely obtain the relevant results (e.g., the health status is "*66% normal*" with "*criticality level 34%*"). In order to limit the access permissions to resources exploiting such fuzzy and other conditions is both a strength and a challenge.

Nowadays, a large number of data and information resources have been produced as a result of the abundance of cloud and Big Data sources from multiple environments. This data abundance creates new opportunities and also raises new challenges to develop new form of access control mechanisms along with data integration and mapping capabilities. In the above application scenario, Jane needs to access different types of medical records (e.g., Bob's health records) from multiple data sources in different contexts. That is, such data and resources may come from centralized or distributed sources. Thus, the access controller needs to deal with multiple data sets within different organizations (e.g., medical, insurance and diagnosis companies). In order to access data from multiple sources, there is a need to build a unified data model to specify generic concepts and map all the local data sources to the generic unified schema. In this fashion, on the one hand, we can reduce the number of access control policies. On the other hand, we can overcome the processing overheads.

In the light of above-pointed observations, Figure 1 illustrates the relationship chain among the users, cloud servers and an intermediary fog node. The relationship chain connects the users to the cloud sources. In order to support such mapping to different medical databases (e.g., health records, insurance records, diagnosis records) and access data and resources subsequently, there is a need for a new form of context-sensitive access control application in today's fog and cloud environments. In particular, an intermediary fog node is required to facilitate access control to cloud-based data resources from multiple sources.

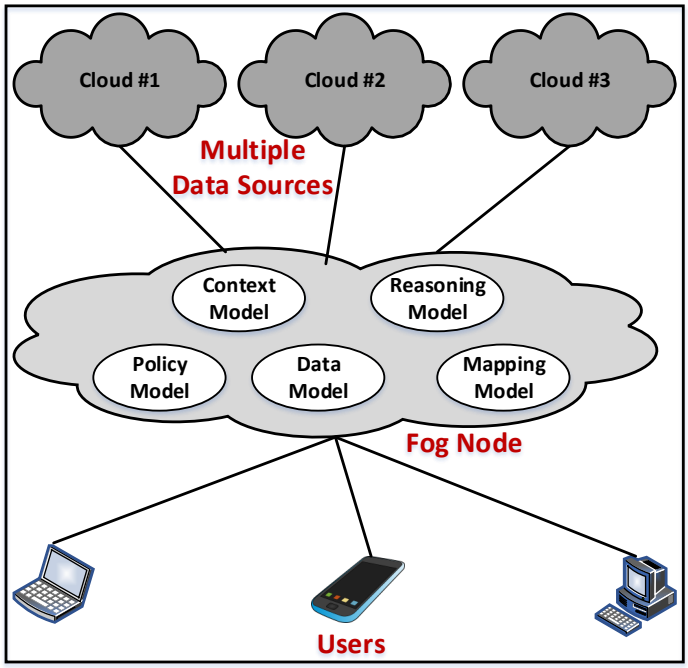


Figure 1: The Relationship Chain from Users to Cloud Servers

### *2.3. General Requirements*

The general requirements of developing the context-specific access control with imprecise fuzzy characterization are as follows:

(Req.1) There is a need for a new form of access control approach to capture the low-level imprecise fuzzy facts and consequently derive the precise fuzzy conditions from them. In this respect, we introduce a *context representation and reasoning approach* to represent the raw facts from the context sources and infer the relevant conditions from them.

(Req.2) Also, an access controller needs to take into account both the fuzzy conditions and other relevant contextual conditions for context-specific decision making. As such, we introduce a *policy model* to incorporate these conditions into the access control policies.

(Req.3) In addition, an access controller needs to deal with multiple data sources.

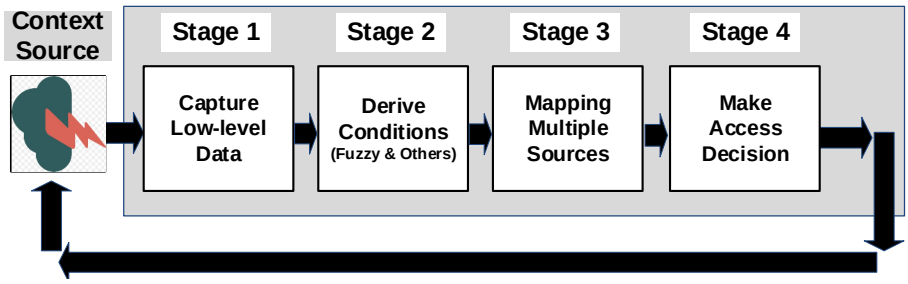


Figure 2: Our FCAAC Approach

As such, we introduce *a unified data model and a mapping model* to map all the local data sources to the generic schema.

## 3. Formal FCAAC Approach

In this section, we first introduce a high-level approach to Context-Aware Access Control using Fuzzy logic (FCAAC), including the stages of representing and reasoning fuzzy and normal contextual conditions, mapping multiple sources and making context-sensitive access control decisions.

Figure 2 shows a conceptual high-level approach to FCAAC, which includes 4 basic steps: capture low-level contextual facts, derive relevant contextual conditions (fuzzy and normal), map between multiple data sources and make context-sensitive access control decisions. Stage 1 is the process of gathering low-level contextual facts from the relevant context sources. Stage 2 is the process of inferring relevant fuzzy and other contextual conditions from the low-level contextual facts. Stage 3 is the process of mapping all local data sources to a unified data schema. Finally, Stage 4 is the process of making context-sensitive access control decisions based on the relevant contextual conditions. In the following, we present all fundamental definitions with the purpose to illustrate our FCAAC approach.

### *3.1. Context Model*

The development of a relevant Context-Aware Access Control (CAAC) approach is a complex task because of the need to accommodate for a wide variety of contextual conditions. The first step in achieving this is to define these conditions.

#### *3.1.1. Representation of Contextual Conditions:*

In the literature, many researchers have defined the context information. The most well accepted definition is given by Dey [37], *context is any information about the situation of an entity, where an entity can be a person, place or object*. In general, it is a broad and generalized vision of what the context means for context-aware applications. However, based on our application scenario, we need to represent the different types of contextual conditions as some conditions which only can be derived by utilizing fuzzy sets and fuzzy logic-based reasoning.

**Definition 1.** *(Fuzziness of Context Information) According to the degree of fuzziness of context information, we classify contextual conditions into fuzzy conditions and normal conditions, i.e., contextual conditions (CC) is the set of all fuzzy conditions (FC) and all normal conditions (NC).*

$$CC \;=\; FC \;\cup\; NC \tag{1}$$

**Definition 2.** *(Fuzzy Contextual Condition) A fuzzy contextual condition is an implicit context information and it can be derived from a fuzzy set by means of a concept (i.e., contextual condition) with its values. On the basis of the fuzzy set theory [19], a decimal point or truth value ranging from 0 to 1 is generally used to characterize the degree of membership of the values to a concept.*

The elements (fuzzy contextual conditions) of a fuzzy set have the truth values (tValues) ranging from 0 for non-membership to 1 for full-membership.

$$\mu_{fc(v)} \;\in\; [0,1] \tag{2}$$

In the above expression, '$fc$' denotes a fuzzy condition ($fc \in FC$) and '$\mu_{fc(v)}$' denotes a membership degree of a concept '$fc$' for a certain value '$v$'.

**Example 1.** *A patient's current health status (PCHState) is 95% critical, which is a fuzzy contextual condition. The degree of membership is represented in the following expression.*

$$\begin{gathered} \mu_{PCHState(critical)} \ = \ 0.95, \ i.e., \\ PCHState \ = \ \text{“}critical\text{”}, \ where \ tValue \ = \ 0.95 \end{gathered} \tag{3}$$

**Definition 3.** *(Normal Contextual Condition) A normal contextual condition is an implicit context information and it can be derived from a classical crisp set by means of a concept with its values. On the basis of the classical crisp set theory, a truth value 0 or 1 is generally used to characterize the degree of membership of the values to a concept.*

The elements (normal conditions) of a crisp set have the truth values either 0 for non-membership or 1 for full-membership. The degree of membership of a concept '$nc$' ($nc \in NC$) to its value '$v$' is represented in the following expression.

$$\mu_{nc(v)} \ \in \ \{0, 1\} \tag{4}$$

**Example 2.** *In our application scenario, the interpersonal relationship (interRelationship) between Bob and Mary is assigned nurse, which is a normal contextual condition. The degree of membership is represented in the following expression.*

$$\begin{gathered} \mu_{interRelationship(assignedNurse)} \ = \ 1, \ i.e., \\ interRelationship \ = \ \text{“}assignedNurse\text{”} \end{gathered} \tag{5}$$

**Example 3.** *In the same application scenario, the relationship between Bob and Jane is non-treating physician. The degree of membership is represented in the following expression.*

$$\begin{gathered} \mu_{interRelationship(treatingPhysician)} \ = \ 0, \ i.e., \\ interRelationship \ = \ \text{“}non - treatingPhysician\text{”} \end{gathered} \tag{6}$$

*3.1.2. Reasoning about Contextual Conditions:*

The context reasoning part includes two types of inference rules to derive fuzzy and normal contextual conditions. The first set of rules are used to infer the fuzzy contextual conditions for the precise linguistic labels and the crisp boundary values (e.g., a patient's current health status is "*66% normal*" with "*criticality level 34%*") from the low-level fuzzy facts through fuzzy-logic based reasoning. The second set of rules are used to infer the normal contextual conditions from the low-level context information through normal rule-based reasoning.

Further details of the reasoning about these conditions using fuzzy logic-based and ontology-based inference rules are discussed in Section 4.2.

*3.2. Policy Model*

Role-Based Access Control [38] is an emerging model of access control and is well recognized for its many advantages in large-scale authorization management [39]. It provides the core concepts of user-role and role-permission assignments in which a user can exercise organizational functions that are associated with the roles. Our core CAAC policy model [3] extends the traditional RBAC model to support context-oriented access control according to normal contextual conditions. This section introduces a formal FCAAC policy model, which extends our core CAAC policy model to a further coverage of fuzzy contextual conditions.

**Definition 4.** *(FCAAC Policy Model) A Fuzzy logic-based Context-Aware Access Control (FCAAC) policy model is denoted by a 4-tuple relation.*

$$FCAAC \;=\; \langle U,\; R,\; CC,\; P \rangle \tag{7}$$

In the above relation, '$U$' represents a set of system users who are the resource requesters, '$R$' represents a set of roles, '$CC$' represents a set of contextual conditions, and '$P$' represents a set of permissions or rights to perform some operations on resources (read or write) by the users who initiate access requests.

If '$u$' represents a user ($u \in U$), '$r$' represents a role ($r \in R$), '$cc$' represents a contextual condition ($cc \in CC$) and '$p$' represents a permission ($p \in$

$P$), then, together the elements '$Users$', '*Roles*', '*Contextual Conditions*' and '$Permissions$' form the *FCAAC Policy Model*.

$$\begin{aligned} CC &= FC \cup NC \\ Users(U) &= a\ set\ of\ users \\ Roles(R) &= a\ set\ of\ roles \\ ContextualConditions(CC) &= a\ set\ of\ contextual\ conditions \\ Permissions(P) &= a\ set\ of\ permissions \end{aligned} \tag{8}$$

**Definition 5.** *(A FCAAC Policy) A FCAAC policy specifies whether a user in an appropriate role is granted a permission associated with that role to access the information resource(s) in order to perform some operations on that resources(s), when the relevant contextual conditions are satisfied. We consider the contextual conditions as the policy constraints and they can be formed by integrating the relevant fuzzy and/or normal contextual conditions.*

**Example 4.** *Consider the application scenario presented in Section 2, where Mary wants to access certain medical records of patient Bob, the FCAAC policy determines whether the access permission is granted or denied. An example FCAAC policy associated with this scene can be read as: "a user by playing a registered nurse (RN) role is permitted to access the daily medical records (DMR) of a patient, during her ward duty time from the location where the patient is located in the general ward, and if she is assigned to monitor his health status, and only when his current health status is within normal ranges". The rule shown in Table 1 expresses the policy,* $fcaac_1 = \langle Mary,\ RN,\ cc_1,\ DMR \rangle$.

In this example, the access control decision is based on the following constraints: *who* the user is (e.g., $Mary$), *what* role the user can play (e.g., $RN$), *what* resource is being requested (e.g., write operation on DMR, $writeDMR$) and under *what* contextual conditions (e.g., $cc_1$). Looking at our application scenario, the contextual condition '$cc_1$' is based on a normal condition '$nc_1$' (e.g., Mary's location address is "general ward" and request time is "duty time", and the interpersonal relationship between Mary and Bob is "assigned nurse") and

Table 1: An Example FCAAC Policy for the Registered Nurses

***If***

$FCAACPolicy(fcaac_1) \wedge User(u_1)$

$\wedge\ hasUser(fcaac_1, u_1) \wedge equal(u_1, \text{“}Mary\text{”})$

$\wedge\ \ Role(r_1) \wedge hasRole(fcaac_1, r_1) \wedge\ \ equal(r_1, \text{“}RN\text{”}) \wedge Permission(p_1)$

$\wedge\ hasPermission(fcaac_1, p_1) \wedge equal(p_1, \text{“}writeDMR\text{”})$

$\wedge\ ContextualCondition(cc_1) \wedge hasCondition(fcaac_1, cc_1)$

$\wedge\ NormalCondition(nc_1) \wedge FuzzyCondition(fc_1)$

$\wedge\ hasContext(cc_1, nc_1 \vee fc_1)$

***Then***

$canAccess(u_1, p_1)$

a fuzzy condition '$fc_1$' (e.g., Bob's current health status is "66% normal" with "criticality level 34%"), and it can be represented as, $cc_1 = nc_1 \vee fc_1$.

*3.3. Unified Data and Mapping Models for Multiple Cloud Sources*

In this section, we present a unified data model and its associated mapping rules in order to correlate the general data schema with other local data sources.

**Definition 6.** *(Unified Data Schema) A unified data schema (UDS) is represented as a 2-tuple relation. It includes the general and equivalent concepts.*

$$UDS = \langle GC,\ EC \rangle \tag{9}$$

In the next section, we have introduced the FCAAC ontology, where $GC$ is represented as a set of general classes ($gc \in GC$) and $EC$ is represented as a set of domain-specific subclasses or equivalent class ($ec \in EC$). The object properties are used to represent the associations between the general classes and the equivalent classes.

**Definition 7.** *(Mapping Model) A set of mapping rules is represented as one-to-one or one-to-many relationships between the general and equivalent concepts.*

*An equivalent concept of the general concept can be formed based on the other single or multiple concepts. Let us consider another set of concepts $C$ ($c \in C$), each equivalent concept $ec \in EC$ can be a single concept of the subset elements of $C$ or can be represented by making conjunctions of the subset elements of $C$.*

$$\begin{aligned} GC &\equiv EC \\ EC &= \{(..., (c_1), (c_2), (c_1 \wedge c_2), \\ &(c_1 \wedge c_2 \wedge c_3), ...) | ec \in EC \ \& \ c \in C\} \end{aligned} \tag{10}$$

In the above relations, we use $c \in C$ to represent a single concept, $C$ to represent a set of concepts, $ec \in EC$ to represent an equivalent concept and $gc \in GC$ to represent a general concept.

**Example 5.** *Looking at the application scenario, the concept User in the healthcare professional snapshot is equivalent to the concept Person in different domain, the concept Location in the healthcare scenario is equivalent to the concept Place, and the concept Resource in the healthcare scenario is equivalent to the concept Object in another domain.*

Our proposed FCAAC approach with unified data and mapping models facilitates access control to cloud-based data resources from multiple sources. Further details of the FCAAC specification to access data and information resources from multiple sources using ontology-based languages are discussed in the following section (see Section 4.4).

## 4. Ontology-based FCAAC Approach

In the previous section, we have presented all preliminary formal definitions of our FCAAC approach. This section introduces an ontology-based approach, to realize our formal approach in practice. In addition, we show the related examples from the application scenario.

We introduce the FCAAC ontology to model the contextual conditions, utilizing user-defined inference rules to derive the relevant conditions from the low-level context information. In the FCAAC ontology, we also model the access

control policies, incorporating these contextual conditions. Riboni and Bettini [40] have shown that ontologies are well-suited for representing and modelling dynamic contextual conditions and are very useful semantic technologies for pervasive computing applications. The FCAAC ontology is defined in Web Ontology Language (OWL) [41]. We have chosen OWL rather than other ontology languages, because it is more expressive to specify the contextual conditions and policies in an easy and natural manner, than others [40]. Also, it is a widely used ontology language in semantic Web. In order to infer new knowledge, the expressivity of OWL is extended by incorporating the SWRL (Semantic Web Rule Language) rules [42] to the FCAAC ontology.

The FCAAC ontology, as depicted in Figure 3, has the core concepts *User*, *Role*, *ContextualCondition*, *Permission*, *Resource*, *Operation* and *AccessDecision*, which are organized into a *FCAACPolicy* hierarchy. It is divided into three layers. The top layer, which extends our core CAAC policy ontology [3] to a further coverage of fuzzy contextual conditions and includes the concepts for modelling the FCAAC policies. The middle layer includes the core concepts for modelling the fuzzy and normal contextual conditions. The bottom layer includes the core concepts for modelling the context information.

The detailed representation of a wide range of context information is out of the scope of this article. In our earlier research [3, 4, 12], we have already introduced context ontologies to represent and model the access control-specific context information (e.g., the interpersonal relationships, the purpose-oriented situations, the social, health and personal profiles).

### *4.1. Modelling Contextual Conditions*

The middle layer in Figure 3 has the concepts *NormalCondition*, *FuzzyCondition* and *Membership*, which are organized into a *ContextualCondition* hierarchy. The relationships between these concepts are represented by object and data type properties. The links between a concept (classes and subclasses) and its attributes are achieved via data type properties, and the links between two concepts are achieved by means of object properties (built-in and user-defined)

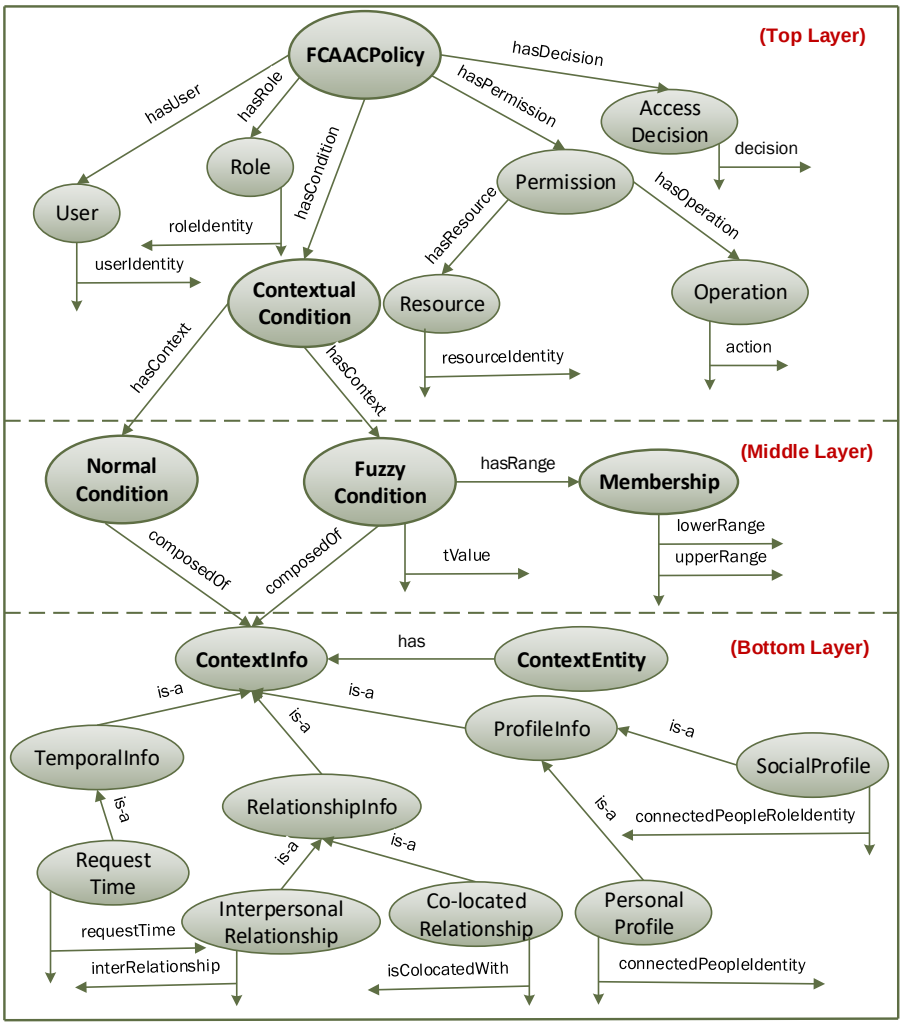


Figure 3: The FCAAC Ontology

with '*rdfs:domain*' and '*rdfs:range*'.

A contextual condition consists of the relevant fuzzy and normal conditions. Thus, the *ContextualCondition* class has an object property named *hasContext*, which is used to link the *ContextualCondition* class and the union of *NormalCondition* and *FuzzyCondition* classes (see Table 2).

The object property *hasRange* is used to link the classes *FuzzyCondition* and

Table 2: 'hasContext' Object Property Definition in OWL

```
<owl:ObjectProperty rdf:ID="hasContext">
  <rdfs:domain rdf:resource="#ContextualCondition"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#NormalCondition"/>
        <owl:Class rdf:about="#FuzzyCondition"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
```

*Membership* (see the ontological definition in Table 3).

Table 3: 'hasRange' Object Property Definition in OWL

```
<owl:ObjectProperty rdf:ID="hasRange">
  <rdfs:domain rdf:resource="#FuzzyCondition"/>
  <rdfs:range rdf:resource="#Membership"/>
</owl:ObjectProperty>
```

The *FuzzyCondition* class contains a '*xsd:float*' type data property named *tValue* (see Table 4), which denotes a membership degree (or truth value) of a concept for a certain value. For example, concerning our application scenario, Bob's current health status is "*66% normal*", which means that the criticality level (*tValue*) is 0.34.

The class *Membership* has two '*xsd:float*' type data properties, named *lowerRange* and *upperRange* (see Tables 5 and 6), which denote the ranges of membership degree for a fuzzy condition. These properties are used to specify the fuzzy conditions in the FCAAC policies. For example, a patient's current health

Table 4: 'tValue' Data Type Property Definition in OWL

```
<owl:DatatypeProperty rdf:ID="tValue">
  <rdfs:domain rdf:resource="#FuzzyCondition"/>
  <rdfs:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>
```

status is "*normal*", which has a *lowerRange* of criticality 0 and an *upperRange* of criticality 0.50.

Table 5: 'lowerRange' Data Type Property Definition in OWL

```
<owl:DatatypeProperty rdf:ID="lowerRange">
  <rdfs:domain rdf:resource="#Membership"/>
  <rdfs:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>
```

Table 6: 'upperRange' Data Type Property Definition in OWL

```
<owl:DatatypeProperty rdf:ID="upperRange">
  <rdfs:domain rdf:resource="#Membership"/>
  <rdfs:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>
```

The normal and fuzzy contextual conditions are composed of the relevant context information specific to access control, using an object property named *composedOf*. The *NormalCondition* and *FuzzyCondition* classes use the concept *ContextInfo* (which is a bottom layer concept) from the core context ontology, which is already introduced in our earlier work [3]. Table 7 specifies the 'composedOf' definition in OWL. It shows that the union of *NormalCondition* and *FuzzyCondition* classes is linked to the class *ContextInfo*.

The bottom layer of the FCAAC ontology defines the general concepts concerning the different types of context entities under the hierarchy of *ContextEntity* and the general concepts concerning the different types of context informa-

Table 7: 'composedOf' Object Property Definition in OWL

```
<owl:ObjectProperty rdf:ID="composedOf">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#NormalCondition"/>
        <owl:Class rdf:about="#FuzzyCondition"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#ContextInfo"/>
</owl:ObjectProperty>
```

tion under the hierarchy of *ContextInfo*. For example, to define the relationship, profile and temporal context information, we define the classes *RelationshipInfo*, *ProfileInfo* and *TemporalInfo*, and their subclasses *InterpersonalRelationship* and *Co-locatedRelationship*, *PersonalProfile* and *SocialProfile*, and *RequestTime* respectively. The classes *RelationshipInfo*, *ProfileInfo* and *TemporalInfo* are the subclasses of the class *ContextInfo* (see the ontological definition in Table 8).

The top and bottom layers of the FCAAC ontology also contain the data type and object properties. The domain and range of object properties are specified in Table 9 and the data type properties are shown in Table 10.

### *4.2. Reasoning about Contextual Conditions*

The reasoning part includes two sets of inference rules to derive the normal and fuzzy contextual conditions: ontology-based and fuzzy logic-based rules.

#### *4.2.1. Inferring Normal Contextual Conditions:*

The semantic rules that are used to derive the normal conditions are expressed in SWRL by means of FCAAC ontology concepts/properties and SWRL

Table 8: ContextInfo Class and Its Subclasses

| |
|---|
| <owl:class rdf:ID="**RelationshipInfo**"><br><rdfs:subClassOf rdf:resource="#**ContextInfo**"/><br></owl:class> |
| <owl:class rdf:ID="**ProfileInfo**"><br><rdfs:subClassOf rdf:resource="#**ContextInfo**"/><br></owl:class> |
| <owl:class rdf:ID="**TemporalInfo**"><br><rdfs:subClassOf rdf:resource="#**ContextInfo**"/><br></owl:class> |

built-ins functions. An example reasoning rule to derive the interpersonal relationship between user and patient is specified in Table 11. The interpersonal relationship is inferred from the low-level context information which is already represented in our context ontology [3, 11], i.e., from the user's personal profile and the patient's social profile information.

*4.2.2. Inferring Fuzzy Contextual Conditions:*

The inference rules that are used to derive the fuzzy conditions are expressed in "if-then statements" by means of the specification of linguistic labels, where the first part (*if*) contains the input conditions and the second part (*then*) contains an action output. An example set of fuzzy logic-based reasoning rules to derive the current health status of the patients is specified in Table 12. The first rule in Table 12 can be read as, if *PAge* is "*Young*" and *PulseR* is "*T4*", then *PCHState* is "*Normal*". Further details can be found in prototype implementation section (see Section 5.1).

One of the main contributions of this research is to derive the fuzzy contextual conditions from the low-level information, utilizing fuzzy-logic-based context reasoning. Towards this goal, Figure 4 shows our fuzzy context information system, which includes three main steps for mapping between crisp

Table 9: Domain and Range Restrictions for Object Properties

| **Object Property** | **Domain** | **Range** | **Description** |
|---|---|---|---|
| hasUser | FCAACPolicy | User | A FCAAC policy is connected to a user |
| hasRole | FCAACPolicy | Role | A FCAAC policy is connected to a role which is played by a user |
| hasCondition | FCAACPolicy | ContextualCondition | A FCAAC policy has the relevant contextual conditions |
| hasDecision | FCAACPolicy | AccessDecision | A FCAAC policy has a relevant access decision |
| hasPermission | FCAACPolicy | Permission | A FCAAC policy has a permission |
| hasResource | Permission | Resource | A FCAAC policy has a permission to access resource |
| hasOperation | Permission | Operation | A FCAAC policy has a permission to perform different operations on resource |

and fuzzy datasets: fuzzification, fuzzy reasoning and defuzzification [19]. Our FCAAC ontology captures the low-level data from the context sources and sends them for fuzzification. Fuzzification is the process of representing these inputs (from the crisp values) into their linguistic labels using membership functions. Fuzzy reasoning is the process of deriving the linguistic outputs from the given

Table 10: Data Type Properties

| **Data Type Property** | **Domain** |
|---|---|
| userIdentity | User |
| roleIdentity | Role |
| resourceIdentity | Resource |
| decision | AccessDecision |
| action | Operation |
| requestTime | RequestTime |
| interRelationship | InterpersonalRelationship |
| isColocatedWith | Co-locatedRelationship |
| userIdentity | PersonalProfile |
| roleIdentity | PersonalProfile |
| connectedPeopleIdentity | SocialProfile |
| connectedPeopleRoleIdentity | SocialProfile |

Table 11: A Reasoning Rule to Infer the Interpersonal Relationship

| |
|---|
| User(?u) ∧ Role(?role) ∧ hasRole(?u, ?role) ∧ swrlb:equal(?role, "RN") ∧ Owner(?o) ∧ Resource(?r) ∧ isOwnedBy(?r, ?o) ∧ InterpersonalRelationship(?rel) ∧ hasRelationship(?u, ?rel) ∧ hasRelationship(?o, ?rel) ∧ **PersonalProfile**(?pp) ∧ hasProfile(?u, ?pp) ∧ userIdentity(?pp, ?userID) ∧ roleIdentity(?pp, ?roleID) ∧ **SocialProfile**(?sp) ∧ hasProfile(?o, ?sp) ∧ connectedPeopleIdentity(?sp, ?connID) ∧ connectedPeopleRoleIdentity(?sp, ?connRoleID) ∧ **swrlb:equal**(?userID, ?connID) ∧ **swrlb:equal**(?roleID, ?connRoleID) → interRelationship(?rel, **"assignedNurse"**) |

linguistic inputs in terms of fuzzy logic. As such, it selects the required reasoning rules from a fuzzy rule-base and executes them using the fuzzy inference engine. Defuzzification is the process of combining all linguistic outputs into a

Table 12: A Set of Reasoning Rules to Infer the Current Health Status

| | | | |
|---|---|---|---|
| ***If*** | $PAge(Young) \wedge PulseR(T4)$ | ***Then*** | $PCHState(Normal)$ |
| ***If*** | $PAge(Young) \wedge PulseR(T5)$ | ***Then*** | $PCHState(Normal)$ |
| ***If*** | $PAge(MiddleAge) \wedge PulseR(T4)$ | ***Then*** | $PCHState(Normal)$ |
| ***If*** | $PAge(MiddleAge) \wedge PulseR(T5)$ | ***Then*** | $PCHState(Critical)$ |

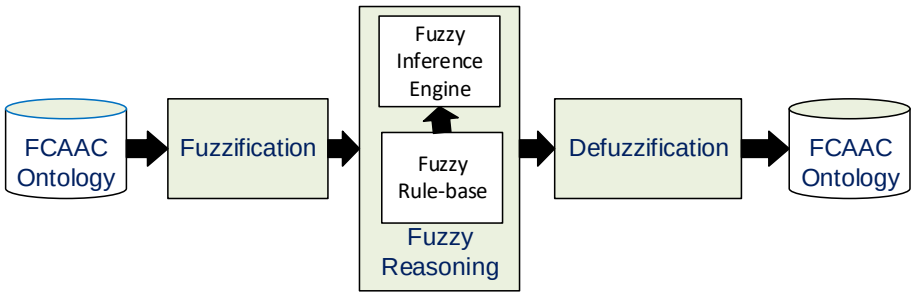


Figure 4: The Fuzzy Context Information System

single/composite crisp result. Finally, Our FCAAC ontology stores such inferred result/condition.

### *4.3. FCAAC Policy*

We use the OWL ontology language to represent the FCAAC policy concepts and their relationships (see top layer in Figure 3). OWL-based reasoning rules are not always sufficient to infer the implicit information from the low-level information. For example, in order to compare the first and second arguments (e.g., they are the 'same', 'less than' or 'greater than'), we use the SWRL language and its built-in functions to represent the fuzzy contextual conditions in our ontology, in terms of their linguistic labels and the ranges of their degree of membership. As such, we codify the FCAAC policies with OWL and SWRL languages.

Table 13: An Example Policy in Ontology format for the Registered Nurses

| | |
|---|---|
| 1 | <**FCAACPolicy** rdf:ID="fcaac$_1$"> |
| 2 | <hasUser rdf:resource="#User_RN$_1$"/> |
| 3 | <hasRole rdf:resource="#Role_RN"/> |
| 4 | <hasPermission rdf:resource="#Permission_writeDMR"/> |
| 5 | <hasCondition rdf:resource="#ContextualCondition_cc$_1$"/> |
| 6 | <hasDecision rdf:resource="#AccessDecision_Granted"/> |
| 7 | </**FCAACPolicy**> |
| 8 | FCAACPolicy(?fcaac$_1$) ∧ User(?u) ∧ hasUser(?fcaac$_1$, ?u) ∧ userIdentity(?u, "**RN**$_1$") |
| 9 | ∧ Role(?r) ∧ hasRole(?fcaac$_1$, ?r) ∧ canPlay(?u, ?r) ∧ roleIdentity(?r, "**RN**") ∧ |
| 10 | Permission(?per) ∧ hasPermission(?fcaac$_1$, ?per) ∧ Resource(?res) ∧ |
| 11 | hasResource(?per, ?res) ∧ resourceIdentity(?res, "**DMR**") ∧ |
| 12 | Owner(?o) ∧ isOwnedBy(?res, ?o) ∧ Operation(?op) ∧ |
| 13 | hasOperation(?per, ?op) ∧ action(?op, "**Write**") ∧ |
| 14 | **ContextualCondition**(?**cc**$_1$) ∧ hasCondition(?fcaac$_1$, ?cc$_1$) ∧ |
| 15 | **NormalCondition**(?**nc**$_1$) ∧ **hasContext**(?cc$_1$, ?nc$_1$) ∧ |
| 16 | InterpersonalRelationship(?rel) ∧ hasRelationship(?u, ?rel) ∧ |
| 17 | hasRelationship(?o, ?rel) ∧ interRelationship(?**rel**, **"assignedNurse"**) ∧ |
| 18 | RequestTime(?rt) ∧ hasRequestTime(?u, ?rt) ∧ requestTime(?**rt**, "**dutyTime**") |
| 19 | ∧ Co-locatedRelationship(?col) ∧ hasRelationship(?u, ?col) ∧ |
| 20 | hasRelationship(?o, ?col) ∧ isColocatedWith(?**col**, **yes**) ∧ |
| 21 | **composedOf**(?nc$_1$, ?**rel**) ∧ **composedOf**(?nc$_1$, ?**rt**) ∧ |
| 22 | **composedOf**(?nc$_1$, ?**col**) ∧ |
| 23 | **FuzzyCondition**(?**fc**$_1$) ∧ **hasContext**(?cc$_1$, ?fc$_1$) ∧ **PCHState**(?hs) ∧ |
| 24 | **composedOf**(?fc$_1$, ?**hs**) ∧ **swrlb:equal**(?hs, "**normal**") ∧ **tValue**(?fc$_1$, ?tv) ∧ |
| 25 | Membership(?m) ∧ hasRange(?fc$_1$, ?m) ∧ lowerRange(?m, ?lr) ∧ |
| 26 | **swrlb:equal**(?lr, **0**) ∧ upperRange(?m, ?ur) ∧ **swrlb:equal**(?ur, **0.50**) ∧ |
| 27 | **swrlb:greaterThan**(?tv, lr) ∧ **swrlb:lessThan**(?tv, ur) ∧ |
| 28 | AccessDecision(?dec) ∧ hasDecision(?fcaac$_1$, ?dec) → decision(?dec, "**Granted**") |

*4.3.1. An Example FCAAC Policy:*

Let us consider the registered nurses' policy shown in Table 1. In this policy, the access decision is based on the following constraints: *who the requester/user* is (e.g., registered nurse, *RN*), *what resource* is being requested (e.g., daily medical records (DMR) on write operation) and *under what contextual conditions* the user sends the request (current health status, request time, and interpersonal and co-located relationships).

An example FCAAC policy rule in OWL is shown in the top part in Table 13. The core policy concepts are specified in *Line #1 to 7*). The policy illustrates that a user, by playing a registered nurse ($RN$) role and satisfying the relevant contextual condition ($cc_1$), can be granted to access the daily medical records ($DMR$) of the patients.

The bottom part in Table 13 illustrates the specification of contextual conditions and other policy constraints (e.g., fuzzy conditions, role identity) in SWRL. The main conditions and constraints are represented in bold type. Firstly, the user and role specifications are shown in *Line #8 to 9*, and the permission specification is shown in *Line #10 to 13*. The, the basic condition construction is specified in *Line #14*, which is related to the normal and fuzzy contextual conditions. The normal contextual condition is specified in *Line #15 to 22*, and the fuzzy contextual condition is specified in *Line #23 to 27*. Finally, the access control decision is specified in *Line #28*.

In the previous section, an example SWRL-based reasoning rule in Table 11 is used to determine the user and patient have a '*assignedNurse*' relationship, and an example set of fuzzy logic-based reasoning rules in Table 12 is used to determine a patient's current health status is '*normal*'. The reasoning rules to derive the *request time* and *co-located relationship* can be found in our earlier work [3].

One of the key features of our FCAAC ontology is its ability to specify the fuzzy contextual conditions at different membership/criticality levels (see the middle layer in Figure 3). For example, in the above policy, Mary can access

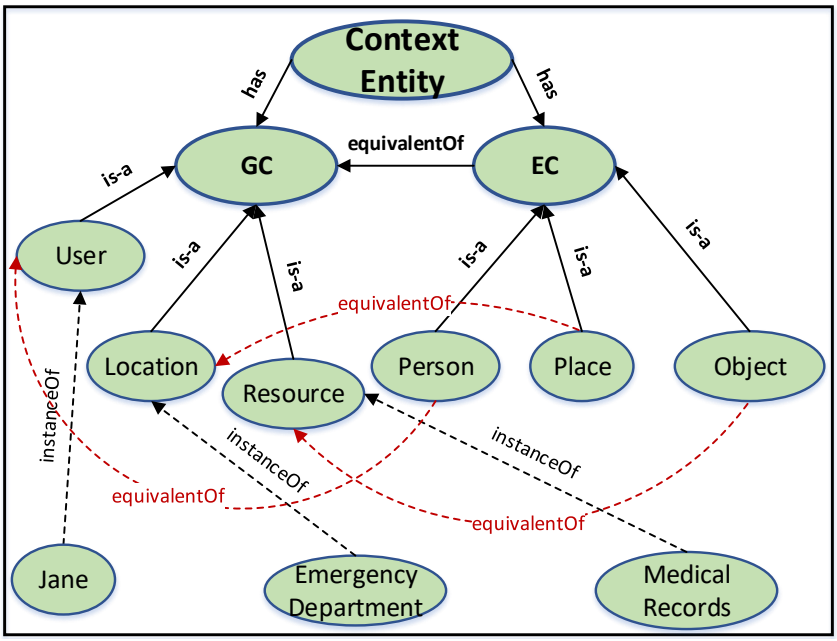


Figure 5: An Excerpt of the UDS Data Ontology

Bob's DMR when his current health status is "*normal*", which means that the criticality levels of the degree of membership are between 0 (*lowerRange*) to 0.50 (*upperRange*). However, Mary is not granted access to Bob's DMR from the general ward of the hospital, when his current health status is "*high critical*" or "*critical*", as he needs to admit immediately in the emergency department of the hospital in such a situation. That is, our FCAAC policy model provides access control decisions by taking into account the fuzzy contextual conditions.

### *4.4. Data and Mapping Ontologies for Cloud-Based Data Resources*

We extend the bottom layer of our FCAAC ontology (see Figure 3) in order to correlate the general data schema with multiple data sources. As such, we define the general concepts (i.e., general context entities) and the equivalent concepts (i.e., domain-specific context entities). Figure 5 shows an excerpt of the UDS data ontology, named *Unified Data Schema* (UDS). It has two core classes: *GC* (general concepts) and *EC* (equivalent concepts). The classes *GC*

and *EC* are associated with an object property named *equivalentOf*. The general concepts are the base classes and they are the subclasses of the class *GC*. For example, in the UDS ontology, the classes *User*, *Location* and *Resource* are the subclasses of the class *GC*. The subclasses are represented by *is-a* relationships. The classes *Person*, *Place* and *Object* are the domain-specific concepts and they are the subclasses of the class *EC*. In our UDS ontology, the class *Person* is equivalent to the class *User* and an individual named *Jane* is represented as an instance of the class *User*. In the following, a set of mapping rules are used to specify such equivalent relationships.

We incorporate a set of mapping rules into the UDS data ontology in order to correlate the general schema with multiple data sources. In the literature, the description logic (DL) semantics are well accepted by the modeling constructs of OWL ontology [43]. Accordingly, we specify a set of DL rules in Table 14. One of the mapping rules specifies that *Place* is an equivalent concept of *Location*.

Table 14: A Set of Mapping Rules

| |
|---|
| **GC** $\equiv$ **EC** |
| Person $\equiv$ User |
| Place $\equiv$ Location |
| Owner $\equiv$ Resource |

The detailed representation of a wide range of general context entities is out of the scope of this article. In our earlier research [3], we have already introduced different context ontologies to represent and model the general context entities (e.g., user, owner, resource, relationship).

## 5. The Evaluation of Our Approach

In this section, we first present a prototype architecture to assist application developers in rapid prototyping. Using this prototype, we develop a healthcare application, called *eHealthcare*, to validate the functionalities of our FCAAC

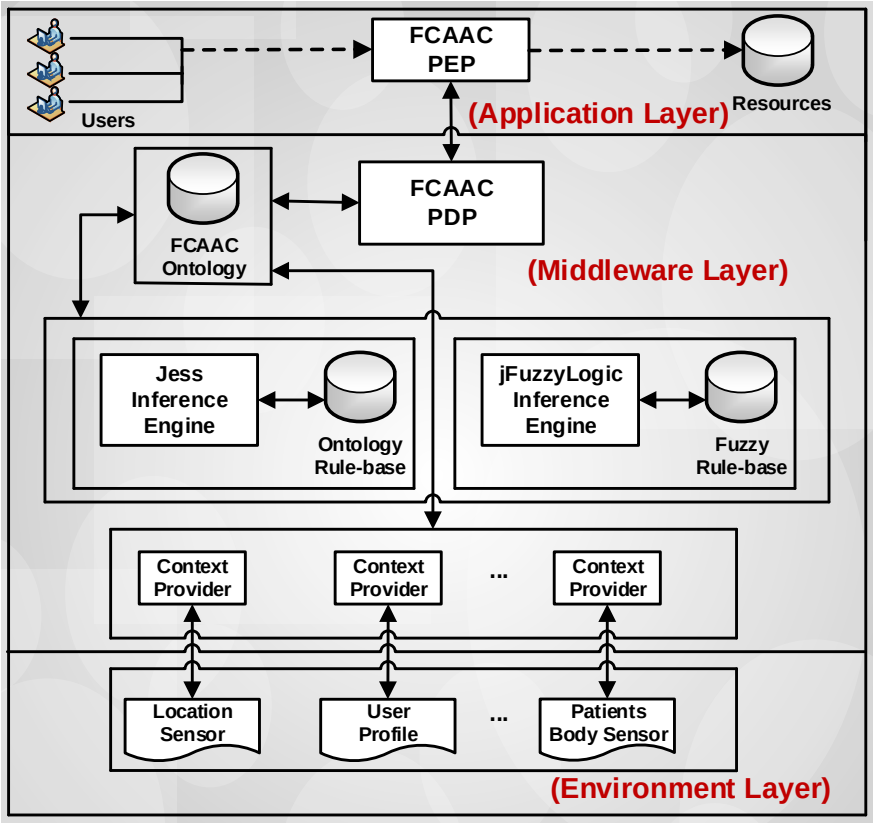


Figure 6: Overview of the Prototype Architecture of FCAAC

approach. In particular, we present two case studies from the healthcare domain to demonstrate the practicality of our context-sensitive access control approach. In addition, we conduct a usability study to demonstrate a walkthrough of our proposal. Furthermore, the deployment of *eHealthcare* application is performed by presenting an experimental evaluation of our approach.

### *5.1. Software Prototype*

To alleviate the complexities of building context-sensitive access control applications, we in this section present a software prototype of the FCAAC framework.

Figure 6 shows an architecture of the software prototype, which extends our

earlier prototype [3], utilizing both the fuzzy logic and ontology-based reasoning capabilities. It mainly includes environment, middleware and application layers. It has a set of software components that support software engineers to develop FCAAC applications using this architecture.

*5.1.1. Environment Layer:*

The environment layer includes the different types of sensors or data sources. The functional components of this layer are application-specific, which are outside our research scope. In this article, our main focus is the middleware layer and its associated components of the application layer.

*5.1.2. Middleware Layer:*

The middleware layer includes the following main components: context providers, context reasoner, fuzzy logic engine and access control processor. The context providers receive the raw context facts from the sensors or data sources, extract the low-level contextual conditions and convert these conditions to OWL representation, according to the FCAAC ontology. The FCAAC ontology captures such low-level contextual conditions from the relevant context providers.

The context reasoner consists of the context inference engine and an ontology rule-base. The context reasoner derives the high-level contextual conditions from the the low-level contextual conditions from the ontology by using the reasoning rules. These reasoning rules are user-defined and stored into the FCAAC ontology rule-base.

The FCAAC ontology is defined by using ontology languages OWL [41], SWRL [42], SWRL Built-ins [44] and DL [43], and the ontology has been generated with the Protégé-OWL graphical tool [45]. We develop an ontology rule-base to derive the normal contextual conditions from the low-level context facts using ontology-based reasoning rules, which have been generated with the Protégé-SWRLTab. We have used a rule engine that is written in Java, named Jess [46], to facilitate reasoning tasks for executing such rules. We use the Jess

rule engine because of its rule management capabilities.

The fuzzy logic engine consists of the fuzzy inference engine and a fuzzy rule-base. We develop a fuzzy rule-base to derive the fuzzy contextual conditions from the imprecise fuzzy facts using fuzzy reasoning rules, which have been expressed in the form of fuzzy conditional "if-then" statements. For executing such fuzzy rules, we have used the fuzzy inference engine, named jFuzzyLogic [47], which is written in Java. We have already shown the fuzzy reasoning processes in Figure 4. In order to execute such fuzzy and normal reasoning rules and consequently derive the implicit information (normal and fuzzy conditions), we have implemented two Java functions. The first function is used to execute the reasoning rules and infer the implicit high-level contextual conditions using low-level contextual conditions from the FCAAC ontology. The other function is used to transfer the inferred information in the ontology.

The access control processor is responsible for the evaluation of access requests. We have implemented the FCAAC PDP (policy decision point) and the FCAAC PEP (policy enforcement point) as parts of the access control processor. The FCAAC PDP is implemented in Java to determine the access request is "granted" or "denied", according to the applicable policies and the necessary contextual conditions. The context-sensitive access control policies are also stored in FCAAC ontology.

*5.1.3. Application Layer:*

We have implemented the application layer (application interface) using Java and Web technologies. Users normally communicate (by sending requests and getting responses) through this interface. The application layer includes the FCAAC PEP. Upon receiving an access request from the user, the FCAAC PDP forwards the request to the FCAAC PDP for evaluation. The detailed implementation of the context providers, FCAAC PEP and FCAAC PDP can be found in our earlier prototype [3]. We in this article mainly have discussed the implementation of the context reasoner and the fuzzy logic engine to derive the contextual conditions (fuzzy and normal contextual conditions).

### *5.2. Walkthrough of Our Proposal*

In this section, we present several case studies from the healthcare domain to demonstrate the applicability of our proposed approach. In addition, we present a usability study to demonstrate our approach in a real setup with real users.

#### *5.2.1. Case Study #1:*

We evaluate our FCAAC prototype using an *eHealthcare* application scenario described in Section 2. The *eHealthcare* application provides the healthcare professionals (e.g., emergency doctors, treating doctors, registered nurses) to access different medical records of patients based on the dynamic context information (normal and fuzzy contextual conditions).

Consider the motivating example where Mary wants to access the daily medical records (DMR) of the patient Bob, an access request is submitted to the *FCAAC PEP* for evaluation. The *FCAAC PEP* forwards the request to the *FCAAC PDP* to determine whether the access request is "granted" or "denied", according to the current contextual conditions in effect and the applicable access control policies. The applicable FCAAC policy is already specified in Table 13, which defines the permission is granted when both of the two Boolean conditions "$nc_1$" and "$fc_1$" are true. The normal contextual condition $nc_1$ is composed based on the following sub-conditions (context information): the nurse is *assigned* to monitor the patient's health condition and they both are *co-located* in the general ward during her *duty time*. The fuzzy contextual condition $fc_1$ is composed of the context information: the patient's current health status (*PCHState*). In the following, we further discuss how our proposed approach captures the *PCHState* of Bob.

For simplicity, in our *eHealthcare* application, we consider the pulse rate (*PulseR*) and age of a patient (*PAge*) are the two input fuzzy sets to derive the *PCHState* (an output fuzzy set). We also consider three fuzzy age groups: *VeryYoung*, *Young* and *MiddleAge*, a normal pulse rate that is between 75 to 110 beats per minute (bpm) (which represents seven fuzzy sets, *T1* to *T7*), and a patient's current health status which is represented using three fuzzy

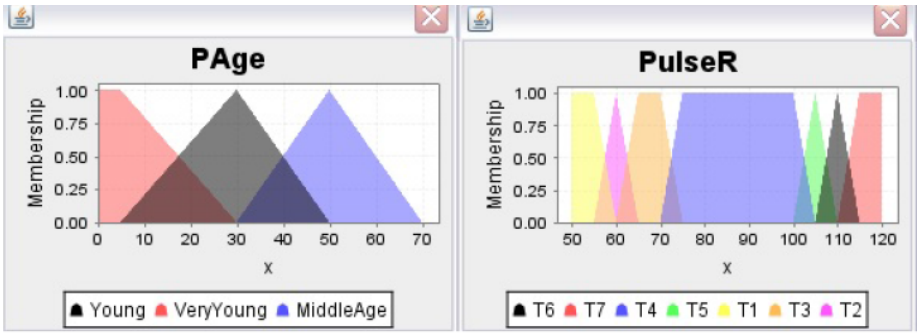


Figure 7: The Inputs Membership Functions

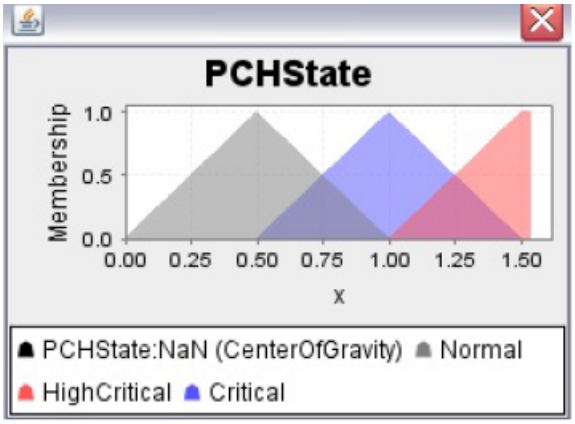


Figure 8: The Output Membership Function

sets: *Normal*, *Critical* and *HighCritical*. Based on the experience from our group's earlier research on fuzzy linguistic representations [19], these input and output fuzzy sets are characterized by triangular and trapezoidal membership functions (see Figures 7 and 8) and Mamdani's center of gravity (COG) method in conjunction with max-min inference is used for fuzzy reasoning (see Figure 9). We have specified 21 linguistic rules to cover all the possible values of *PAge* and *PulseR*.

We assume that Bob's age is 35, which belongs to the fuzzy sets *Young* and *MiddleAge* and his pulse rate is captured as 102 bpm, which belongs to the fuzzy sets *T4* and *T5*. These inputs are fired four rules, which are already specified in

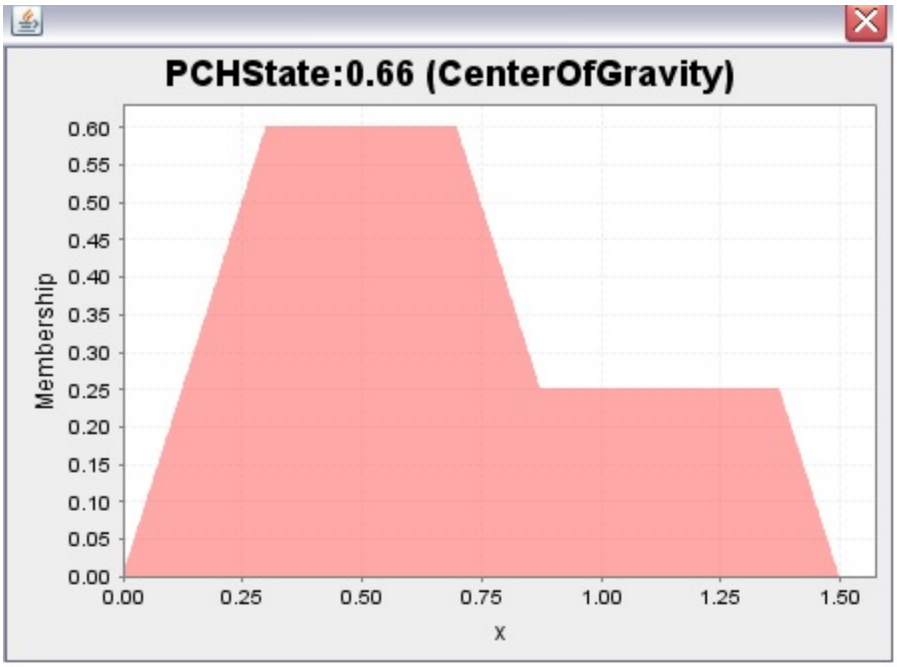


Figure 9: The Patient's Current Health Status (PCHState)

Table 12. Finally, Bob's *PCHState* is derived using the COG max-min inference method (see Figure 9). In this scenario, Mary is assigned to look after Bob and we can observe that she is granted access to Bob's DMR in his normal health condition (i.e., "*66% (0.66) normal with criticality level 0.34*").

In FCAAC, we model the criticality ranges of the *normal*, *critical* and *high critical* health status are *[0, 0.50]*, *[0.50, 0.75]* and *[0.75, 1.0]*, respectively. However, Mary is not granted access to Bob's DMR when the context changes (e.g., Bob's health condition is critical or high critical again, i.e., the criticality level is beyond the normal ranges).

Table 15 shows the access control decisions in terms of Mary's requests. We have observed that Mary, by playing a "*registered nurse*" role, has "*granted*" access to Bob's "*daily medical records (DMR)*" when his current health condition is "*normal*". However, when Bob's current health condition becomes critical or high critical again, Mary has been "*denied*" to access Bob's medical records, as

Table 15: The Context-Sensitive Access Control Decision (Case Study #1)

| **User** | **Role** | **Health Condition** | **Resource** | **Access Decision** |
|---|---|---|---|---|
| Mary | Registered Nurse | Normal | DMR | Granted |
| Mary | Registered Nurse | Critical | DMR | Denied |
| Mary | Registered Nurse | High Critical | DMR | Denied |

Bob needs emergency treatments in such situations.

*5.2.2. Case Study #2:*

Consider another case study from the same emergency scenario presented in Section 2, where Jane wants to access necessary health records of the patient Bob from multiple sources and consequently provides him emergency treatments.

The following are the contextual conditions that are included in this scenario: current health conditions (e.g., normal, critical or high critical), location addresses (e.g., emergency department), and so on. Our implemented access control processor (FCAAC PEP and FCAAC PDP) exploits these fuzzy and normal contextual conditions for making access control decisions and provides relevant resource access permissions accordingly.

Table 16 shows the access control decisions in terms of Jane's requests. In this case study, we can observe that Jane, by playing the "*emergency doctor*" role, has "*granted*" access to Bob's "*emergency medical records (EMR)*" when his current health condition is "*critical*" or "*high critical*". Such emergency medical records (EMR) are based on the different health records (e.g., health records, diagnosis records, and so on) from multiple sources, in order to provide him emergency treatments in the emergency department of the hospital. However, when Bob's current health condition becomes normal again, Jane is not allowed to provide emergency treatments to Bob. Consequently, Jane has been "*denied*" to access Bob's health records coming from multiple sources.

Table 16: The Context-Sensitive Access Control Decision (Case Study #2)

| **User** | **Role** | **Health Condition** | **Resource** | **Access Decision** |
|---|---|---|---|---|
| Jane | Emergency Doctor | Critical | EMR | Granted |
| Jane | Emergency Doctor | High Critical | EMR | Granted |
| Jane | Emergency Doctor | Normal | EMR | Denied |

*5.2.3. Case Study #3:*

In order to demonstrate the practical applicability of our proposed CAAC approach, we have demonstrated another healthcare scenario. Amanda, who is working as data analyst, can access and analyse the patients' medical records. However, she only can access Bob's medical records when the patient's current health status is normal in specific contexts (e.g., from the inside of the office during her duty time). She also can access and use such records for research purpose at anytime from anywhere, by playing a data scientist role.

The PEP forwards Amanda's request to the PDP for evaluation and consequently determines the access control decision in terms of associated contextual conditions. The contextual conditions, such as the *purpose* or *intention* to access the medical records, *request time*, *location* and patient's current *health condition* are included in this scenario.

Table 17 shows the access control decisions in terms of Amanda's requests. We have observed that Amanda, by playing a "*data analyst*" role, has "*granted*" access to Bob's medical records from her office location of the inside-of-hospital during her working hours. She also has credentials to play the "*data scientist*" role and consequently can access Bob's medical records at anytime from anywhere, however, the access is only allowed for research purpose.

*5.2.4. Usability Study:*

This section demonstrates the usability testing of our FCAAC approach with real users and in a real setup. We analyse the access control requests from dif-

Table 17: The Context-Sensitive Access Control Decision (Case Study #3)

| **User(Role)** | **Purpose** | **Location** | **Time** | **Decision** |
|---|---|---|---|---|
| Amanda(Analyst) | Analysis | Inside-of-Hospital | OfficeHours | Granted |
| Amanda(Scientist) | Research | Anywhere | Anytime | Granted |

Table 18: Access Control Requests and Responses

| **User** | **Role** | **Request** | **Granted** | **Denied** |
|---|---|---|---|---|
| Jane | *EmergencyDoctor* | 50 | 35 | 15 |
| Mary | *RegisteredNurse* | 50 | 13 | 37 |

ferent users and the responses as well. We have asked different healthcare users to send their requests to access necessary health records of the patients. We have checked the applicable access control policies and the contextual conditions in our FCAAC ontology in order to evaluate their requests under different situations (different health status, different locations, and so on).

We have asked Jane (who is healthcare doctor) to send a number of requests using several Windows machines from different locations (e.g., emergency department) to access Bob's health records (e.g., emergency medical records). We have also asked Mary (who is a registered nurse) to send a number of requests from different locations (e.g., general ward) to access Bob's health records (e.g., daily medical records). We have evaluated these requests using our developed prototype, based on the applicable context-sensitive access control policies and the contextual conditions in the FCAAC ontology. We have investigated the different requests when they all (Jane, Mary and Bob) are in the same location or different locations. We have recorded the access decisions accordingly.

In this study, we have also asked users to send their notes if they are not satisfied with the access decisions. We have analysed all such notes according to the access requests. We believe that this process is helpful to improve our software prototype, by specifying the new access control policies or refining the

existing policies accordingly. Overall, we have checked 100 access control requests from Jane and Mary (see Table 18), which are the repeated requests in different situations. Out of the 100 requests for this study, Jane has received 35 "Granted" responses, whereas Mary has received only 13 "Granted" responses. These variations are confirmed that some access requests are originated from different locations where Bob is not located. Actually, Mary only can access Bob's daily medical records if Bob or Jane is co-located with her and also satisfying other relevant contextual conditions.

Let us consider another real-world application scenario and discuss how to apply our proposed FCAAC approach in such a scenario. A paramedic Richard is allowed to play the emergency-paramedic role (which is a dynamic contextual role) if he is co-located with the patient at the place of an accident. Using our proposed FCAAC approach, he can access different health records of the patient from multiple sources. Consequently, he can acquire all the permissions assigned to both paramedic and emergency-paramedic roles to provide emergency treatments to the patient.

In summary, the purpose of the above-mentioned two case studies and a usability study demonstrates the practical applicability and a walkthrough of the whole FCAAC approach. The prototype and its software components provide an infrastructural support for building such FCAAC applications.

### *5.3. Experimental Evaluation*

In this section, we measure the query response time in order to assess the performance overhead of our proposed FCAAC approach.

We conduct two sets of experiments in our simulated healthcare environment with the aim of measuring the response time and scalability of our FCAAC proposal. In each set of experiments, we vary the number of context-sensitive access control policies with respect to different numbers of roles (healthcare roles) and contextual conditions. The conducted tests are carried out in a Windows PC with an Intel Core i7@3.6GHz Processor and 16GB of RAM. We deduce the average response time after making repeated measurements of the

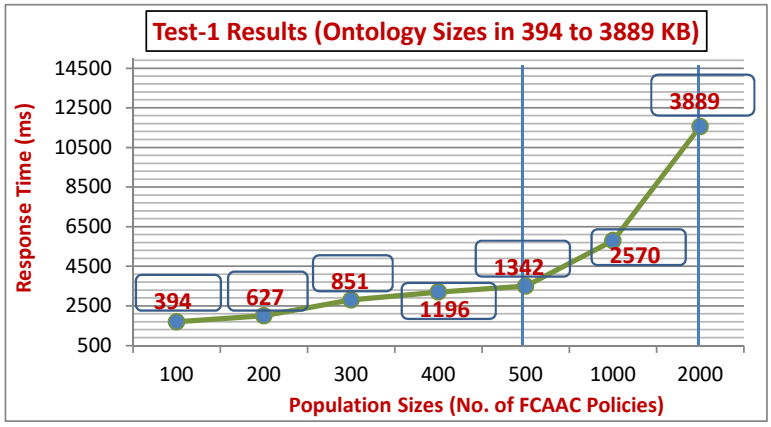


Figure 10: Populations vs Response Time

same size. The final results have been obtained by executing the experiments 10 times and computing their arithmetic mean.

In order to model the healthcare roles (e.g., general practitioners, emergency doctors), we follow the Australian standard classification of occupations (ASCO) of the health professionals [48]. In order to model the information resources (patients' health records), we follow the most implemented health level seven (HL7) standard [49]. Further details of the necessary components of a patient's medical records and different health professional roles can be found in our earlier research (please see the role and resource ontologies) [50].

*5.3.1. Experiment #1:*

In our first set of experiments, we vary the number of FCAAC policies with respect to different healthcare professional roles (e.g., emergency doctors, registered nurses, researchers). We measure the response time to provide resource access permissions to users. The number of policies contained in our FCAAC ontology is referred as population. Actually, we measure the FCAAC performance with different variations of population size. We first define an initial population of 100 policies and increase this population up to 500 for an increment of 100.

Figure 10 depicts how the response time varies, measured in milliseconds

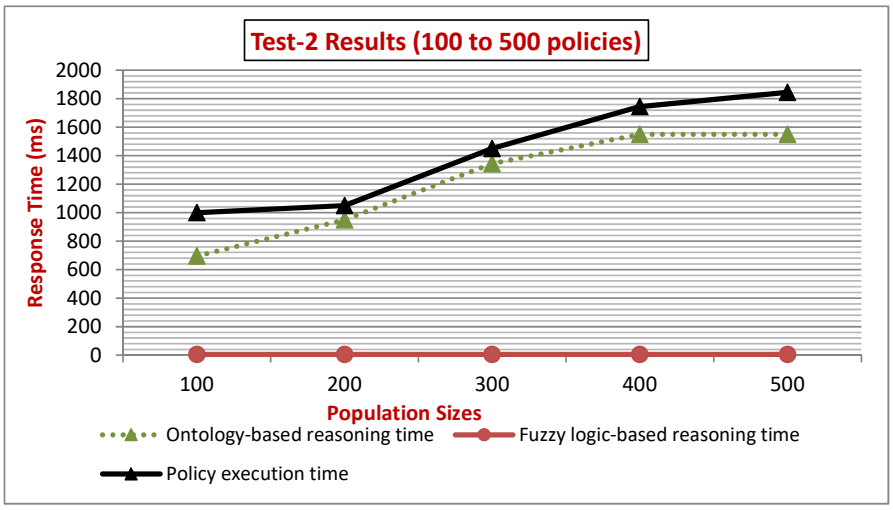


Figure 11: Stages of Response Time

(ms), considering different population sizes associated to the policies. We observe that the response time is linearly increased according to the number of policies up to 500 and it varies from 1.7 to 3.5 seconds approximately. For all populations, the difference in response time between the sizes of 394 kilobytes (KB) and 1342 KB of ontology is around a few seconds. We can say that the performance is acceptable in such a computer setup with limited computing resources.

*5.3.2. Experiment #2:*

The FCAAC reasoning model based on the fuzzy and ontology-based inference rules is one of the important parts of our proposed context-sensitive access control approach. In order to check the reasoning time and its scalability, we conduct another set of experiments. Actually, we measure the different breakdowns of the response time, where we observe the following main stages: time taken to (i) derive the fuzzy contextual conditions, (ii) derive the normal contextual conditions, and (iii) execute the access control policies for making decisions.

Figure 11 depicts the time, measured in milliseconds (ms), depending on the different stages of response time breakdown. We observe that the fuzzy logic-based reasoning in order to derive the implicit knowledge which does not have a great impact in total reasoning time (fuzzy reasoning and ontology-based reasoning). This is due to the following reasons. In our experiments, the current health status of a patient (i.e., an output fuzzy set) is derived from the pulse rate and age of the patient (i.e., two input fuzzy sets). However, the numbers of input and output fuzzy sets usually appear to be limited according to the inherent nature of context-aware access control (CAAC) applications. We also note that it does not even impact the size of the FCAAC ontology when we increase the number of fuzzy inference rules. Thus, the time taken to derive the fuzzy condition seems a straight line in Figure 11.

In these two sets of experiments, we separate the ontology loading time from the access request processing time and we only consider the access request processing time as the total response time. However, the ontology loading occurs once when the system runs the first time. Regarding the performance of our FCAAC approach, the fuzzy logic-based reasoning time has a very low impact in the overall response time to process a user's request to access the resources (see Figure 11), as the search space is limited to a small number of fuzzy inference rules. On the other hand, when we linearly increase the number of policies in our FCAAC ontology, the response time also increases linearly. However, the results fluctuate greatly at the point when we specify a large number of policies and they are more stable up to 500 policies (see Figure 10). This is due to the growing numbers of users, roles, contextual conditions and reasoning rules in the ontology. In this sense, we can conclude that the population size (i.e., the number of policies with OWL and SWRL) mainly affects the overall system performance of our FCAAC approach. Furthermore, the linearity property behind the results allows us to deduce that a better computer system with powerful computing resources would obtain a lower response time. Based on the experience from our previous work on improving system performance [51], we may adopt RDF language to build a new approach as an alternative of using

OWL language.

*5.3.3. Discussion:*

The main objective of these two sets of experiments is to evaluate the performance of our proposed FCAAC approach when it is applied to facilitate access control to necessary medical records of the patients from a single data source. Considering the experimental results, we can conclude that our proposed context-sensitive access control approach has an acceptable response time. There is still a possibility of performance improvement by using more powerful machine, however, there is a need to further investigate the performance issues in supporting context-sensitive access control to data and resources from multiple homogeneous and heterogeneous sources.

## 6. Related Work and Comparative Analysis

This section provides a short overview of the relevant access control approaches. This includes the (i) context-aware access control and (ii) fuzzy logic-based access control approaches. It also includes the (iii) cloud and fog-based access control approaches. In addition, this section includes a comparative analysis of these existing access control approaches with respect to our proposed approach.

### *6.1. Context-Aware Access Control Approaches in the Centralized Environments*

Different approaches have been proposed in literature to model role-based access control policies in conjunction with context information. Mostly these policies are based on involving the normal contextual conditions, which can be derived from the crisp sets.

Joshi et al. [6] have proposed a role-based access control (RBAC) approach and incorporated the temporal information into the RBAC policies. Bertino et al. [5] have proposed another RBAC approach, incorporating the spatial information into the policies. However, these temporal and spatial approaches

are not context-aware and adequate enough to capture and infer a wide variety of dynamically changing conditions of the environments (e.g., the relationships).

On the other hand, Bonatti et al. [7] have introduced an event-driven extension to the temporal RBAC approach. They provide an implementation of RBAC in which access control is managed by means of context information (e.g., location, time, an event such as "surgery in progress"). Schefer-Wenzl and Strembeck [8] have proposed a context-aware RBAC approach to ubiquitous systems, incorporating the context information such as time and location into the policies. Similar to [8], Hosseinzadeh et al. [9] and Trnka and Cerný [10] have proposed the context-aware RBAC approaches. Using these approaches, users can access the resources by playing the appropriate roles and based on the context information. For example, in the healthcare domain, a doctor is restricted to read the medical history of the patients after the office time or outside the hospital locations. Different from these approaches, our FCAAC approach utilizes fuzzy sets to derive the fuzzy conditions from the low-level fuzzy facts, and incorporates such fuzzy conditions along with normal contextual conditions into the policies. However, these existing context-aware RBAC approaches are not adequate to exploit the relevant contextual conditions together with fuzzy conditions for context-specific decision making at different granularity levels.

We have a successful history of using a wide range of contextual conditions for context-oriented decision making. In [3, 11], we have introduced an ontology-based context-aware RBAC approach to information resources, where we consider the context information about the state of the users, resources and their surrounding environments (e.g., patients' profiles, users' locations, users' request times). In [12], we have introduced an ontology-based relationship-aware RBAC approach, incorporating the relationship context information (e.g., the different granularity levels of relationship, the relationship types, the relationship strengths) into the policies. In [4, 13], we have introduced an ontology-based situation-aware RBAC approach, where we incorporate the purpose-oriented situation information (e.g., normal/emergency treatment purpose, research purpose) into the policies. Similar to above-mentioned context-aware approaches,

however, our earlier approaches do not provide adequate functionalities to derive and incorporate the fuzzy contextual conditions into the access control policies.

Overall, the existing context-aware RBAC approaches are not adequate to deal with imprecise context characterization and consequently derive the fuzzy conditions from the low-level fuzzy facts. For example, concerning our application scenario, Bob's current health status is "66% normal with criticality level 0.34" only can be derived from Bob's pulse rate and body temperature.

### *6.2. Fuzzy Logic-Based Access Control Approaches*

Different access control approaches have been proposed in the literature to model policies based on involving the fuzzy conditions, which can be derived from the fuzzy sets.

In [15], the authors have proposed a trust-based access control approach based on the trust values [52], allowing only authorized users to access sensitive data (and information resources) that are usually confidential. They also propose a trust model to dynamically derive the trust degrees of high, medium and low. Cheng et al. [16] have proposed a risk-adaptive access control approach for an organization to protect its sensitive information. They quantify risk as the expected value of damage and consider risk to make access control decisions (e.g., the access decision is "denied" because the risk is too high).

Takabi et al. [17] have proposed a trust-based RBAC approach to online services based on trustworthiness which is fuzzy in nature. They use fuzzy relations to compute trust values from the relevant attributes (e.g., behavioral, personal). In [18], the authors have proposed a fuzzy RBAC approach to deal with authorization-related imprecise information through fuzzy relations. They consider the various strengths of user-permission assignments as fuzzy relations to deal with such imprecise information and consequently propagate them to make access decisions.

However, these fuzzy logic-based access control approaches are not context-aware and still limited to incorporate a wide variety of access-control specific normal contextual conditions together with fuzzy conditions into the access

control policies for context-specific decision making. Different from these fuzzy logic-based approaches, our FCAAC approach provides context-specific access permissions to users exploiting both the fuzzy and normal contextual conditions, and further limits the users' access to information resources accordingly.

### *6.3. Cloud and Fog-Based Access Control Approaches in the Distributed Environments*

The above mentioned context-sensitive access control approaches have been applied to access data and information resources from centralized sources. In order to support data integration from multiple sources, in the literature, different data integration approaches have been developed, such as schema matching [22, 53], entity resolution [54], record linkage [55, 23], data fusion [56], global view [57], and ontology-based [58, 24] approaches. These integration techniques mostly have been used to map between original sources of data (i.e., different schemas) and result in a global schema. However, these approaches are still limited in order to provide the "granted" or "denied" access control decision to the users.

Due to the technological advancements in the cloud and fog environments, currently, different stakeholders need to access data and resources from multiple sources. The integration of such data that is directly coming from multiple sources raises semantic namespace and latency problems [59, 60], due to lack of semantics and cloud-based services. Towards this end, the richer semantics of data model is needed to resolve the semantic namespace problem, dealing with the homogeneous and heterogeneous nature of such big data sets. However, the latter is forcing the organizations to overcome the latency issue by adding intermediary computational nodes at the edges of the networks [61].

In recent years, fog-based access control approaches have been introduced in order to overcome the processing overheads and administration costs by moving the execution of application logic from the cloud-level to an intermediary-level (e.g., [30, 31, 32, 33]).

Zaghdoudi et al. [30] propose a generic and scalable access control approach

for fog computing with low overhead, considering information about subjects, objects and the context of operation. In [31], the authors present a recent study on intelligent transport systems utilizing fog computing and by identifying corresponding fog-based access control issues. Both research works have been concerned with several important requirements of the fog-based access control schemes, such as context-awareness and distributed architecture. The decentralization of authority from a single administrative location to other locations is also discussed. Recently, Yu et al. [32] and Zhang et al. [33] propose the fog-based access control approaches in order to provide a way to securely share data along with the benefits of encryption and decryption system. These existing fog-based access control approaches are developed to access data and resources from centralized environments. However, they are not truly context-aware and robust enough to introduce fog-based context-aware access control solutions when accessing data from multiple sources.

### *6.4. Comparative Analysis*

This section presents a comparative analysis of the existing context-sensitive, fuzzy logic-based and fog-based access control approaches with respect to our proposed FCAAC approach. In this comparative analysis, we have considered the following aspects of our FCAAC approach.

Following the traditional context-sensitive RBAC approaches [5, 6, 7, 8, 9, 10], they are not adequate to derive the **(i) Fuzzy Conditions** from the low-level contextual facts and incorporate them into the access control policies for context-sensitive decision making. In addition, our earlier context-aware RBAC approaches [3, 11, 4, 13, 12, 1] are not adequate to facilitate access control to data and resources from multiple sources. Moreover, the fuzzy logic-based approaches [15, 16, 17, 18] are not context-aware and robust enough to capture and derive the dynamically changing **(ii) Contextual Conditions** from the low-level contextual facts. On the other hand, the cloud and fog-based access control approaches [30, 31, 32, 33] consider the **(iii) Decentralization** of authority from a cloud level to the different fog locations. However, they are not

Table 19: Comparative Analysis of the Access Control Approaches

| **Approaches** | **Modeling Different Conditions** | | | **Multiple Sources** |
|---|---|---|---|---|
| | **Contextual Conditions** | **Fuzzy Conditions** | **Decentralization** | **FCAAC Policies with Mapping Capability** |
| Temporal RBAC Approach [6, 7] | *PYES* | *NO* | *NO* | *NO* |
| Spatial RBAC Approach [5] | *PYES* | *NO* | *NO* | *NO* |
| Other RBAC Approaches [8, 9, 10] | *PYES* | *NO* | *NO* | *NO* |
| Our OntCAAC Approaches [3, 11] | *YES* | *NO* | *NO* | *NO* |
| Our PO-SAAC Approaches [4, 13] | *YES* | *NO* | *NO* | *NO* |
| Our RelBOSS Approach [12] | *YES* | *NO* | *NO* | *NO* |
| Fuzzy Logic-Based Approaches [15, 16, 17, 18] | *NO* | *YES* | *NO* | *NO* |
| Our Fuzzy CAAC Approach [1] | *YES* | *YES* | *NO* | *NO* |
| Fog-Based CAAC Approaches [30, 31, 32, 33] | *PYES* | *NO* | *YES* | *NO* |
| **Our FCAAC Approach** | *YES* | *YES* | *YES* | *YES* |

truly context-aware and robust enough to support context-sensitive access control to data and information resources. Also, they are not adequate to support access control to data from **(iv) Multiple Sources**.

In this respect, different from these existing context-sensitive, fuzzy logic-based and fog-based access control approaches, our proposed FCAAC approach is robust enough and truly context-aware. It considers a wide range of contextual conditions and introduces a fuzzy model to deal with the fuzzy contextual conditions. In particular, our approach exploits the raw imprecise fuzzy facts, derives the fuzzy conditions from them and incorporates such conditions together with other contextual conditions into the access control policies for context-sensitive decision making at different granularity levels. The practical significance of our research is that it addresses the integration of fuzzy contextual conditions and other relevant contextual conditions with access control processes for access control to data and resources from multiple sources.

Table 19 shows all the results of this comparative study in which we use "**YES**" when a feature is available, "**PYES**" when a feature is partially available, and "**NO**" when a feature is not available.

Overall, our article includes both the formal and ontology-based implementation models to specify the contextual conditions (fuzzy and normal conditions) and the context-sensitive access control policies. It includes a unified data schema and a mapping model in order to access data and resources from multiple sources. It presents a walkthrough of the entire mechanism by demonstrating two case studies, a usability study and a prototype testing. Finally, it includes two sets of experiments in order to validate the feasibility of our proposed approach.

## 7. Conclusion and Future Research

In this article, we have addressed a significant research issue in order to access data and resources from multiple sources. The existing context-sensitive access control approaches can lead to a large number of access control policies

in order to access data and resources from multiple sources. On the other hand, the existing fog-based access control approaches are not adequate in today's dynamic environments due to the lack of context-awareness. Towards this end, this article introduces a fog-based access control approach to deal with cloud-based data resources from multiple sources.

The FCAAC approach proposed in this article represents a flexible policy specification solution to the problem of incorporating fuzzy contextual conditions, in the domain of access control to data and information resources utilizing the benefits of fuzzy sets. Our approach significantly differs from the existing access control approaches in that (i) it integrates the fuzzy conditions together with other relevant contextual conditions into the access control policies for context-sensitive decision making and (ii) it can facilitate access control to data and resources from multiple sources by utilizing our proposed unified data and mapping models. We have presented the formal and ontology-based approaches to represent and reason about the fuzzy and other contextual conditions, and specify the access control policies by taking into account these conditions.

Furthermore, we have demonstrated the feasibility of our approach by considering the factors such as practicality and performance. In particular, we have developed a software prototype in order to assist the application developers in rapid prototyping. Using this prototype, software practitioners can build context-sensitive access control applications to cope with the complexities in the integration of fuzzy and other contextual conditions. Using this prototype, we have demonstrated the practicality of our approach by showing several case-based proof of concepts from healthcare domain. In addition, we have carried out a usability study and demonstrated a walkthrough of our whole mechanism. Finally, we have conducted two sets of experiments with our prototype and measured the query response time and scalability of our proposal. Both the prototype implementation and the experimental evaluations show that the new approach to access control using fuzzy logic is efficient and can be used in practice.

In this article, we have defined the membership functions using the necessary

information from the existing literature (e.g., the criticality ranges of the degree of membership for a "normal" health status are specified from 0 to 0.50). However, it may require *special modelling to define the membership functions*, which are domain dependent, and thus, further investigation to effectively represent them using the crisp boundary conditions is required in the future.

In this article, we have considered the context-sensitive access control to data and information resources from homogeneous cloud sources. Our proposed approach also can be applied to deal with the issue of *data heterogeneity* through accessing data from heterogeneous cloud sources. In such perspective, there is a need to investigate a generic data model to achieve semantic interoperability between heterogeneous data models from distributed cloud sources.

Further investigation is also required to demonstrate the feasibility of our proposal by considering an *empirical evaluation* of our proposed approach in this article with respect to our earlier approach [1]. One of the main goals is to show that our proposed CAAC approach with the benefits of unified data model and its associated policy and mapping models can be effectively used in practice to access information and data resources from multiple, distributed environments.

Another important research challenge is related to measure the performance overheads from the ontology complexity perspective. In this article, we have used OWL, DL and SWRL languages to model the data, policy and mapping ontologies. Currently, in both sets of experiments, we have measured the performance based on the different sizes of such ontologies. Future research can investigate the variations of performance overheads based on the single ontology versus multiple smaller different ontologies, considering reasoning time to map multiple data sources, reasoning time to derive relevant contextual conditions, loading time (single ontology versus multiple ontologies) and so on.

In our proposal, we have considered a general data model and its associated policy and mapping models to link multiple data sources and consequently access data resources from these different sources. However, accessing data and information resources from distributed sources has increasingly become chal-

lenging due to privacy issue. It is particularly important from the standpoint of integrating required data from different sources with the goal of privacy and utility trade-off. This is the case, for instance, in healthcare and military applications, where experts only want to share parts of the client's data they have. As a result, how to provide integrated results to the users by maintaining privacy of client's records is another key challenge that is required to be investigated in the future.

## References

## References

[1] Kayes, A., Rahayu, W., Dillon, T., Chang, E., Han, J.: Context-aware access control with imprecise context characterization through a combined fuzzy logic and ontology-based approach. In: CoopIS. (2017) 132–153

[2] Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**(7) (1993) 75–84

[3] Kayes, A.S.M., Han, J., Colman, A.: Ontcaac: An ontology-based approach to context-aware access control for software services. Comput. J. **58**(11) (2015) 3000–3034

[4] Kayes, A.S.M., Han, J., Colman, A.W.: An ontological framework for situation-aware access control of software services. Inf. Syst. **53** (2015) 253–277

[5] Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: *GEO-RBAC*: a spatially aware rbac. In: SACMAT. (2005) 29–37

[6] Joshi, J., Bertino, E., Latif, U., Ghafoor, A.: A generalized temporal role-based access control model. IEEE Trans. Knowl. Data Eng. **17**(1) (2005) 4–23

[7] Bonatti, P., Galdi, C., Torres, D.: Event-driven rbac. Journal of Computer Security **23**(6) (2015) 709–757

[8] Schefer-Wenzl, S., Strembeck, M.: Modelling context-aware rbac models for mobile business processes. IJWMC **6**(5) (2013) 448–462

[9] Hosseinzadeh, S., Virtanen, S., Rodríguez, N.D., Lilius, J.: A semantic security framework and context-aware role-based access control ontology for smart spaces. In: SBD@SIGMOD. (2016) 1–6

[10] Trnka, M., Cerný, T.: On security level usage in context-aware role-based access control. In: SAC. (2016) 1192–1195

[11] Kayes, A.S.M., Han, J., Colman, A.: An ontology-based approach to context-aware access control for software services. In: WISE. (2013) 410–420

[12] Kayes, A.S.M., Han, J., Colman, A., Islam, M.S.: Relboss: A relationship-aware access control framework for software services. In: CoopIS. (2014) 258–276

[13] Kayes, A.S.M., Han, J., Colman, A.: PO-SAAC: A purpose-oriented situation-aware access control framework for software services. In: CAiSE. (2014) 58–74

[14] Kayes, A.S.M., Han, J., Colman, A.: A semantic policy framework for context-aware access control applications. In: TrustCom. (2013) 753–762

[15] Almenárez, F., Marín, A., Campo, C., García, C.: Trustac: Trust-based access control for pervasive devices. In: SPC, Springer (2005) 225–238

[16] Cheng, P.C., Rohatgi, P., Keser, C., Karger, P.A., Wagner, G.M., Reninger, A.S.: Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In: IEEE Symposium on Security and Privacy, IEEE (2007) 222–230

[17] Takabi, H., Amini, M., Jalili, R.: Trust-based user-role assignment in role-based access control. In: AICCSA, IEEE (2007) 807–814

[18] Martínez-García, C., Navarro-Arribas, G., Borrell, J.: Fuzzy role-based access control. Information processing letters **111**(10) (2011) 483–487

[19] Feng, L., Dillon, T.S.: Using fuzzy linguistic representations to provide explanatory semantics for data warehouses. TKDE **15**(1) (2003) 86–102

[20] Colombo, P., Ferrari, E.: Towards virtual private nosql datastores. In: Data Engineering (ICDE), 2016 IEEE 32nd International Conference on, IEEE (2016) 193–204

[21] Colombo, P., Ferrari, E.: Fine-grained access control within NoSQL document-oriented datastores. Data Science and Engineering **1**(3) (2016) 127–138

[22] Bellahsène, Z., Bonifati, A., Rahm, E.: Schema matching and mapping. Springer (2011)

[23] Guo, S., Dong, X.L., Srivastava, D., Zajac, R.: Record linkage with uniqueness constraints and erroneous values. Proceedings of the VLDB Endowment **3**(1-2) (2010) 417–428

[24] Calbimonte, J.P., Corcho, O., Gray, A.J.: Enabling ontology-based access to streaming data sources. In: International Semantic Web Conference, Springer (2010) 96–111

[25] Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (iot): A vision, architectural elements, and future directions. Future generation computer systems **29**(7) (2013) 1645–1660

[26] Botta, A., De Donato, W., Persico, V., Pescapé, A.: On the integration of cloud computing and internet of things. In: Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, IEEE (2014) 23–30

[27] Rehman, Z.U., Hussain, O.K., Hussain, F.K., Chang, E., Dillon, T.S.: User-side qos forecasting and management of cloud services. World Wide Web **18**(6) (2015) 1677–1716

[28] Alhamad, M., Dillon, T., Chang, E.: A trust-evaluation metric for cloud applications. International Journal of Machine Learning and Computing **1**(4) (2011) 416–421

[29] Dillon, T.S., Wu, C., Chang, E.: Cloud computing: Issues and challenges. In: 24th IEEE International Conference on Advanced Information Networking and Applications, AINA. (2010) 27–33

[30] Zaghdoudi, B., Ayed, H.K.B., Harizi, W.: Generic access control system for ad hoc mcc and fog computing. In: International Conference on Cryptology and Network Security, Springer (2016) 400–415

[31] Salonikias, S., Mavridis, I., Gritzalis, D.: Access control issues in utilizing fog computing for transport infrastructure. In: International Conference on Critical Information Infrastructures Security, Springer (2015) 15–26

[32] Yu, Z., Au, M.H., Xu, Q., Yang, R., Han, J.: Towards leakage-resilient fine-grained access control in fog computing. Future Generation Computer Systems **78**(2) (2018) 763–777

[33] Zhang, P., Chen, Z., Liu, J.K., Liang, K., Liu, H.: An efficient access control scheme with outsourcing capability and attribute update for fog computing. Future Generation Computer Systems **78**(2) (2018) 753–762

[34] Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM (2012) 13–16

[35] Stojmenovic, I., Wen, S.: The fog computing paradigm: Scenarios and security issues. In: Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, IEEE (2014) 1–8

[36] Stojmenovic, I., Wen, S., Huang, X., Luan, H.: An overview of fog computing and its security issues. Concurrency and Computation: Practice and Experience **28**(10) (2016) 2991–3005

[37] Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1) (2001) 4–7

[38] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer **29** (1996) 38–47

[39] Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed nist standard for role-based access control. ACM TISSEC **4**(3) (2001) 224–274

[40] Riboni, D., Bettini, C.: OWL 2 modeling and reasoning with complex human activities. Pervasive and Mobile Computing **7** (2011) 379–395

[41] OWL: OWL 2 Web Ontology Language (W3C recommendation: 11 december 2012), https://www.w3.org/tr/owl2-overview/ (2017)

[42] SWRL: Semantic Web Rule Language, http://www.w3.org/submission/swrl/ (2017)

[43] De Bruijn, J., Lara, R., Polleres, A., Fensel, D.: OWL DL vs. OWL Flight: Conceptual modeling and reasoning for the semantic web. In: Proceedings of the 14th international conference on World Wide Web, ACM (2005) 623–632

[44] SWRLB: SWRL built-ins, http://www.daml.org/2004/04/swrl/builtins.html/ (2017)

[45] Protégé: Protégé-OWL API, http://protege.stanford.edu/ (2017)

[46] Jess: Jess rule engine, http://herzberg.ca.sandia.gov/ (2017)

[47] jFuzzyLogic: Fuzzy Concepts and Fuzzy Control System in Java, http://sourceforge.net/projects/jfuzzylogic (2017)

[48] ASCO: Australian standard classification of occupations: Health professionals, http://www.abs.gov.au/ (2017)

[49] HL7: Health level seven standard, http://www.hl7.org.au/ (2017)

[50] Kayes, A.S.M., Han, J., Colman, A.: OntCAAC: An ontology-based approach to context-aware access control for software services. Comput. J. **58**(11) (2015) 3000–3034

[51] Wong, A.K.Y., Wong, J.H.K., Lin, W.W.K., Dillon, T.S., Chang, E.: Semantically Based Clinical TCM Telemedicine Systems. Volume 587 of Studies in Computational Intelligence. Springer (2015)

[52] Chang, E., Hussain, F., Dillon, T.: Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence. John Wiley & Sons (2006)

[53] Kettouch, M., Luca, C., Hobbs, M.: Schema matching for semi-structured and linked data. In: ICSE. (2017) 270–271

[54] Getoor, L., Machanavajjhala, A.: Entity resolution: theory, practice & open challenges. Proceedings of the VLDB Endowment **5**(12) (2012) 2018–2019

[55] Koudas, N., Sarawagi, S., Srivastava, D.: Record linkage: similarity measures and algorithms. In: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, ACM (2006) 802–803

[56] Liu, X., Dong, X.L., Ooi, B.C., Srivastava, D.: Online data fusion. Proceedings of the VLDB Endowment **4**(11) (2011) 932–943

[57] Castano, S., De Antonellis, V.: Global viewing of heterogeneous data sources. IEEE Transactions on Knowledge and Data Engineering **13**(2) (2001) 277–297

[58] Gagnon, M.: Ontology-based integration of data sources. In: Information Fusion, 2007 10th International Conference On, IEEE (2007) 1–8

[59] Waingold, E., Taylor, M., Srikrishna, D., Sarkar, V., Lee, W., Lee, V., Kim, J., Frank, M., Finch, P., Barua, R., et al.: Baring it all to software: Raw machines. Computer **30**(9) (1997) 86–93

[60] Ylitalo, J., Nikander, P.: A new name space for end-points: Implementing secure mobility and multi-homing across the two versions of ip. In: 5th European Wireless Conference. (2004) 435–441

[61] Saurez, E., Gupta, H., Mayer, R., Ramachandran, U.: Demo abstract: Fog computing for improving user application interaction and context awareness. In: Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on, IEEE (2017) 281–282

**A. S. M. Kayes** is a Lecturer in Cyber Security in the Department of Computer Science and Information Technology, La Trobe University, Australia. He received his PhD from Swinburne University of Technology, Australia in 2014. His research interests include information modeling, context-aware access control, big data integration, IoTs, cloud and fog computing, advanced data analytics, fuzzy computation, security and privacy protection.

**Wenny Rahayu** is a Professor and the Head of School of Engineering and Mathematical Sciences at La Trobe University, Australia. Prior to this appointment, she was the Head of Department of Computer Science and Information Technology from 2012 to 2014. The main focus of her research is the integration and consolidation of heterogeneous data and systems to support a collaborative environment within a highly data-rich environment. In the last 10 years, she has published two authored books, three edited books and more than 150 research papers in international journals and conference proceedings.

**Tharam Dillon** is an adjunct Professor in the School of Engineering and Mathematical Sciences at La Trobe University, Australia. He has published eight authored books and more than 500 research papers in international journals and conference proceedings. His research works have been widely cited and therefore have considerable impact. Over the last few years, he has been cited several times in over 1000 scientific articles (source: Google Scholar). He has an H-index of 55 (Google Scholar) and over 14,000 citations, which put him in the

top percentile of researchers. He is a Fellow of the Institution of Electrical and Electronic Engineers (USA), Fellow of the Institution of Engineers (Australia), Fellow of the Safety and Reliability Society (UK), and Fellow of the Australian Computer Society.

**Elizabeth Chang** is a Professor of Logistics in IT at University of New South Wales, Canberra, Australia. She currently leads the Defence Logistics research group at UNSW Canberra, targeting the key issues in Logistics ICT, Big Data Management, Defence Logistics and Sustainment, Predictive Analytics, Situation Awareness, IoT and Cyber-Physical Systems, Trust, Security, Risk and Privacy. In a 2012 article, published in MIS Quarterly vol. 36 issue 4, she was listed fifth in the world for researchers in Business Intelligence. She has published 7 authored books and over 500 international journals and conference papers with an H-Index of 45 (source: Google Scholar) and she has over 11,000 citations.

**Jun Han** is a Professor of Software Engineering at Swinburne University of Technology, Australia. He is a software systems and services specialist with Swinburne's School of Software and Electrical Engineering. He is also the leader of the Networked Software Systems and Services research group, and a research leader with the Smart Services Cooperative Research Centre and the Cooperative Research Centre in Advanced Automotive Technology. His research interests span the areas of adaptive and context-aware software systems, cloud software systems, services engineering and management, and software and system architectures. He is also concerned with software security and performance as well as system integration, evolution and interoperability.