Context-Aware Access Control with Imprecise Context Characterization Through a Combined Fuzzy Logic and Ontology-Based Approach

A. S. M. Kayes[1], Wenny Rahayu[1], Tharam Dillon[1], Elizabeth Chang[2], and Jun Han[3]

[1]La Trobe University, Melbourne, Australia
[2]University of New South Wales, Canberra, Australia
[3]Swinburne University of Technology, Melbourne, Australia
{a.kayes,w.rahayu,t.dillon}@latrobe.edu.au,elizabeth.chang@adfa.edu.au,
jhan@swin.edu.au

Abstract. Context information plays a crucial role in dynamically changing environments and the different types of contextual conditions bring new challenges to 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. For example, a patient's current health status is "*critical*" or "*high critical*" which are imprecise fuzzy facts, whereas "*95% level of maximum blood pressure allowed*" is precise. Thus, there is a growing need for integrating these kinds of fuzzy and other conditions to appropriately control context-specific access to information resources at different granularity levels. Towards this goal, this paper introduces an approach to *Context-Aware Access Control using Fuzzy logic (FCAAC)* for information resources. It includes a *formal context model* to represent the fuzzy and other contextual conditions. It also includes 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 justify the feasibility of our approach by demonstrating the *practicality* through a prototype implementation and a healthcare case study, and also evaluating the *performance* in terms of response time.

Keywords: Context-aware access control, Fuzzy facts, Contextual conditions, Context model, Fuzzy reasoning model, Policy model

1 Introduction

Over the years, access control mechanisms have shifted from a fixed desktop environment to dynamic environments (e.g., pervasive, cloud and mobile computing environments) [1]. 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 information resources and

services from anywhere and at anytime fashion, even when they are on the move. In terms of *context-aware access control* systems [2, 3], 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 information resources and services. 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* 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%*"). Context-specific access control to information resources together with such conditions can provide an extra level of safety for patients in such emergency medical situations. In order to achieve *context-awareness* and integrate the different types of fuzzy and other contextual conditions into the access control processes, the following research issues need to be addressed.

(**R1**) How to derive precise contextual conditions from imprecise fuzzy facts for context-specific decision making?

(**R1**) 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?

Context-aware access control is a mechanism to determine whether a user's request to limit the access permissions to 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., [4]), the *temporal information* (e.g., [5]), the *event-driven information* such as surgery in progress (e.g., [6]), and other *environment context information* such as the range of IP addresses (e.g., [7–9]), 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 [2, 10], the *relationship context information* utilizing the process of inferring implicit knowledge [11], and the *purpose-oriented situation information* based on the currently available context information [3, 12]. We also propose a context-aware access control policy model in our earlier research [13], 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-aware access control approaches, several research works consider the use of fuzzy conditions (e.g., computing resource owners' trust degrees [14], quantifying risks [15], measuring trust levels [16], calculating user-permission strengths [17]) 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 [18], we develop our fuzzy model that is used in this paper to deal with imprecise context characterization.

The above-identified gap in the literature suggests that there is still a need for a new form of dynamic access control approach that can further limit the applicability of the available access permissions to information resources, integrating both the fuzzy facts and other contextual conditions together with access control policies for context-specific decision making. Our paper makes the following contributions towards achieving this goal.

(C1) ***Formal Access Control Approach:*** We introduce a new form of access control approach, Context-Aware Access Control using Fuzzy logic (FCAAC), specifically addressing the following aspects:

  (i) ***Context Representation and Reasoning Model:*** We present a formal analysis of the fuzziness of (imprecise) context information. We introduce a formal context model to represent the fuzzy and other contextual conditions from the low-level information.

  (ii) ***Policy Model:*** We present a formal analysis of the context-specific access control decision making by taking into account the relevant fuzzy and other contextual conditions.

(C2) ***Ontology-based FCAAC Approach:*** Using our formal context and policy models, we introduce an ontology-based approach to model and reason about the relevant fuzzy and other contextual conditions, and consequently model the context-specific access control policies, incorporating the relevant conditions into the access control processes.

(C3) ***Evaluation:*** Other than the above two main contributions, we justify the feasibility of our approach by demonstrating the following factors:

(i) ***Practicality:*** We develop a prototype of the FCAAC approach that assists software practitioners in rapid prototyping. Using this prototype, a case study from the healthcare domain is presented which demonstrates the practicality of the proposed approach.

(ii) ***Performance:*** We conduct two sets of experiment in a healthcare environment and evaluate the applicability of our access control approach by means of response time.

The rest of this paper is organized as follows. We first present an application scenario in Section 2 to motivate our work. Section 3 introduces our formal access control approach, including the context representation and reasoning model and its associated policy model. Using the formal context and policy models, Section 4 introduces an ontology-based access control approach. Section 5 demonstrates the practicality of our approach against a healthcare case study and the performance in terms of response time. Section 6 briefly presents the related work. Finally, Section 7 concludes the paper and outlines future work.

2 Significance of Our Research and General Requirements

This section presents an extended application scenario from our earlier work [2]. In addition, we identify the general requirements of developing a new access control approach by integrating both the fuzzy conditions and other contextual conditions together with access control policies.

2.1 Application Scenario

Let us consider our extended healthcare scenario where *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 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.*

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.

2.2 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 and a software prototype* to incorporate these conditions into the access control policies.

3 Our Formal FCAAC Approach

In this section, we introduce an approach to Context-Aware Access Control using Fuzzy logic (FCAAC), including context and policy models.

Figure 1 presents the conceptual FCAAC approach, which includes 3 basic steps: capture low-level data, derive relevant information (fuzzy and other contextual conditions) and make access control decision. Stage 1 is the process of gathering low-level

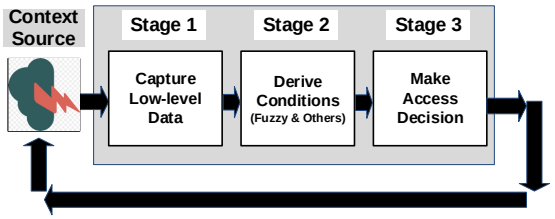

Fig. 1: Our FCAAC Approach

data from the context sources. Stage 2 is the process of inferring relevant fuzzy and other contextual conditions from the low-level data. Finally, stage 3 is the process of making access control decision based on the relevant conditions. In the following, we present a formal analysis of the 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.

Representation of Fuzzy and Normal Contextual Conditions: In the literature, many researchers have defined the context information. The most well accepted definition is given by Dey [19], *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 [18], 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.*

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

Reasoning about Fuzzy and Normal 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 [20] is an emerging model of access control and is well recognized for its many advantages in large-scale authorization management [21]. 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 [2] 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

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
$canAccess(u_1, p_1)$

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 ($cc \in CC,\ CC = FC \cup NC$) and '$p$' represents a permission ($p \in P$), then, together the elements '$Users$' ($U = \{u_1, u_2, ..., u_m\}$), '$Roles$' ($R = \{r_1, r_2, ..., r_i\}$), '*Contextual Conditions*' ($CC = \{cc_1, cc_2, ..., cc_j\}$) and '$Permissions$' ($P = \{p_1, p_2, ..., p_n\}$) form the *FCAAC Policy Model*.

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 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, $cc_1 = nc_1 \vee fc_1$.

Further details of the FCAAC policy specification using ontology-based languages are discussed in the following section (see Section 4.3).

4 Ontology-based FCAAC Approach

This section introduces an ontology-based approach, to realize the formal models.

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 [22] 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) [23]. 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 [22]. 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 [24] to the FCAAC ontology.

The FCAAC ontology, as depicted in Figure 2, 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 [2] 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.

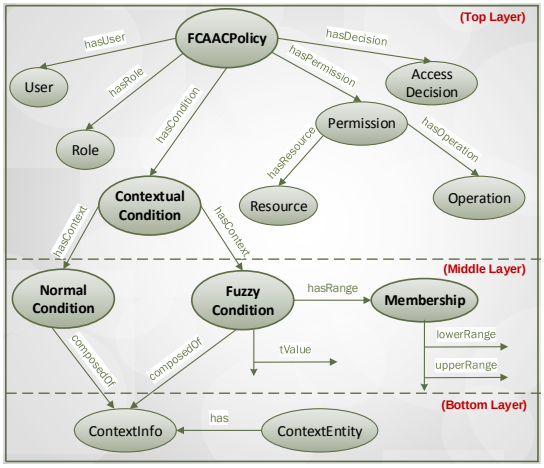

Fig. 2: The FCAAC Ontology

The detailed representation of a wide range of context information is out of the scope of this paper. In our earlier research [2, 3, 11], we have already introduced context ontologies to represent and model the access control-specific context information (e.g., the interpersonal relationships, the situations).

4.1 Modelling Contextual Conditions

The middle layer in Figure 2 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 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) 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 *NormalCon-*

Table 2: 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"**)

dition and *FuzzyCondition* classes. 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 concepts (*ContextInfo*) from the core context ontology, which is already introduced in our earlier work [2, 10]. The object property *hasRange* is used to link the classes *FuzzyCondition* and *Membership*.

The *FuzzyCondition* class contains a '*xsd:float*' type data property named *tValue*, 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*, 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 status is "*normal*", which has a *lowerRange* of criticality 0 and an *upperRange* of criticality 0.50.

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.

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 built-ins functions. An example reasoning rule to derive the interpersonal relationship between user and patient is specified in Table 2. The interpersonal relationship is inferred from the low-level context information which is already represented in our context ontology [2, 10], i.e., from the user's personal profile and the patient's social profile information.

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 3. The first rule in Table 3 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 3 shows our fuzzy context information system, which includes three main steps for mapping between crisp and fuzzy

Table 3: 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***	$PCHState(Critical)$

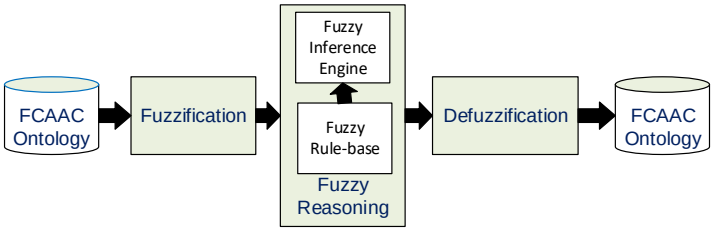

Fig. 3: The Fuzzy Context Information System

datasets: fuzzification, fuzzy reasoning and defuzzification [18]. 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 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 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 2). 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.

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). The FCAAC policy rule in OWL is shown in the top part in Table 4 (the core policy concepts are specified in *Line #1 to 7*), including the definition of contextual condition (which is defined in *Line #8 to 20*). The bottom part in Table 4 shows the specification of contextual conditions and other policy constraints (e.g., fuzzy conditions, role identity) in SWRL (where the main conditions/constraints are represented in bold type). The user and role specifications are shown in *Line #21 to 22*, the permission specification is shown in *Line #23 to 26*, the contextual condition construction is specified in *Line #27*, the normal condition

Table 4: An Example Policy in Ontology format for the Registered Nurses

1	<**FCAACPolicy** rdf:ID="fcaac$_1$">
2	<hasUser rdf:resource="#User_canPlay_RN"/>
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	<owl:Class rdf:ID="**ContextualCondition**">
9	<owl:ObjectProperty rdf:ID="**hasContext**">
10	<rdfs:domain rdf:resource="#ContextualCondition"/>
11	<rdfs:range>
12	<owl:Class>
13	<**owl:unionOf** rdf:parseType="Collection">
14	<owl:Class rdf:about="#**NormalCondition**"/>
15	<owl:Class rdf:about="#**FuzzyCondition**"/>
16	</**owl:unionOf**>
17	</owl:Class>
18	</rdfs:range>
19	</owl:ObjectProperty>
20	</owl:Class>
21	FCAACPolicy(?fcaac$_1$) ∧ User(?u) ∧ hasUser(?fcaac$_1$, ?u) ∧ Role(?r) ∧
22	hasRole(?fcaac$_1$, ?r) ∧ canPlay(?u, ?r) ∧ roleIdentity(?r, "**RN**") ∧
23	Permission(?per) ∧ hasPermission(?fcaac$_1$, ?per) ∧ Resource(?res) ∧
24	hasResource(?per, ?res) ∧ resourceIdentity(?res, "**DMR**") ∧
25	Owner(?o) ∧ isOwnedBy(?res, ?o) ∧ Operation(?op) ∧
26	hasOperation(?per, ?op) ∧ action(?op, "**Write**") ∧
27	**ContextualCondition**(?**cc**$_1$) ∧ hasCondition(?fcaac$_1$, ?cc$_1$) ∧
28	**NormalCondition**(?**nc**$_1$) ∧ **hasContext**(?cc$_1$, ?nc$_1$) ∧
29	InterpersonalRelationship(?rel) ∧ hasRelationship(?u, ?rel) ∧
30	hasRelationship(?o, ?rel) ∧ interRelationship(?**rel**, **"assignedNurse"**) ∧
31	RequestTime(?rt) ∧ hasRequestTime(?u, ?rt) ∧ requestTime(?**rt**, "**dutyTime**")
32	∧ Co-locatedRelationship(?col) ∧ hasRelationship(?u, ?col) ∧
33	hasRelationship(?o, ?col) ∧ isColocatedWith(?**col**, **yes**) ∧
34	**composedOf**(?nc$_1$, ?**rel**) ∧ **composedOf**(?nc$_1$, ?**rt**) ∧
35	**composedOf**(?nc$_1$, ?**col**) ∧
36	**FuzzyCondition**(?**fc**$_1$) ∧ **hasContext**(?cc$_1$, ?fc$_1$) ∧ **PCHState**(?hs) ∧
37	**composedOf**(?fc$_1$, ?**hs**) ∧ **swrlb:equal**(?hs, "**normal**") ∧ **tValue**(?fc$_1$, ?tv) ∧
38	Membership(?m) ∧ hasRange(?fc$_1$, ?m) ∧ lowerRange(?m, ?lr) ∧
39	**swrlb:equal**(?lr, **0**) ∧ upperRange(?m, ?ur) ∧ **swrlb:equal**(?ur, **0.50**) ∧
40	**swrlb:greaterThan**(?tv, lr) ∧ **swrlb:lessThan**(?tv, ur) ∧
41	AccessDecision(?dec) ∧ hasDecision(?fcaac$_1$, ?dec) → decision(?dec, "**Granted**")

composition is specified in *Line #28 to 35*, the fuzzy condition composition is specified in *Line #36 to 40*, and the access decision is specified in *Line #41*. In the previous section, an example SWRL-based reasoning rule in Table 2 is used to determine the user and patient have a '*assignedNurse*' relationship, and an example set of fuzzy logic-based reasoning rules in Table 3 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 [2].

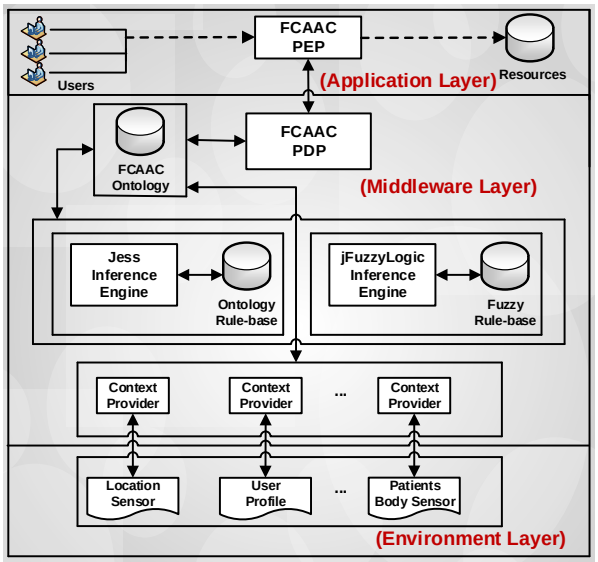

Fig. 4: Overview of the Prototype Architecture of FCAAC

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 2). For example, in the above policy, Mary can access 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.

5 Prototype and Evaluation

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 approach. In particular, we present a case study from the healthcare domain to demonstrate the practicality of our access control approach. Furthermore, the deployment of *eHealthcare* application is performed for measuring the performance of our approach. The performance results are presented in Section 5.2.

5.1 Practicality

Prototype: Figure 4 shows an architecture of the software prototype, which extends our earlier prototype [2], utilizing both the fuzzy logic and ontology-based reasoning capabilities. It includes environment, middleware and application layers. The environment layer includes the sensors or data sources and the middleware layer includes the context provides, FCAAC ontology, context reasoner and access control processor. The context providers receive the raw context facts from the data sources and the FCAAC ontology captures the low-level information

from the context providers. The access control policies are also stored in FCAAC ontology. The context reasoner derives the relevant contextual conditions by using the information from the ontology. The access control processor includes the FCAAC PDP (policy decision point), which is implemented in Java to determine the access request is "granted" or "denied", according to the applicable policies and the necessary contextual conditions. The application layer includes the FCAAC PEP (policy enforcement point), which forwards the request to the FCAAC PDP. The detailed implementation of the context providers, PEP and PDP can be found in our earlier prototype [2]. We in this paper mainly discuss the implementation of the context reasoner to derive the contextual conditions.

The FCAAC ontology is defined by using ontology languages OWL [23] and SWRL [24], and the ontology has been generated with the Protégé-OWL graphical tool [25]. We develop an ontology rule base to derive the normal contextual conditions from the low-level information 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 [26] to facilitate reasoning tasks for executing such rules. 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 [27], which is written in Java. We have already shown the fuzzy reasoning processes in Figure 3. In order to execute these reasoning rules and consequently derive the implicit information (normal and fuzzy conditions), we have implemented a context reasoner in Java. In particular, we have implemented two Java functions, the first function is used to execute the reasoning rules and infer the implicit information using low-level data from the FCAAC ontology, and the other function is used to transfer the inferred information in the ontology.

Case Study: 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 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 4, 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.

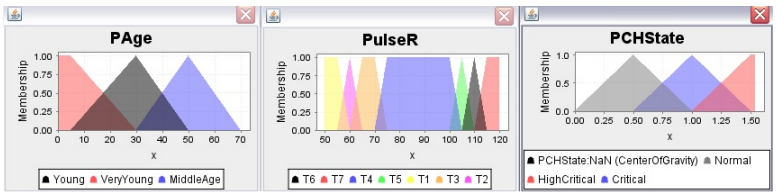

Fig. 5: Inputs and Output Membership Functions

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 sets: *Normal*, *Critical* and *HighCritical*. Based on the experience from our group's earlier research on fuzzy linguistic representations [18], these input and output fuzzy sets are characterized by triangular and trapezoidal membership functions (see Figure 5) and Mamdani's center of gravity (COG) method in conjunction with max-min inference is used for fuzzy reasoning (see Figure 6). We have specified 21 linguistic rules to cover all the possible values of *PAge* and *PulseR*.

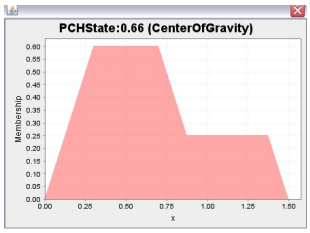

Fig. 6: PCHState

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 Table 3. Finally, Bob's *PCHState* is derived using the COG max-min inference method (see Figure 6). 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% 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). In summary, the purpose of the case scenario and prototype testing is to provide a walkthrough of the whole FCAAC approach.

5.2 Performance

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. The conducted tests are carried out in a Windows PC with an Intel Core i7@3.6GHz Processor and 16GB of RAM. The results have been obtained by executing the experiments 10 times and computing their arithmetic mean.

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 7 depicts how the response time varies, measured in milliseconds (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.

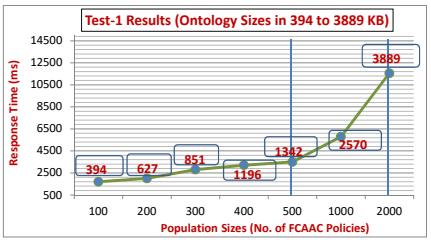

Fig. 7: Populations vs Response Time

The FCAAC reasoning model based on the fuzzy and ontology-based inference rules is one of the important parts of our proposed 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 8 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 8.

Fig. 8: Stages of Response Time

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 8), 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 7). 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 [28], we may adopt RDF language to build a new approach as an alternative of using OWL language.

6 Related Work and Discussion

This section provides a short overview of the relevant access control approaches.

Context-Aware Access Control Approaches: 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. [5] have proposed a role-based access control (RBAC) approach and incorporated the temporal information into the RBAC policies. Bertino et al. [4] 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. [6] 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 [7] have proposed a context-aware RBAC approach to ubiquitous systems, incorporating the context information such as time and location into the policies. Similar to [7], Hosseinzadeh et al. [8] and Trnka and Cerný [9] 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 [2, 10], 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 [11], 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 [3, 12], 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.

Fuzzy Logic-Based Access Control Approaches: Different access control approaches have been proposed in literature to model policies based on involving the fuzzy conditions, which can be derived from the fuzzy sets.

In [14], the authors have proposed a trust-based access control approach based on the trust values [29], 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. [15] 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. [16] 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 [17], 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.

Discussion: Following the traditional context-aware RBAC approaches, they are not adequate to derive the fuzzy conditions from the low-level fuzzy facts and incorporate them into the access control policies for decision making. On the other hand, the fuzzy logic-based approaches are not context-aware and robust enough to capture and derive the dynamically changing contextual conditions from the low-level information. In this respect, different from these exist-

ing access control approaches, our proposed FCAAC 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-specific decision making at different granularity levels.

7 Conclusion and Future Research

The FCAAC approach described in this paper represents a flexible policy specification solution to the problem of incorporating fuzzy contextual conditions, in the domain of access control to information resources utilizing the benefits of fuzzy sets. Our approach significantly differs from the existing access control approaches in that it integrates the fuzzy conditions together with other relevant contextual conditions into the access control policies for context-specific decision making. 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 engineers in rapid prototyping. Using this prototype, software practitioners can build context-specific 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 a case-based proof of the applicability of the FCAAC concepts against a healthcare case study. In addition, we have conducted two sets of experiment with our prototype and measured the response time and scalability of our proposal. Both the prototype implementation and the performance analysis results show that the new approach to access control using fuzzy logic is efficient and can be used in practice.

In this paper, 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.

References

1. Weiser, M.: Some computer science issues in ubiquitous computing. Commun. ACM **36**(7) (1993) 75–84
2. 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
3. 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
4. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: *GEO-RBAC*: a spatially aware rbac. In: SACMAT. (2005) 29–37
5. 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

6. Bonatti, P., Galdi, C., Torres, D.: Event-driven rbac. Journal of Computer Security **23**(6) (2015) 709–757
7. Schefer-Wenzl, S., Strembeck, M.: Modelling context-aware rbac models for mobile business processes. IJWMC **6**(5) (2013) 448–462
8. 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
9. Trnka, M., Cerný, T.: On security level usage in context-aware role-based access control. In: SAC. (2016) 1192–1195
10. 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
11. 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
12. 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
13. Kayes, A.S.M., Han, J., Colman, A.: A semantic policy framework for context-aware access control applications. In: TrustCom. (2013) 753–762
14. Almenárez, F., Marín, A., Campo, C., García, C.: Trustac: Trust-based access control for pervasive devices. In: SPC, Springer (2005) 225–238
15. 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
16. Takabi, H., Amini, M., Jalili, R.: Trust-based user-role assignment in role-based access control. In: AICCSA, IEEE (2007) 807–814
17. Martínez-García, C., Navarro-Arribas, G., Borrell, J.: Fuzzy role-based access control. Information processing letters **111**(10) (2011) 483–487
18. Feng, L., Dillon, T.S.: Using fuzzy linguistic representations to provide explanatory semantics for data warehouses. IEEE Trans. Knowl. Data Eng. **15**(1) (2003) 86–102
19. Dey, A.K.: Understanding and using context. Personal Ubiquitous Computing **5**(1) (2001) 4–7
20. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer **29** (1996) 38–47
21. 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
22. Riboni, D., Bettini, C.: OWL 2 modeling and reasoning with complex human activities. Pervasive and Mobile Computing **7** (2011) 379–395
23. OWL: Web ontology language, http://www.w3.org/2007/owl/ (2017)
24. SWRL: Semantic web rule language, http://www.w3.org/submission/swrl/ (2017)
25. Protégé: Protégé-OWL API, http://protege.stanford.edu/ (2017)
26. Jess: Jess rule engine, http://herzberg.ca.sandia.gov/ (2017)
27. jFuzzyLogic: Fuzzy Concepts and Fuzzy Control System in Java, http://sourceforge.net/projects/jfuzzylogic (2017)
28. 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)
29. 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)