Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 836546, 12 pages
http://dx.doi.org/10.1155/2015/836546

Research Article

CBDAC: Context-Based Dynamic Access Control Model Using Intuitive 5W1H for Ubiquitous Sensor Network

Jiseong Son,[1] Jeong-Dong Kim,[1] Hong-Seok Na,[2] and Doo-Kwon Baik[1]

[1] *Department of Computer and Radio Communications Engineering, Korea University, Seoul 136-713, Republic of Korea*
[2] *Department of Computer and Information Communication, The Cyber University of Korea, Seoul 136-713, Republic of Korea*

Correspondence should be addressed to Jeong-Dong Kim; kjdvhu@gmail.com and Doo-Kwon Baik; baikdk@korea.ac.kr

Received 5 December 2014; Accepted 9 April 2015

Academic Editor: Naveen Chilamkurti

Currently, access control is facing many issues for information protection in the ubiquitous sensor network (USN) environment. In particular, dynamic access control is a central problem where context always changes because of volatile ubiquitous sensors. The use of context is important in USN. In this paper, we focus on the context-driven privacy protection model. In context-based access control research, the access permission technique that uses context is being intensely investigated because of the ease with which various dynamic access permissions can be assigned in accordance with the various changes in context. A key feature of this approach is dynamic access control. Therefore, we propose a model for privacy preservation that is context-based dynamic access control that uses intuitive 5W1H for USN. According to this model, the access control strategy can be determined dynamically based on context elements and subject attributes, in addition to objects and operations, using access control entities; therefore, it is relatively easy to infer the dynamic access control of context expressivity both accurately and efficiently.

1. Introduction

Ubiquitous computing is an environment unrestricted by spatiotemporal conditions [1]. It allows computing among humans, objects, and information through the interaction of diverse sensors immersed in the ubiquitous sensor network (USN) environment [2]. The context is information produced in the diverse ubiquitous sensors, and such information provides various services through an intersystem information exchange in various domains and interaction. Such context is volatile information strongly influenced by environmental elements, such as time and space. Therefore, in order to process such information, a dynamic processing technology is required in USN. Furthermore, because the access permission to entities that will perform the service according to the changing context is also volatile, an approach control model to accommodate the dynamic change is required. In other words, despite the fact that the same user gains access to a certain system, depending on the surrounding context (e.g., time, space, and user), the user access authorization can change dynamically.

There are already preexisting studies on various technologies and techniques to process context [3–10]. Many of the studies offered ontological-modeling-based techniques for the processing of semantic context [7–10]. The main issue of the ontology-based context-aware model is the maintenance of interoperability between the modeling level and context information. The modeling level for context recognition is divided into tightly and loosely coupled pervasive system modeling. Tightly coupled pervasive system modeling can provide high-performance services that can be specialized in certain domains, but additional time and cost are incurred for the additional processes necessary for analyzing the context information of different domains. Loosely coupled pervasive system modeling is advantageous in that it is independent of the domain and utilizes a variety of context information. However, it is less appealing because it is not adequate for processing inferences or providing the high-performance service required for processing context application (generating context B through context A) and context combination (context B + context C = context D in context A) processing.

In order to remedy such problems, in a previous research [11] that we conducted, we proposed an intuitive ontology-based CA$_{5W1H}$Onto model that supports semantic context through the use of 5W1H. In [11], in order to express context information, ontological concepts and properties were utilized for defining the semantic context and a wide range of standardized relationships. Such applications are very advantageous in terms of adaptability and interoperability with the ontologies already developed in diverse domains. Because of these features, the CA$_{5W1H}$Onto model exhibits high levels of expandability and recyclability.

Role-based access control (RBAC) [12–14] is the most representative access control model. Recent studies proposed various extended RBAC (such as relationship-based, purposed-based, and context-based). The basic RBAC concept is as follows: permissions are assigned to functional roles within an enterprise or individual user, and then the necessary permissions are authorized by assigning them to a role or a set of roles [15]. The role-based model grants (or denies) a subject access to data regardless of the request context. In addition, a privacy policy is mainly concerned with which data object is used for which purposes. Thus, purpose is a central concept in many privacy-protecting access control models [16–19].

Unfortunately, privacy protection cannot be readily conducted by traditional RBAC models. The first reason is that whereas traditional RBAC models focus on which user is performing which action on which data object, privacy policies are concerned with which data object is used [20]. Another reason for the difficulty of privacy protection is that the comport level of data usage varies from individual to individual [15]. In addition, in order to process the dynamic context change in role- and purpose-based research, conditions or constraints were defined to support the changing access control. Nevertheless, conditions or constraints are defined only in part by the developer; hence, they have limitations in covering/supporting the various context changes required in a ubiquitous computing environment. In this paper, we focus on the context-driven privacy protection model. In fact, context and access control are essentially interdependent. In context-based access control research, an access permission technique that uses context is being intensely investigated because of the ease with which various dynamic access permissions can be assigned in accordance with the various changes in context [21–29].

In this paper, we propose a CA$_{5W1H}$Onto-based dynamic access control (CBDAC) model for the USN environment. The proposed model makes use of the ontological concept using 5W1H to process the context information. Furthermore, the proposed model guarantees privacy protection when processing the context information in various domains and assigns access permission without any limitation to certain domains. In other words, access permission is assigned dynamically according to the change in context information such that, even for a subject with the same role, access permission is defined differently depending on the context information and access condition. Not only does the proposed model authorize access based on roles, which is the key concept of RBAC, but also dynamic access control is also possible because access permission is assigned based on context information, access condition, and intended goal.

Consequently, we address this goal by presenting a comprehensive approach to dynamic access control management, which is the fundamental problem on which context-aware access control can be developed. In our paper, the purpose term has been superseded by goal. Furthermore, our notion of role attributes is closely related to the notions presented in [30, 31] in that it allows the specification and enforcement of context-based policies in RBAC. However, we build upon and further elaborate the existing notions with the presence of conditional roles with context-aware entities in order to achieve fine-grained administrative access control.

The remainder of this paper is organized as follows. Section 2 describes the background and related work; the model of the CBDAC that formally defines the notation of access permission is illustrated in Section 3; Section 4 describes its use case scenario and implementation. Finally, in Section 5, we suggest the conclusion and future work.

2. Background and Related Work

2.1. CA$_{5W1H}$Onto. Context refers to a special form of knowledge and, for this reason, it constitutes an important modeling requirement that is the tradeoff between expressiveness and complexity. To resolve the tradeoff issue that concerns context-aware modeling, ontology (i.e., OWL-DL) may be utilized. The ontological concept is an important component in determining the expressiveness of knowledge and reasoning capability for context awareness. Ontology amply expresses concepts and their relationships and automatic reasoning in processing context based on the expressive capacity. The model CA$_{5W1H}$Onto is a context-aware ontological model based on the five Ws and one H (5W1H) [11]. Furthermore, CA$_{5W1H}$Onto is a method for interpreting and abstracting semantic context designed to support the intuitive integration of different context-aware schemas, to which the maxim (such as why, who, what, where, when, and how) is applied [11].

Figure 1 shows the key elements that constitute the CA$_{5W1H}$Onto model that consists of <*Concept, Instance, Context*> triples, where the first two elements of the triple set utilize the properties defined through the existing ontology. The Context element carries all six attributes of the maxim. The element of Context contains various contextual activities. In turn, the Context component helps define the basic characteristics of the maxim and the schemas utilized among ontological concepts. The CA$_{5W1H}$Onto model proffers services tailored for a specific time, space, and set of user preferences across different domains. The model performs modeling of the essential elements by defining, in accordance with the maxim, the context required for the integration and interoperability of the defined contextual information.

The CA$_{5W1H}$Onto model defines, in the unit of <*Concept, Instance, Context*>, the ontological elements (e.g., *concept*, *instance*, *datatype*, *data property*, and *object property*) for context-aware definition. By defining each in an independent component module, adaptability and independence

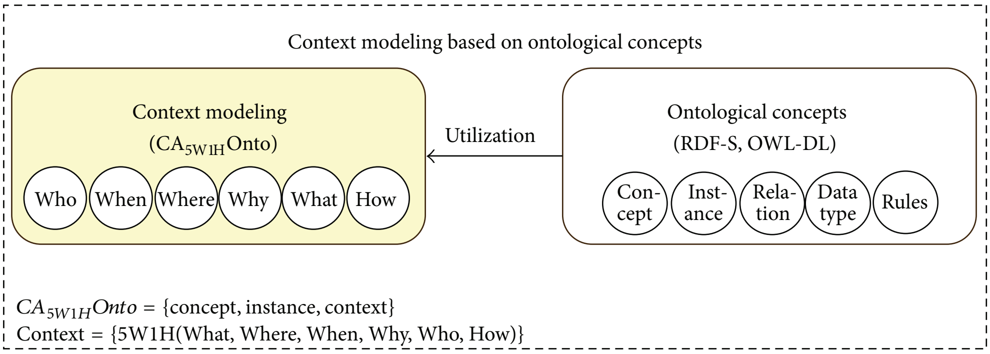

Context modeling based on ontological concepts
Context modeling
(CA_{5W1H}Onto)
Who
When
Where
Why
What
How
Utilization
Ontological concepts
(RDF-S, OWL-DL)
Con-cept
Inst-ance
Rela-tion
Data type
Rules
$CA_{5W1H}Onto$ = {concept, instance, context}
Context = {5W1H(What, Where, When, Why, Who, How)}

Figure 1: Conceptual model of CA_{5W1H}Onto.

are guaranteed when developing a context-aware model applicable to diverse domains. In other words, the maxim-applied context-aware modeling technique is an intuitive model in nature and thereby allows interoperability between systems or models throughout integration and sharing of schemas that belong to diverse domains. Detailed explanation ensues hereunder as to how to map between the properties of the maxim and the context-aware elements defined by the CA_{5W1H}Onto model.

One of the merits of ontologically modeling contextual information lies in its ability to automatically extract new knowledge on current context, in addition to providing ample formalism with streamlined expressiveness about the knowledge.

2.2. RBAC. RBAC was proposed by Sandhu et al. in 1996 [32]. The fundamental idea of RBAC is to authorize data access based on user role. Role means the function of the user or the organization, and such role and user have an N:N relationship. In other words, the user can have various different roles, and one role can be assigned to several users. The relationships among roles are defined through the role hierarchy. RBAC can be applied to various environments or applications through simple role-based access control, where the role attributes are associated with the roles in order to enforce global constraints such as the principle of the separation of duty [33]. That is, there is a limitation to providing privacy protections by defining conditions flexibly based on the attributes of the role (e.g., conditional activation, role deactivation, and role membership qualification). RBAC can be divided into the more detailed attributes of roles, permissions, users, and sessions, and a more diverse definition of constraints is necessary [8, 9]. In addition, because the role hierarchy is a predefined static role, in dynamic environments such as context-aware, wireless computing, and ubiquitous computing environments, where a volatile context is produced, instead of determining the access control permission based on simply on role, dynamic access control is required for the process of such a volatile context.

Purpose-based Access Control (PBAC) [34] defines the access control model using not only an RBAC-based role, but also purpose as a key concept. Purpose describes the reasons for data collection and data access. PBAC defines the relationships among purposes with the purpose hierarchy (purpose tree) and, based on this, the intended and access purposes are specified. Access purpose specifies the purpose for the access when a request for data access is made. When the user makes a request for data access, the access control engine compares the access and intended purposes of the data requested by the user and then verifies whether the user has access authorization. However, the access purpose of the PBAC is declared by the users, which implies low flexibility, and the storage of privacy metadata is based on labeling schemes, which is triggered overhead.

Dynamic Purpose-based Access Control (DPBAC) [35] includes a dynamic concept in the PBAC model. In order to implement strengthened privacy preservation, DPBAC separates the access purpose authorization from the access decision. For the protection of privacy data, the data provider predefines the intended purpose (AIP or PIP). In contrast, the data owner holds the responsibility for the policy that manages the authorization of the access purpose. DPBAC defines the conditional role and supports the preexisting RBAC and other dynamic access controls. Because the conditional role compares the predefined static role based on subject attributes and context attributes of the system, it dynamically determines the access purpose and purpose compliance.

Conditional Purpose-based Access Control Model with Dynamic Role (CPBAC) [20] makes use of the preexisting PBAC and DPBAC to support the conditional role. In CPBAC, the conditional intended purpose is proposed so that data access is permitted only when a certain purpose satisfies some conditions. In order to fulfill such a requirement, the data provider should predefine the intended purpose for the protection of the privacy data and set the scope of the data to be made publicly.

2.3. Context-Based Access Control Models. The concept of the basic RBAC model grants or denies role access to data regardless of the requested context. However, the requested context affects the decision to access objects or systems. In other words, context information provides significant access control parameters. For example, in mobile banking systems, environmental context such as location, time, and others could affect the access decision to grant or deny data access to the account or operating function of the banking system. For this reason, extended access control models based on RBAC were proposed to support context information [21–29].

CRBAC [21] is a contextual role-based access control authorization model for electronic patient records (EPRs). Contextual authorizations use the environmental information available at access time, such as user/patient relationship, in order to determine whether a user is allowed to access an EPR resource. This model extends RBAC by data-access rules for processing the context of large-scale healthcare. The data-access rules are defined by a five-tuple <*Role, Privilege-Type, Operation, Object, Authorization-Type*>. The five-tuple represents more expressions than the basic RBAC. However, a logical expression of CRBAC is difficult for the modeling that uses data-access rules. Another proposed contextual extension of RBAC is the Attribute-Based Access Control (ABAC) model [22]. The ABAC model authorizes or denies service access based on the attributes communicated by the subject. In order to specify the attribute-based policies, a data structure that uses algebraic operations was proposed. Geo-RBAC is an access control model for processing spatial and location-based information by expanding RBAC [23]. In Geo-RBAC, spatial entities are used for modeling objects, user positions, and geographically bounded roles. A physical position includes the information provided from mobile devices or smartphones and logical, device-independent position information such as roads, villages, buildings, or locations. Geo-RBAC is a flexible model for spatial information processing with high reusability. Generalized Temporal Role-Based Access Control (GTRBAC) [24] proposed the access control model to consider the time context and offered an extended RBAC model capable of expressing a wider range of temporal constraints. In particular, this model is designed possibly not only for period constraints, but also for the regular expressions of roles, user-role assignments, and role-permission assignments. The CAP model [25] is the access control model for resources in pervasive computing environments. This model consists of a two-step access control with the user session registering to the domain authority and the session agent self-governing access through the session permission assignment database. Dynamic Role-Based Access Control model (DRBAC) [26] extends the basic RBAC model to support the dynamic context information. This model dynamically adjusts the static role and permission assignments based on context information but depends on a central authorizer to change the active role of the user's agent according to context change. Carminati et al. [27] proposed an access control model for the purpose of controlling information sharing in a web-based social network. This model uses the rule-based approach to specify access policies. The context of the certified users is defined by the type, depth, and trust level of the relationship among the nodes in the network. The difference between such a model and the conventional access control system is that access control enforcement is conducted on the client side through semidecentralized architecture. RelBac [28] proposed permissions as being relationships between subjects and objects, where subjects and objects are entity sets, and permissions are a relationship set. The novel idea that this model presents is that, for the process of dynamic contexts, it formalizes the binary relationship between subjects and objects as permissions. SitBAC [29] is a model that utilizes those situations that define context elements and attributes for applying context to the access control. Furthermore, in this research, an inference is supported with OWL-DL and SWRL. However, SitBAC is specialized to a domain called healthcare; hence, it has limitations with respect to the processing context of other domains.

3. CBDAC Model

In this section, we propose an overall structure and formalized definition for the CBDAC model, which is the extended form of RBAC, during the application of ontology technologies. It is ultimately aimed at guaranteeing the dynamic access condition of the data access control.

Based on the RBAC model, the CBDAC model extends mainly in the aspects described following Figure 2. Figure 2 shows the proposed CBDAC model that consists of the *Profile Manager, Ontological Concepts, Context Manager, Concepts*, and *Access Permission*. In particular, at the research stage of privacy protection in the CBDAC model, the *Access Conditions* and *Intended Goal* (that consists of *Allowable Intended Goal*, *Conditional Intended Goal*, and *Prohibited Intended Goal*) into *Access Permission* are significant.

(i) *Profile Manager* defines and manages the information on *Profile* and *Roles* with regard to *Subject* in order to generate context. The key concept of the RBAC model is *Role*, which represents a certain specific job function in an organization. A *Role* is assigned to the *Subject*, such as RBAC, by the *Profile Manager*. In other words, the *Role* of the *Subject* represents a working position or working function of the user/system assigned within the *Profile Manager*.

(ii) *Ontological Concept* is defined by the ontological elements (e.g., *concept*, *instance*, *datatype*, *data property*, and *object property*). It is utilized for managing context in the *Context Manager*.

(iii) *Context Manager* considers the context of a subject based on $\mathrm{CA}_{5\mathrm{W1H}}$Onto by applying 5W1H (Why, Who, What, Where, When, and How). The $\mathrm{CA}_{5\mathrm{W1H}}$Onto model is designed to support intuitive integration of different context-aware schemas, to which the 5W1H is applied. The $\mathrm{CA}_{5\mathrm{W1H}}$Onto model consists of <*Concept, Instance, Context*> triples. *Concept* and Instance are defined by ontological concepts described through the existing ontology. *Context* is the entities of concept and instance (such as time, location, user profile, and access goal) defined by 5W1H.

(iv) *Concepts* consist of *Subject* and *Object* by ontological concepts. The *Subject* can be a user, application, or agent; that is, the *Subject* is various context elements generated by ubiquitous sensors. *Object* is the target data, system, or services that the *Subject* requests.

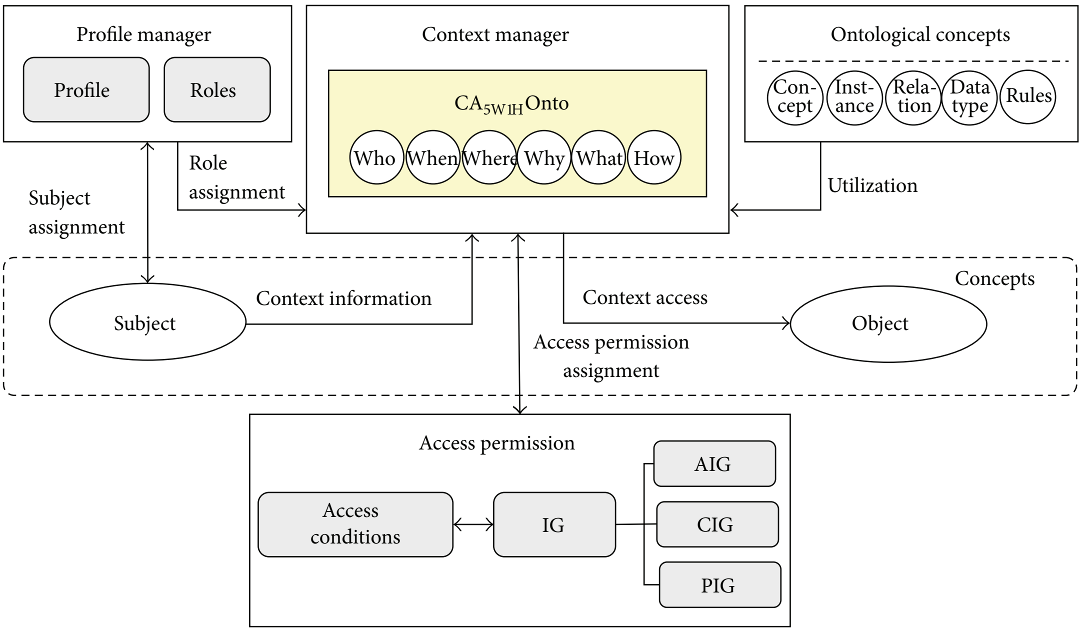


Figure 2: Overview of the CBDAC model.

(v) *Access Permission* signifies the operation of a certain *role* of the *Subject*, whether access to a certain *Object* under certain *Access Condition* and *Intended Goal* is granted or denied. The *Access Condition* and *Intended Goal* are the constituent of the *Access Permission*.

(vi) *Access Condition* consists of the six elements <*Goal*, *Role*, *Action*, *Status*, *Location*, *Time*>. The *Access Condition* is used to check whether the *Context* (5W1H) concepts of the CA_{5W1H}Onto are in accordance with the constituent of the *Access Condition*.

The formalized definition of Role Assignment is as follows.

Definition 1 (role assignment). (i) *Subject Assignment SA* ⊆ *Subject* × *Role* is a many-to-many mapping relationship between *subjects* and their assigned *roles*.

(ii) *Role Assignment RA* ⊆ *Role* × CA_{5W1H}*Onto*.

(iii) *Roles* ⊆ *Role* × *Role* are set of roles that define the function and relationship between roles.

The formalized definition and simple scenario of CA_{5W1H}Onto is as follows.

Definition 2 (CA_{5W1H}Onto). CA_{5W1H}*Onto* = {<*Concept, Instance, Context*> | *Concept* ∈ *Ontological Concepts, Instance* ∈ *Ontological Instances, Context*} *is context-aware modeling that processes context in Context Manager*.

(i) *Concept* = {<*sub, obj*> | *sub* ∈ *Subject, obj* ∈ *Object*} consists of subject and object. The concept is defined by ontological concepts.

(ii) *Instance* = {<*sub_element, obj_element*> | *sub_element* ∈ *Subject Elements, obj_element* ∈ *Object Elements*} consists of sets of subject and object elements.

(iii) *Context* = {*5W1H* <*Why, Who, What, Where, When, How*> | *why* ∈ *Why, who* ∈ *Who, what* ∈ *What, where* ∈ *Where, when* ∈ *When, how* ∈ *How*} is a set of context-aware elements:

(a) *Who: the subject assigned role, namely, an agent that may be a person, organization, or system involved in a context;*

(b) *Why: the reason the context occurred;*

(c) *How: the action leading to the context, namely, a context that may occur when it is acted upon by another entity that is often a human or software agent;*

(d) *What: the access target (Object)*;

(e) *Where: the location of the subject (including spatial information)*;

(f) *When: the time when the context occurred.*

Simple Scenario. User A wants to transfer 100 dollars to friend B's account through the mobile banking system of bank C using a SmartPhone at 00:00 AM. The available time for credit transfer on the mobile banking system is from 00:10 AM to 11:50 PM.

By examining the above scenario, we can find that there are several contexts: (1) Concepts, such as *User* and *Mobile_Banking_System*; (2) Instances, such as *User_A, Transfer_Process, Mobile_Banking_System_of_Bank_C, SmartPhone*, and *00:00 AM*; (3) Context between concepts, such as *Credit_Transfer*; (4) Condition, such as *from 00:10 AM to 11:50 PM*. The following represents the CA_{5W1H}Onto model defined based on the previous scenario:

CA_{5W1H}*Onto* = {*Concept* → *User, Mobile_Banking_System; Instance* → *User_A, Transfer_Process,* *Mobile_Banking_System_of_Bank_C, SmartPhone, 00:00 AM Context → Credit_Transfer*};

Context = {*5W1H* (*Who, Why, How, What, Where, When*)}:

Why = *Credit_Transfer,*
Who = *User_A,*
How = *Transfer_Process,*
What = *Mobile_Banking_System_of_Bank_C,*
Where = *GPS & SmartPhone,*
When = *Start_Time* (*00:00 AM*).

The formalized definition of Dynamic Access Control model for dynamic access permission is as follows.

Definition 3 (dynamic access control). *Dynamic Access Control DyAC* = {*Subject, Object, Access Permission*} means that when the contexts are changed dynamically, the *Subject* is granted (or is denied) access to the *Object* by the *Access Permission*. In other words, the *Access Permission* is applied dynamically depending on the volatile context.

The formalized definition of Access Permission is as follows.

Definition 4 (access permission). (i) *Access Permission AP* = {*Context, Access Condition, Intended Goal*} means an allowable condition whether *Subject* can access *Object* based on *Context*. This is not assigned to static roles but to conditional roles (such as *Access Conditions*). Subjects of context entities dynamically activate access conditions according to their context element (such as 5W1H) during the access process.

(ii) *Access Permission Assignment APA* ⊆ *Context Manager* × *Access Permission* determines access privileges by mapping the elements of *Context Manager* to the elements of *Access Permission*. The *Subject* is allowed to access the *Object* only if the context elements (5W1H) in CA$_{5W1H}$Onto are in accordance with the condition of the *Access Permission*.

(iii) *Access Conditions AC* = {<*goal, role, action, status, location, time*> | *goal* ∈ *Goal, role* ∈ *Role, action* ∈ *Action, status* ∈ *Status, location* ∈ *Location, time* ∈ *Time*} are the set of conditions expressed for object access:

(a) *Goal: the purpose of Object access and of performing context-based system;*

(b) *Role: the acceptable Role set of Subject;*

(c) *Action: the acting range performed by Context;*

(d) *Status: the state of accessible Object;*

(e) *Location: the range of accessible location and spatial information;*

(f) *Time: the range of accessible time.*

(iv) *Intended Goal IG* = {<*aig, cig, pig*> | *aig* ⊆ *Goal, cig* ⊆ *Goal, pig* ⊆ *Goal*} is the set of intended goals. *IG* is the use specified for the determination of the dynamic access permission by the subject role and goal. Consider the following:

(a) *AIG: the Allowable Intended Goal (AIG) is a goal always accessible through determination of who and why in CA$_{5W1H}$Onto, regardless of AC;*

(b) *CIG: the Conditional Intended Goal (CIG) is a conditionally accessible goal only if Context of CA$_{5W1H}$Onto corresponds with AC;*

(c) *PIG: the Prohibited Intended Goal (PIG) is a goal always prohibitive through determination of who and why in CA$_{5W1H}$Onto, regardless of AC.*

(v) *Context Access CA* ⊆ *Context* × *Object* is a many-to-many mapping relationship between *CA$_{5W1H}$Onto* and *Object*.

(vi) *Intended Goal Compliance IGC* ⊆ *APA* ⋈ *IG* is a one-to-one relationship between each *AP* and *Object*, as well as bound *IG*.

A key feature of our proposed model is that it supports not only semantic context, but also dynamic access control. Thus, in this paragraph, we described the stage to provide function definitions to facilitate the discussion of CBDAC model using the context elements and its attributes.

Definition 5 (CBDAC model function). (i) *assigned_role*: *Subject* → *Role*, the mapping of subject *s* onto a set of roles. Formally, *assigned_role*(*s*) = {*r* ∈ *Role* | <*s, r*>∈ *SA*}.

(ii) *access_goal_authorization*: *C* → *AP*, the mapping of context *c* in *CA$_{5W1H}$Onto* onto *AP*. Formally, *access_goal_authorization*(*c*) = {*ap* ∈ *AP* | <*c, ap*>∈ *APA*}.

(iii) *intended_goal_binding*: *Object* → *IG*, the mapping of object *o* onto *ig*, which means finding the bound *ig* of the *Object*.

(iv) *intended_goal_compliance: C* × *AC* × *IG* → {*Grant_Access, Conditionally_Grant_Access, Deny_Access*} is used to determine compliance among the *contexts, access condition,* and the *object's intended goal.* Formally,

(a) *intended_goal_compliance(c, ac, ig)* = *Grant_Access* iff *c* × *ac* ∈ *AIG;*

(b) *intended_goal_compliance(c, ac, ig)* = *Conditionally_Grant_Access* iff *c* × *ac* ∈ *CIG;*

(c) *intended_goal_compliance(c, ac, ig)* = *Deny_Access* iff *c* × *ac* ∈ *PIG.*

3.1. Mapping CA$_{5W1H}$Onto and Access Condition.

In order to support CA$_{5W1H}$Onto-based dynamic access control, we herein define the mapping relationship between CA$_{5W1H}$Onto construct types and *AC* description types. Figure 3 shows this mapping relationship.

The *Why* element from CA$_{5W1H}$Onto compares the purpose that the context is attempting to accomplish by mapping it to the *Goal* of *AC* (such as *Why*::*Goal*). *Who* compares the *Subject* that produced the context with the role authorized to access the context by mapping it to the *Role* (such as *Who*::*Role*). *How* is the process performed by the access objective to process the context, and it compares each process required for the context processing by mapping it to the *Action* of the *AC* (such as *How*::*Action*). *What* becomes

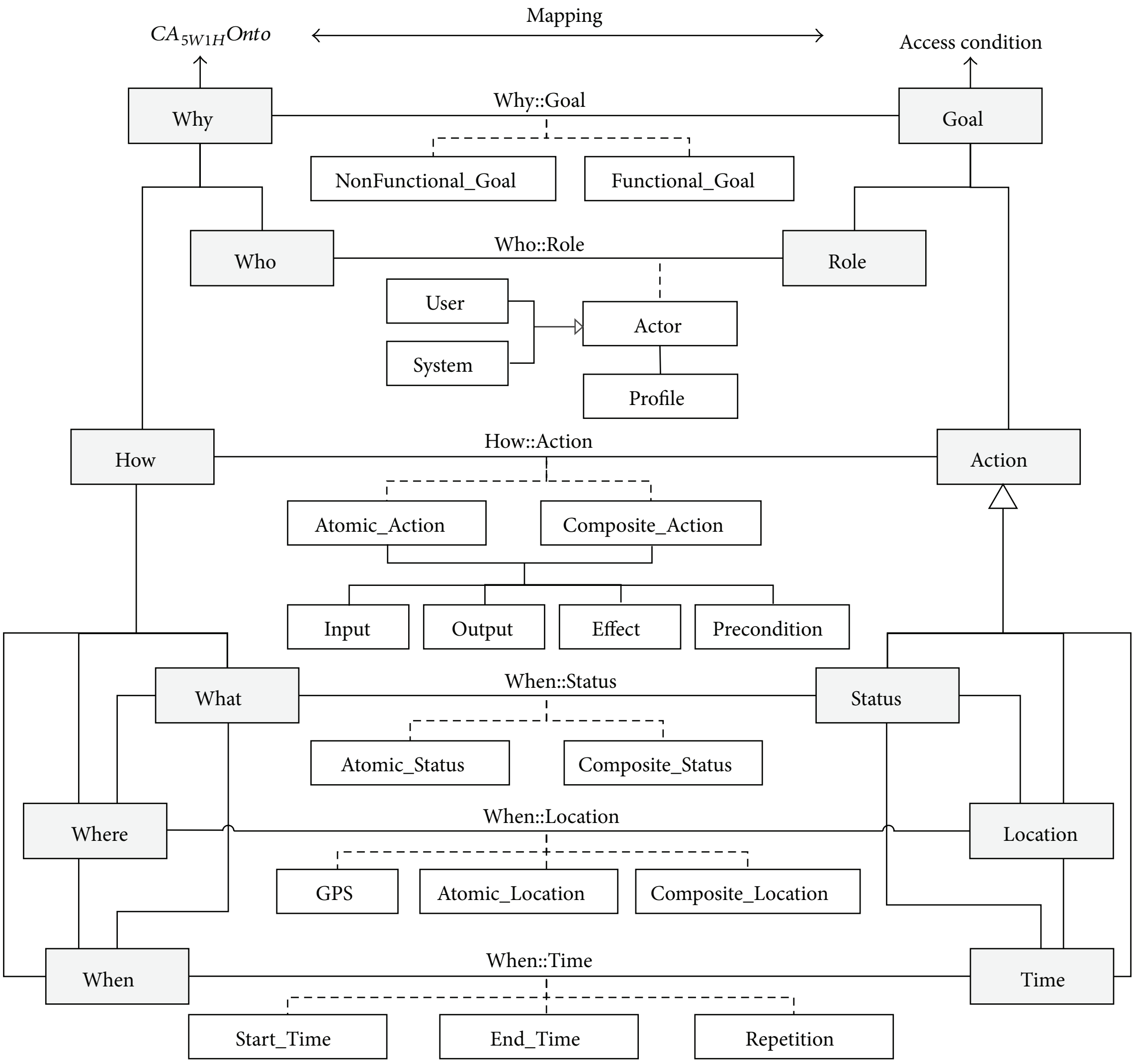


Figure 3: Mapping relationship between $\mathrm{CA}_{5\mathrm{W1H}}$Onto and *AC*.

the *Object*, that is, access objective, and the *Status* being mapped indicates the state of *Object* (such as *What*::*Status*). *Where* indicates the location or spatial information of the context, whereas *Location* compares to determine whether access permission is within the permissible location (such as *Where*::*Location*). *When* implies the time when the context was produced, and *Time* compares to determine whether they are within the allowed time range (such as *When*::*Time*). *Location*, *Status*, and *Time* classes are subclasses of *Action*. For the *Subject* accesses the *Object*, *Location*, *Status*, and *Time* of access condition depend on acting range because the acting range is performed by context information. Table 1 defines and explains the mapping relationship between $\mathrm{CA}_{5\mathrm{W1H}}$Onto and *AC* in detail.

3.2. Dynamic Access Decision. The important challenge for implementing role-based and PBAC models (not context-based access control model) is that it may be difficult to infer the access purpose both accurately and efficiently. With our proposed CBDAC model, the access control strategy can be determined dynamically based on the context elements (such as 5W1H) and subject attributes, in addition to the objects and operations, using access control entities (such as goal, role, action, status, location, and time), and thus it is relatively easy to infer the dynamic access control of context expressivity both accurately and efficiently. Accordingly, we determine dynamic access control by means of Definitions 6, 7, and 8.

Definition 6 (context attribute). Context Attributes are defined as the set of 5W1H properties linked to the granting (or denial) of *AC*. Let *ContextAttribute* denote the set of context attributes that consist of the values {*5W1H* (*Why*, *Who*, *What*, *How*, *Where*, *When*)}. Every *Context* $c \in Context$ is associated with a set of context attributes denoted by $c_n = \{\{why_1, who_1, what_1, how_1, where_1, when_1\}, \{why_2, \ldots, when_n\}\}$. Let *ContextAttribute* denote the set of all possible values of context information.

Definition 7 (access condition attribute). Access Condition Attributes are defined as the set of properties linked to

Table 1: Mapping definition.

Mapping between $CA_{5W1H}Onto$ and *AC*	Definition
Why::*Goal*	Compares the reason the context occurred (*Why*) with the purpose for *Object* access, and for performing context-based system (*Goal*). Its ingredients are *Functional_Goal,* and *NonFunctional_Goal*.
Who::*Role*	Compares the *Subject* that produced the context (*Who*) with the acceptable role set of *Subject* (*Role*). Its ingredients are *Actor* (*User and System*), and *Profile*.
How::*Action*	Compares the action leading to the contexts (*How*) with the acting range that is performed by context (*Action*). Its ingredients are *Atomic_Action, Composite_Action, Expectation, Precondition, Effect, Input, Output*, and the like.
What::*Status*	Compares the access target context (*What*) with the State of accessible *Object* (*Status*). Its ingredients are *Atomic_Status* and *Composite_Status*.
Where::*Location*	Compares the location of Subject (*Where*) with the range of accessible location (*Location*). Its ingredients are *GPS, Atomic_Location,* and *Composite_Location*.
When::*Time*	Compares the time the occurred (*When*) with the range of accessible time (*Time*). Its ingredients are *Start_Time, End_Time,* and *Repetition_Time*.

granting in the context of the access control system. Let *AccessConditionAttributes* denote the set of *AC* attributes that consist of the values {*goal, role, status, action, location, time*}. Every $ac \in AC$ is associated with a set of access condition attributes denoted by $ac_n = \{\{goal_1, role_1, status_1, action_1, location_1, time_1\}, \{goal_2, \ldots, time_n\}\}$. Each attribute ac_n is associated with a finite domain of possible values, denoted as D_n.

Definition 8 (access condition operation). As previously mentioned, the sets of *Context* and *AC* represent the set of defined context elements (5W1H) and *AC* elements (*Goal, Role, Action, Status, Location, Time*). Let the sets *goal*, *role*, *action*, *status*, *location*, and *time* represent the sets of predefined *GoalAttribute*, *RoleAttribute*, *ActionAttribute*, *StatusAttribute*, *LocationAttribute*, and *TimeAttribute*. In addition, $X = GoalAttribute \cup RoleAttribute \cup ActionAttribute \cup StatusAttribute \cup LocationAttribute \cup TimeAttribute$. Each variable $x \in X$ has a finite domain of possible values, denoted as $Domain(X)$. Each Access Condition in *ac* is of the form $<x, op, value>$, where $x \in X$, $value \in Domain(X)$, and $op \in \{=, \neq, <, >, \leq, \geq\}$.

We propose Algorithm 1 for dynamic access decision of the CBDAC model. The algorithm processes mapping between $CA_{5W1H}Onto$ and *AC*. The input comprises *Context c*, *Access Condition ac*, and *Intended Goal ig*. The output comprises an access decision such as *grant_access* or *deny_access*. During the mapping between *Context* attributes and *AC* attributes, as described in Definitions 6 and 7, we perform the *CHECK_INTENDED_GOAL_COMPLIANCE* function that checks the intended goal compliance between *Context* attributes and *AC* attributes, as described in Definition 5. The *CHECK_CONDITION* is a function to check whether the *Context* attribute corresponds to the *AC* attribute. The respective attributes of *Context* and *AC* are as follows:

$$CHECK\_CONDITION\ (c_i,\ ac_j) = <why_i::goal_j,\ who_i::role_j,\ how_i::action_j,\ what_i::status_j,\ where_i::location_j,\ when_i::time_j>.$$

4. Use Case Scenario and Implementation

In this section, we describe a use case scenario of the CBDAC model to ensure the ability to provide responses for dynamic access control applying the mobile banking system. In addition, we describe an implementation that the CBDAC model is defined using ontology.

The *DyAC* is the triple set <*Subject*, *Object*, *AP*>. Using a use case scenario, we find that *Subject* becomes *User*, which generates *Context* in *mobile_Banking_System*, whereas *Object* is assigned to the *Mobile_Banking_System* as an access objective for the *Context*. The access permission of *AP* is determined by the *Context* element of the CA_{5W1H}Onto, as well as *AC* and *IG*. *AP* is an important entity that allows dynamic access control and notices the change in the context (*AC*).

The *Goal* of *AC* is the goal that the context is attempting to perform, which is *Credit_Transfer*, and the accessible *Role* is allocated only to *user* and *System_Admin*. *Action* is the *Transfer_Process in Mobile_Banking_System_of_Bank_C*, and *Status* is only accessible when the system is in the *Activation* state. *Location* indicates the scope of the accessible spatial information, which is defined by *GPS*, *SmartPhone*, or *IP_Address*. *Time* indicates the system-accessible time slot from *00:10 AM* to *11:50 PM*. The access to an object is only permitted if six elements of the contexts defined by *AC* have been satisfied. *IG* signifies the predetermined intended goal. It determines an access privilege based on the *Subject* and *Context*.

The following shows an example of *DyAC* based on scenario described above:

DyAC = {*Subject* → *User; Object* → *Mobile_Banking_System; AP*};

AP = {*Context* → *Credit_Transfer; AC; IG* → *cig*};

Context = {*5W1H* (*Why, Who, How, What, Where, When*)};

AC = {*Goal, Role, Action, Status, Location, Time*}*:*

Goal → *Functional_Goal* (*Credit_Transfer*),

```
INPUT:
  (i) Access Requirement
  (ii) c, a set of context attribute
  (iii) ac, a set of access condition attribute
  (iv) ig, intended goal
OUTPUT: Access Decision (Grant_Access or Deny_Access)
METHOD:
c_i ← the attribute of contexts in $CA_{5W1H}Onto$
ac_j ← the attribute of predefined Access Condition
i, j is the identification number of attributes
    while MAPPING($c_i$, $ac_j$) do
        CHECK_INTENDED_GOAL_COMPLIANCE($c_i$, $ac_j$, ig)
        if ($c_i \times ac_j \in$ AIG) then
            return Grant_Access
        else if ($c_i \times ac_j \in$ CIG) then
            CHECK_CONDITION($c_i$, $ac_j$)
            if $\forall c_i \subseteq ac_j$ then
                return Grant_Access
            else
                return Deny_Access
        else if ($c_i \times ac_j \in$ PIG) then
            return Deny_Access
```

Algorithm 1: Dynamic access decision by mapping between $CA_{5W1H}Onto$ and AC.

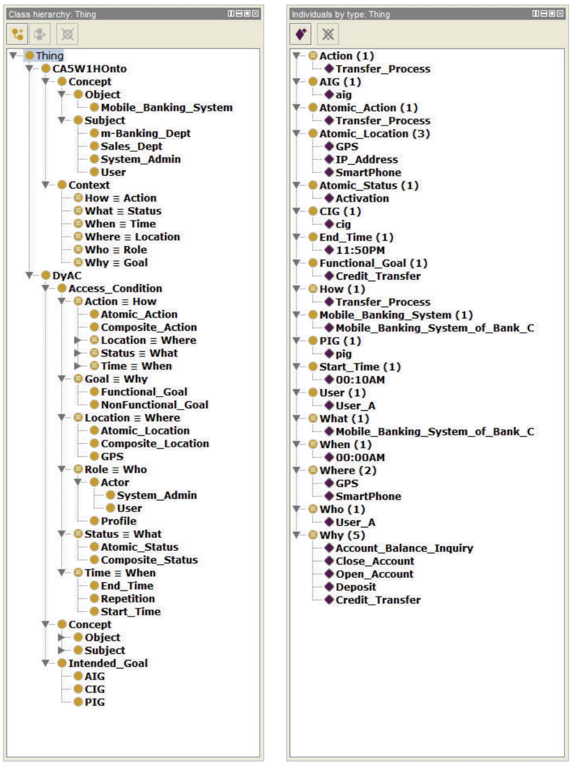
Table 2: Predetermined intended goal.

Context	Subject				
	User	*m-Banking_Dept.*	*Sales_Dept.*	⋯	*System_Admin*
Deposit	cig	pig	pig	⋯	cig
Credit_Transfer	cig	pig	pig	⋯	cig
Open_Account	cig	cig	cig	⋯	cig
Close_Account	pig	cig	pig	⋯	cig
Account_Balance_Inquiry	aig	cig	pig	⋯	pig
⋯	⋯	⋯	⋯	⋯	⋯

Role → *Actor* (*User/System_Admin*),
Action → *Atomic_Action* (*Transfer_Process*),
Status → *Atomic_Status* (*Activation*),
Location → *Atomic_Location* (*GPS/SmartPhone/IP_Address*),
Time → *Start_Time* (*00:10 AM*), *End_Time* (*11:50 PM*).

Table 2 describes the example of *IG* by *DyAC*. In the mobile banking system, *Subject* consists of departments (such as *User*, *m-Banking_Department*, *Sales_Department*, *System_Admin*, etc.); *Context* consists of Deposit, Credit_transfer, Open_account, Close_account, Account_Balance_Inquiry, and so forth. When the *Subject* is User, the User has access to Deposit, Credit_Transfer, Context, and Open_Account only if the context of user corresponds with a particular condition (i.e., *AC is goal, role, action, status, location, and time*). The access of *User* to Close_Account is denied because of *pig*, which is the *intended goal* of the user. On the other hand, the access of *User* to Account_Balance_Inquiry is always granted because of *aig*, which is the *intended goal* of the user. When the *Subject* is *m-Banking_Dept.*, the access of *m-Banking_Dept.* to Deposit and Credit_Transfer is denied despite the correspondence with *AC* because of *pig*, which is intended goal of *m-Banking_Dept.* However, the access of *m-Banking_Dept.* to Open_Account, Close_Account, and Account_Balance_Inquiry is granted by *cig* only if the *Context* of *e-Banking_Dept.* corresponds with *AC*. When the *Subject* is *Sales_Dept.*, the *Sales_Dept.* is allowed access to Open_Account only if the *Context* of *Sales_Dept.* corresponds with *AC*, but the *Sales_Dept.* is denied access to the other *Context*. When the *Subject* is *System_Admin*, the *System_Admin* is denied access to Account_Balance_Inquiry, but the access of *System_Admin* to the other *Context* is allowed only if the *Context* of *Sales_Dept.* corresponds with *AC*. Therefore, the *Subject* is *User* in *DyAC*, and the access of *User* for Credit_Transfer is allowed only if the *Context* of *User* corresponds with all *AC*.

We implement the CBDAC model using Protégé [36, 37]. Figure 4 illustrates the hierarchical classes and instances in the CBDAC model. The class hierarchy shows overall classes and equivalent relation between the $CA_{5W1H}Onto$ class and



```
Class hierarchy: Thing
▼ Thing
  ▼ CA5W1HOnto
    ▼ Concept
      ▼ Object
          Mobile_Banking_System
      ▼ Subject
          m-Banking_Dept
          Sales_Dept
          System_Admin
          User
    ▼ Context
        How ≡ Action
        What ≡ Status
        When ≡ Time
        Where ≡ Location
        Who ≡ Role
        Why ≡ Goal
  ▼ DyAC
    ▼ Access_Condition
      ▼ Action ≡ How
          Atomic_Action
          Composite_Action
        ▶ Location ≡ Where
        ▶ Status ≡ What
        ▶ Time ≡ When
      ▼ Goal ≡ Why
          Functional_Goal
          NonFunctional_Goal
      ▼ Location ≡ Where
          Atomic_Location
          Composite_Location
          GPS
      ▼ Role ≡ Who
        ▼ Actor
            System_Admin
            User
          Profile
      ▼ Status ≡ What
          Atomic_Status
          Composite_Status
      ▼ Time ≡ When
          End_Time
          Repetition
          Start_Time
    ▼ Concept
      ▶ Object
      ▶ Subject
    ▼ Intended_Goal
        AIG
        CIG
        PIG
```

```
Individuals by type: Thing
▼ Action (1)
    Transfer_Process
▼ AIG (1)
    aig
▼ Atomic_Action (1)
    Transfer_Process
▼ Atomic_Location (3)
    GPS
    IP_Address
    SmartPhone
▼ Atomic_Status (1)
    Activation
▼ CIG (1)
    cig
▼ End_Time (1)
    11:50PM
▼ Functional_Goal (1)
    Credit_Transfer
▼ How (1)
    Transfer_Process
▼ Mobile_Banking_System (1)
    Mobile_Banking_System_of_Bank_C
▼ PIG (1)
    pig
▼ Start_Time (1)
    00:10AM
▼ User (1)
    User_A
▼ What (1)
    Mobile_Banking_System_of_Bank_C
▼ When (1)
    00:00AM
▼ Where (2)
    GPS
    SmartPhone
▼ Who (1)
    User_A
▼ Why (5)
    Account_Balance_Inquiry
    Close_Account
    Open_Account
    Deposit
    Credit_Transfer
```

Figure 4: A screenshot of the hierarchical structure and instances in the CBDAC model.

the *DyAC* class. The instances of classes are assigned by contexts in scenario. The ontology of the CBDAC model divides the classes into three groups: CA_{5W1H}*Onto*, *DyAC*, and *Concept*. The CA_{5W1H}*Onto* class includes subclasses to represent the context elements that are the *Context* class. The *DyAC* class includes subclasses to represent the dynamic access control elements. The subclasses of the *DyAC* classes are *Intended Goal* class and *Access Condition* class. The *Intended Goal* class represents the access privileges assigned to subject and context. The subclasses of the *Intended Goal* class consist of the *aig*, *cig*, and *pig* classes, such as those indicated in Table 2. The *Access Condition* class represents the conditions for the object access. The *Access Condition* class consists of six subclasses, as described in Definition 4. The *Context* class that represents the context elements consists of six subclasses, as described in Definition 3. The subclasses of the *Access Condition* and *Context* classes have an equivalent relationship to each other. The *Concept* class includes the subclasses *Subject* and *Object*. The *Concept* class is subclass of the CA_{5W1H}*Onto* and the *DyAC* classes for dynamic access decision.

Based on the scenario presented above, a comparison is made to determine whether the *Subject* satisfies the access condition of the *Object* through the mapping of $\mathrm{CA}_{5\mathrm{W1H}}$Onto of the CBDAC model and the AC elements ($\mathrm{CA}_{5\mathrm{W1H}}$Onto ↔ Access Condition). *Why*::*Goal* makes the comparison by mapping the Credit_Transfer of *Why* and Credit_Transfer of the *Goal*. Because Credit_Transfer of the *Mobile Banking System* is a functional *Goal*, a comparison is made to determine whether the purpose of the process is the same by a mapping such as *Why*::*Goal* → *Credit_Transfer*::*Functional_Goal* (*Credit_Transfer*).

Who::*Role* compares to determine whether the value of *Who* is within the scope allowed to access by mapping *User A* of *Who* and the *User/System_Admin* of the *Role* as *Who*::*Role* → *User_A*::*Actor* (*User/System_Admin*). *How*::*Action* conducts the comparison by mapping the Transfer_Process in the *Mobile Banking System* of *How* and the Transfer_Process in the *Mobile Banking System* of the *Action*. Because Credit_Transfer of the *Mobile Banking System* is a single process, mapping is performed as *How*::*Action* → *Transfer_Process*::*Atomic_Action* (*Transfer_Process*); it compares to determine whether *Context* is an executable process. *What*::*Status* compares to determine at which status of the *What* value a permission for access is granted by mapping the *Mobile_Banking_System_of_Bank_C* of *What* and *Activation* of *Status* as *What*::*Status* → *Mobile_Banking_System_of_Bank_C*::*Atomic_Status* (*Activation*). That is, access is permitted only when the *Mobile_Banking_System_of_Bank_C* is in the state of *Activation*. *Where*::*Location* compares to determine whether the value of *Where* is within the accessible *Location* range by mapping the *GPS* and *SmartPhone* values of *Where* and the *GPS*, *SmartPhone*, and *IP_Address* values of *Location*. Because Credit_Transfer makes access at a specific location or through a specific device, the access permission scope is Atomic_Location and it is expressed as *Where*::*Location* → *GPS and SmartPhone*::*GPS and Atomic_Location* (*SmartPhone/IP_Address*). Finally, *When*::*Time* compares to determine whether the value of *When* is within the allowed time slot by mapping the *Start_Time* (*00:00 AM*) of *When* and *End_Time* (*11:50 PM*). It is expressed as *When*::*Time* → *Start_Time* (*00:00 AM*)::*Start_Time* (*00:10 AM*), *End_Time* (*11:50 PM*):

CA_{5W1H}*Onto* ↔ *Access Condition:*

Why::*Goal* → *Credit_Transfer*:: *Functional_Goal* (*Credit_Transfer*),

Who::*Role* → *User_A*::*Actor* (*User/System_Admin*),

How::*Action* → *Transfer_Process*:: *Atomic_Action* (*Transfer_Process*),

What::*Status* →
Mobile_Banking_System_of_Bank_C::
Atomic_Status (*Activation*),

Where::*Location* → *GPS & SmartPhone*::*GPS & Atomic_Location* (*SmartPhone/IP_Address*),

When::*Time* → *Start_Time* (*00:00 AM*) :: *Start_Time* (*00:10 AM*), *End_Time* (*11:50 PM*).

As shown in the comparison, *Start_Time* (00:00 AM) of When does not correspond with the range of accessible time from *Start_Time* (00:10 AM) to *End_Time* (11:50 PM) on *When*::*Time*. Consequently, *User_A* is denied access to the function *Credit_Transfer* in *Mobile_Banking_System_of_Bank_C* because the *User_A* does not have permissible authority for the function

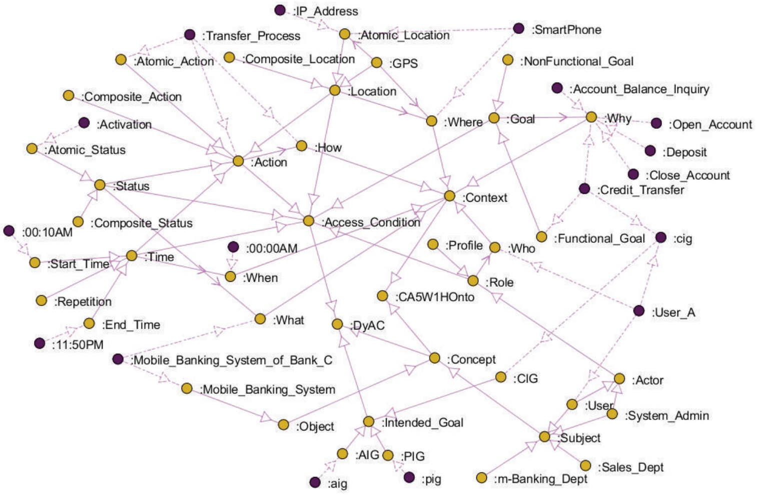


Figure 5: A screenshot of the overall CBDAC model.

Credit_Transfer in *Mobile_Banking_System_of_Bank_C* using *SmartPhone* at *00:00 AM*.

Figure 5 shows the CBDAC model by ontological concepts including classes, instances, and their relationship. Yellow circle represents each class, and the purple circle indicates each instance using the NavigOwl that is a visualization tool Ontology [37]. Further, an arrow indicates each relationship. Our model can be applied and depicted in a real-life scenario using ontological concept. Therefore, we ensure the ability of the CBDAC model to provide correct responses by representing dynamic access decision with a real-life banking system scenario.

5. Conclusion

In this paper, we proposed an access control model for privacy protection based on context in USN named the $\text{CA}_{5\text{W1H}}$Onto-based dynamic access control (CBDAC) model. The CBDAC model makes use of the ontological concept using 5W1H to process context information; it guarantees privacy protection when processing context information in various domains and assigns access permission without any limitation to certain domains in USN. It is possible to process dynamic access control using *DyAC*. In other words, access permission is dynamically assigned according to the change in context information such that, even for a subject with the same role, access permission is defined differently depending on the context information and access condition. Consequently, not only does the CBDAC model authorize access based on roles, which is the key concept of role-based access control, but dynamic access control is also possible because access permission is assigned based on context information, access condition, and intended goal. We also described the algorithm that achieves compliance computation between the access condition and the intended goal. In addition, we showed an applicable use case scenario. To improve our current research, we plan to advance the dynamic access control through the definition of the context-access rule and reasoning method for supporting context inferences, and also a more adequate scenario for experiment and evaluation will be applied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by a Korea University Grant and the basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (2011-0025588), and the Next-Generation Information Computing Development Program through the NRF funded by the Ministry of Science, ICT & Future Planning (2012M3C4A7033346), and the NRF Joint Research Program (2013-054401).

References

[1] H. Cai, F. Pu, R. Huang, and Q. Cao, "A novel ANN-based service selection model for ubiquitous computing environments," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 944–965, 2008.

[2] J. Soldatos, I. Pandis, K. Stamatis, L. Polymenakos, and J. L. Crowley, "Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services," *Computer Communications*, vol. 30, no. 3, pp. 577–591, 2007.

[3] M. Samulowitz, F. Michahelles, and C. Linnhoff-Popien, "Capeus: an architecture for context-aware selection and execution of services," in *New Developments in Distributed Applications and Interoperable Systems*, vol. 70 of *IFIP International Federation for Information Processing*, pp. 23–39, Springer, New York, NY, USA, 2002.

[4] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pp. 85–90, December 1994.

[5] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[6] V. Akman and M. Surav, "The use of situation theory in context modeling," *Computational Intelligence*, vol. 13, no. 3, pp. 427–438, 1997.

[7] CoBrA (Context Broker Architecture), http://cobra.umbc.edu/.

[8] T. Gu, H. K. Pung, and D. Q. Zhang, "A middleware for building context-aware mobile services," in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2656–2660, IEEE, May 2004.

[9] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A middleware infrastructure for active spaces," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 74–83, 2002.

[10] D. Salber, A. K. Dey, and G. D. Abowd, "Context toolkit: aiding the development of context-enabled applications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*, pp. 434–441, May 1999.

[11] J.-D. Kim, J. Son, and D.-K. Baik, "CA_{5W1H} onto: ontological context-aware model based on 5W1H," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 247346, 11 pages, 2012.

[12] ANSI, "American national standard for information technology—role based access control," ANSI INCITS 359-2004, American National Standards Institute, Washington, DC, USA, 2004.

[13] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*, Artech House, Norwood, Mass, USA, 2003.

[14] D. F. Ferraiolo, R. S. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[15] C. E. Shyni and S. Swamynathan, "Reason based access control for privacy protection in object relational database systems," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, pp. 32–37, 2011.

[16] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Hippocratic databases," in *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB '02)*, pp. 143–154, ACM, Hong Kong, August 2002.

[17] P. Ashley, C. S. Powers, and M. Schunter, "Privacy promises, access control, and privacy management. Enforcing privacy throughout an enterprise by extending access control," in *Proceedings of the 3rd International Symposium on Electronic Commerce*, pp. 13–21, IEEE, Research Triangle Park, NC, USA, October 2002.

[18] The Enterprise Privacy Authorization Language (EPAL), IBM, http://www.zurich.ibm.com/security/enterprise-privacy/epal.

[19] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. DeWitt, "Disclosure in Hippocratic databases," in *Proceedings of the 30th International Conference on Very Large Databases (VLDB '04)*, Toronto, Canada, September 2004.

[20] M. Enamul Kabir, H. Wang, and E. Bertino, "A conditional purpose-based access control model with dynamic roles," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1482–1489, 2011.

[21] G. H. M. B. Motta and S. S. Furuie, "A contextual role-based access control authorization model for electronic patient record," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 3, pp. 202–207, 2003.

[22] L. Wang, D. Wijesekera, and S. Jajodia, "A logic-based framework for attribute based access control," in *Proceedings of the ACM Workshop on Formal Methods in Security Engineering (FMSE '04)*, pp. 45–55, October 2004.

[23] M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca, "GEO-RBAC: a spatially aware RBAC," *ACM Transactions on Information and System Security*, vol. 10, no. 1, article 2, 2007.

[24] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 4–23, 2005.

[25] S. S. Emami, M. Amini, and S. Zokaei, "A context-aware access control model for pervasive computing environments," in *Proceedings of the International Conference on Intelligent Pervasive Computing (IPC '07)*, pp. 51–56, IEEE, October 2007.

[26] G. Zhang and M. Parashar, "Context-aware dynamic access control for pervasive computing," in *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS '04)*, San Diego, Calif, USA, January 2004.

[27] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," *ACM Transactions on Information and System Security*, vol. 13, no. 1, article 6, 2009.

[28] R. Zhang, F. Giunchiglia, B. Crispo, and L. Song, "Relation-based access control: an access control model for context-aware computing environment," *Wireless Personal Communications*, vol. 55, no. 1, pp. 5–17, 2010.

[29] D. Beimel and M. Peleg, "Using OWL and SWRL to represent and reason with situation-based access control policies," *Data & Knowledge Engineering*, vol. 70, no. 6, pp. 596–615, 2011.

[30] C. Goh and A. Baldwin, "Towards a more complete model of role," in *Proceedings of the 3rd ACM Workshop on Role-Based Access Control*, pp. 55–61, October 1998.

[31] A. Kumar, N. Karnik, and G. Chafle, "Context sensitivity in role-based access control," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 3, pp. 53–66, 2002.

[32] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Computer role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[33] F. Chen and R. Sandhu, "Constraints for role-based access control," in *Proceedings of the 1st ACM Workshop on Role-Based Access Control*, ACM, Gaithersburg, Md, USA, 1996.

[34] J.-W. Byun and N. Li, "Purpose based access control for privacy protection in relational database systems," *The VLDB Journal*, vol. 17, no. 4, pp. 603–619, 2008.

[35] H. Peng, J. Gu, and X. Ye, "Dynamic purpose-based access control," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '08)*, pp. 695–700, December 2008.

[36] The Protégé-OWL 4.2.0, 2011, http://protege.stanford.edu/.

[37] A. Hussain, K. Latif, A. T. Rextin, A. Hayat, and M. Alam, "Scalable visualization of semantic nets using power-law graphs," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 355–367, 2014.

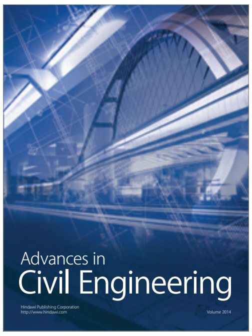
Journal of
Engineering

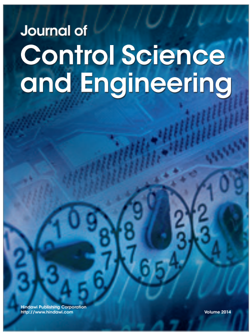
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

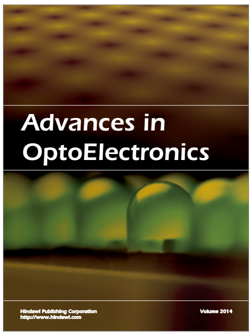
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

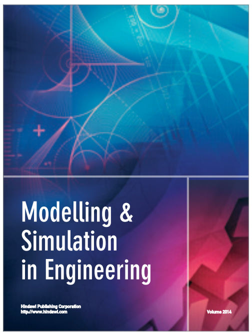
Journal of
Robotics

Journal of
Electrical and Computer Engineering

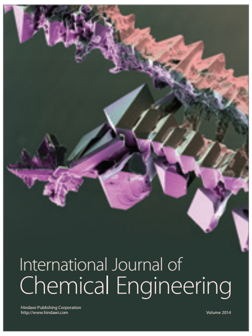

Advances in
OptoElectronics

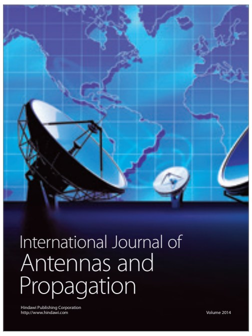
VLSI Design

International Journal of
Navigation and Observation

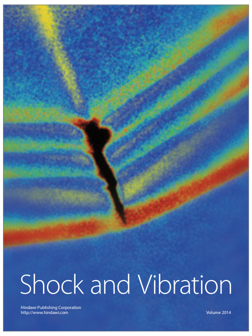

Modelling & Simulation in Engineering

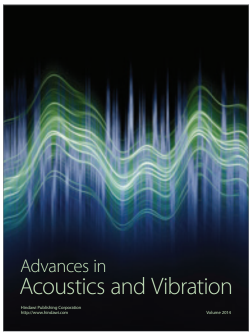
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration