Towards Use-Based Usage Control

Christos Grompanopoulos and Ioannis Mavridis

Department of Applied Informatics, University of Macedonia,
156 Egnatia Street, 54006, Thessaloniki, Greece
{groban,mavridis}@uom.gr

Abstract. In this paper, a new Use-based usage CONtrol (UseCON) approach that supports recording of usages with the help of a new entity, named use, is presented. Uses provide information for the latest state (requested, active, denied, completed or terminated) of every usage and facilitate the fine-grained definition and proper association of attributes to various system entities. The proposed approach provides enhanced contextual information modeling, support of complicated access modes and an alternative approach in obligations modeling. Moreover, UseCON is characterized by high expressiveness and ability to define policy rules in almost natural language.

Keywords: access control, usage control, UCON

1 Introduction

Despite Usage CONtrol's (UCON) [3] virtues, supporting complex usage scenarios of modern computing environments is a challenging procedure [4, 1]. Such a complexity usually results in involving a large number of entities and in utilizing multi party contextual information during the decision making process of a particular usage. Usage control for modern computing environments should support novel access modes on resources, along with new socio-technical abstractions and relations, which are created during the usage process [4]. Enhanced expressiveness is an additional requirement that emerges as of vital importance for next generation usage control models. Furthermore, requirements as the ability to define administrative policies in an ease and efficient way (e.g. expressing policies as closer to the natural language) should be fulfilled.

2 Main Definitions

The UCON family of models [3] is mainly characterized by fine grained control of resources, support for continuity of decisions, and attribute mutability. However, UCON reveals a number of challenging issues when it comes to support modern computing environments. In the rest of this section, the proposed Use based usage CONtrol (UseCON) approach is defined, as an enhancement of UCON.

2.1 Elements

UseCON approach consists of four components viz. subject, object, action and use. Each component is associated with attributes. Moreover, only authorizations are utilized as a usage decision factor.

Components. Subjects are entities that request to exercise operations on objects. A subject can be a human, a device or a software agent acting on behalf of a human. Objects can be physical entities, logical entities or services (e.g. a printer, a file or a database migration service). An entity operating as subject in one usage may be the object of another usage [6].

Actions are operations that subjects exercise on objects. The types of subjects or objects determine the types of the actions that can be exercised. For example, in case of a file, possible actions could be read, write and execute. In a similar manner, different types of subjects (e.g. humans, or agent programs) can exercise different types of actions on the same object (e.g. reload paper into a printer or cancel a job in a printing queue, respectively). Moreover, subjects can exercise through actions direct operations on objects, or delegations of actions to other subjects, or administrative operations on various system elements (in a same way with rights in [3]). In this paper, we further elaborate only on actions that exercise directly operations on objects.

A core component of the UseCON approach is *use*. A use materializes all the characteristics of a usage that are critical for the decision making process. The information contained in a use is not predetermined but is composed at the time a usage is requested. A use actually records the relation between the subject, the object and the action of a particular usage. We define the set *components* (C) containing all the components $(c_i, i = 1, 2, \ldots, n)$, of a system, in the form of subjects (S), objects (O), actions (A) and uses (U).

Attributes. Subjects and objects are associated with security-relevant properties or capabilities, called attributes. In addition, contextual information, which in UCON is stored in condition variables [3], is now proposed to be associated with subject or object attributes. The enhancement of subject or object attributes with context results to the fact that the update of the value of some attributes (those related with contextual information) is performed not through an administrative process, but automatically due to a system's context modification (e.g. time). Additionally, the frequency of attribute mutation has a great range, varying from very rare-changing attributes, e.g. the identity, to always-changing attributes, e.g. time [1]. In order to support complicated operations of modern computing systems, it is required for actions to be associated also with attributes. An example of an action attribute in a file-related operation as write, could be an encryption key. Uses are also associated with attributes that include information related to any combination of a subject, object and action (e.g. the price of a service).

The set of component's attributes (CA) contains the attributes (ca_i, $i = 1, 2, \ldots, m$) of all components. A relation ATT(c_i) denotes the association of a

Towards Use-Based Usage Control 3

component $c_i \in C$ with a tuple of attributes. We adopt the function notation in order to represent the value (range) that is assigned to an attribute (function) of a specific component (domain). For example, in expression $Age(Alice) = 34$, 'Alice' is a component (subject) that has associated to attribute *Age* with assigned value '34'. Every subject, object and action is associated with at least the *id* attribute and through this it is assigned a unique value. The value of an *id* attribute remains constant during the life of the usage control system [5]. When a use instance is created, a tuple of special attributes, namely *sid*, *oid* and *aid*, is initially associated with it. These attributes have the same values to the *id* attributes of the subject, the object and the action, respectively, of the usage that the specific use describes. Moreover, an additional *time* attribute is associated with each one use[1]. The tuple (*sid*, *oid*, *aid*, *time*) is unique for each one use (the usage of a subject on an object with an action at a specific time is also unique) and consequently acts as the identifier of the use.

Each use is further associated with a *state* attribute, which embodies the accomplished status of the usage in progress, as it is described in [5] and augmented in [2]. The *state* attribute represents the possible states of a use, as depicted in Fig.1, and each time it receives one of the following values:

- *Requested*: On a request for a usage, appropriate attributes are associated to a use and proper values are assigned to them. The pre-authorization rules, which govern the requested usage, have not been evaluated yet.
- *Activated*: After a requested usage is allowed, as a result of successfully fulfilled pre-authorization rules, and is now currently being executed.
- *Denied*: After a requested usage has been denied, because it failed to satisfy the pre-authorization rules.
- *Stopped*: After an allowed / ongoing usage has been terminated by the system due to a violation of an ongoing authorization rule.
- *Completed*: A usage that has completed due to a subject's intervention.

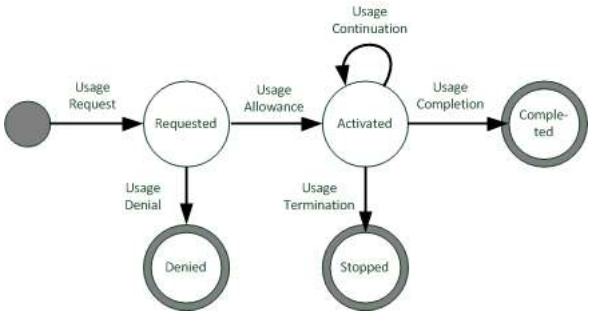

Fig. 1. Use state-transition diagram

[1] The specific value of time attribute could vary from the time of usage request to the time of usage termination/completion and is left open as implementation choice.

Authorizations. Authorizations are functional predicates that utilize component attributes. However, a predicate could be fulfilled by the values of attributes either of the components that are involved in the usage in request, or of the components of other usages in the system. The aforementioned modification makes UseCON authorizations much more flexible than UCON ones, in which the allowance of a usage requested by a subject upon an object is based only on their attributes. For example, an authorization in UCON can model a policy rule requiring that a student can present the work of his team if and only if he has previously been registered in the system. However, UseCON's authorizations can support more complicated rules. Hardening the previous example, it may be required that the presentation of team work by the student is allowed if any member of his team has been registered.

Associating contextual information with component attributes results in the replacement of UCON's conditions with authorizations. Moreover, operations required by UCON's obligations are handled as usual usages in UseCON. Therefore, the exercise of obligation operations in UseCON is verified by searching the uses of the system, supported by authorizations. Based on the aforementioned suggestions, the UseCON elements and their relations are depicted in Fig.2.

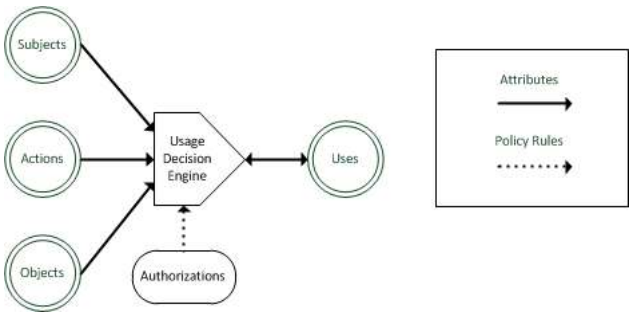

Fig. 2. UseCON elements and relations

2.2 Decision Factors

UseCON utilizes only authorizations as decision factors for the allowance of a usage request. Applying continuity of decision in authorizations results in two UseCON approaches, namely on pre-authorizations and ongoing-authorizations.

Pre-Authorizations. A subject is permitted to exercise an action on an object only if a number of predicates $preA_i, i = 1, 2, \ldots, n$, connected with logical operators are satisfied. There is no further (ongoing) control after the usage's allowance, which is terminated after a subject's request. More specifically, a

UseCON pre-authorization rule is defined as:

$$allowed\,(s, o, a) \Rightarrow preA_1(ATT(u)) \oplus preA_2(ATT(u)) \oplus \ldots \oplus preA_n(ATT(u))$$

where the symbol $\oplus$ is replaced by a logical AND/OR. Each one $preA_i(ATT(u))$, $i = 1, 2, \ldots, l$, is a predicate that utilizes either the attributes from the use that materializes the requested usage or the attributes from other (previous or concurrent) uses in the system. Moreover, each use is associated with a tuple of attributes (*sid*, *oid*, *aid*, *time*) that contains the values of the id attributes of the components participating in the usage. Consequently, with the application of the reverse id^{-1} function (e.g. $id^{-1}(sid(u)) = s$) on the id values of the components, a $preA_i$ predicate is able to utilize the attributes of the components participating in the usage. The semantics of $preA_i$ are defined as follows:

$$preA_i(ATT(u)) \Rightarrow \begin{cases} sp_i(ATT(u)) \\ \mid \{u' \in U \wedge cp_i\,(ATT(u), ATT(u'))\} \mid \otimes k_i \end{cases}$$

where $k_i \in \mathbb{N}$, the symbol $\otimes$ is replaced by $\geq$ or $\leq$, accordingly, and $\mid B \mid$ denotes the number of elements of set B. The term sp_i denotes a simple authorization predicates that is evaluated on the attribute values of the requested use. Alternatively, cp_i is a complex authorization predicate that is evaluated on the attribute values of the requested and the rest uses of the system.

Ongoing-authorizations. An ongoing-authorization utilizes the same elements as a pre-authorization. However, an ongoing-authorization has not to be fulfilled before the usage request, but during the exercise of the usage. A UseCON ongoing-authorization rule is defined as:

$$allowed\,(s, o, a) \Rightarrow true$$
$$stopped\,(s, o, a) \Leftarrow \neg(onA_1(ATT(u)) \oplus onA_2(ATT(u)) \oplus \ldots \oplus onA_n(ATT(u)))$$

The semantics of an onA_i functional predicate are the same with the $preA_i$ ones defined in pre-authorizations and $\oplus$ is replaced by a logical AND/OR.

3 Example

The advantages of the proposed approach are highlighted in this section, via an example, which is a combination of UCON's examples 25 and 26 as presented in [3]. More specifically, a policy rule in a hospital, requires that a doctor can operate a patient only if the patient has given his consent to be operated. In addition, the doctor must have proven experience with at least three operations in the past. The corresponding UseCON modeling is:

AREA : $S, O \rightarrow 2^{Speciality}$	Speciality is a set of medical speciality names
SROLE : $S \rightarrow 2^{Role}$	Role is an unordered set of roles

$$Allowed(s, o, a) \Rightarrow | \{u' \in U \wedge status(u') = \text{ “completed” } \wedge sid(u') = id(s) \wedge$$
$$\text{“doctor”} \in srole(s) \wedge area(s) \cap area(o) \neq \varnothing \wedge aid(u') = \text{ “operate” } \} | \geq 3 \wedge$$
$$| \{u'' \in U \wedge status(u'') = \text{“completed”} \wedge sid(u'') = id(o) \wedge$$
$$aid(u'') = \text{ “consent”} \} | \geq 1$$

The above modeling is composed of two search operations for previous uses of the system. The first search operation returns the number of completed operations performed by the doctor, which must be at least three. The second search operation examines if the patient has exercised the *consent* action. The decision making is based on the conjunction of the logical results of the two search operations.

4 Conclusion

In this paper, the definition of the use entity and a new use-based usage control approach was proposed. A use materializes all the characteristics of a usage that are critical for the decision making process. Consequently, a policy administrator can obtain through uses a detailed view of usages in the system. In addition, UseCON supports an enhanced handling of contextual information together with the support of complicated access modes of subjects on objects and an alternative approach to the utilization of obligations and conditions in UCON. The augmented expressiveness of UseCON was demonstrated in an example composed of two particular ones that have been presented in [3]. Another virtue of the model, as illustrated in the same example, is the similarity between the UseCON modeling and the expression of policy rules in almost natural language.

References

1. Grompanopoulos, C., Mavridis, I.: Towards differentiated utilization of attribute mutability for access control in ubiquitous computing. Informatics, Panhellenic Conference on 0, 118–123 (2010)
2. Katt, B., Zhang, X., Breu, R., Hafner, M., Seifert, J.P.: A general obligation model and continuity: enhanced policy enforcement engine for usage control. In: Proceedings of the 13th ACM symposium on Access control models and technologies. pp. 123–132. SACMAT '08, ACM, New York, NY, USA (2008)
3. Park, J., Sandhu, R.: The ucon abc usage control model. ACM Trans. Inf. Syst. Secur. 7, 128–174 (February 2004)
4. Thomas, R.K., Sandhu, R.: Models, protocols, and architectures for secure pervasive computing: Challenges and research directions. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. pp. 164–. PERCOMW '04, IEEE Computer Society, Washington, DC, USA (2004)
5. Zhang, X., Parisi-Presicce, F., Sandhu, R., Park, J.: Formal model and policy specification of usage control. ACM Trans. Inf. Syst. Secur. 8, 351–387 (November 2005)
6. Zhang, X., Sandhu, R., Parisi-Presicce, F.: Safety analysis of usage control authorization models. In: Proceedings of the 2006 ACM Symposium on Information, computer and communications security. pp. 243–254. ASIACCS '06, ACM, New York, NY, USA (2006)