Information Systems 53 (2015) 253–277

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

An ontological framework for situation-aware access control of software services ☆

A.S.M. Kayes *, Jun Han, Alan Colman

School of Software and Electrical Engineering, Swinburne University of Technology, Victoria 3122, Australia

ARTICLE INFO

Available online 30 April 2015

Keywords:
Situation-awareness
Context information
Purpose
Situation model
Access control policy
Policy model
Situation-aware access control

ABSTRACT

Situation-aware applications need to capture relevant *context information* and *user intention or purpose*, to provide situation-specific access to software services. As such, a situation-aware access control approach coupled with purpose-oriented information is of critical importance. However, modelling *purpose-oriented situations* is a challenging task. Existing modelling approaches for situation-aware systems are not adequate to express *purpose-oriented situations*. Furthermore, existing context/situation-aware access control approaches are highly domain-specific and do not consider *purpose-oriented information*. In this paper we consider *purpose-oriented situations* rather than conventional situations (e.g., user's state) in proposing a generic situation-aware access control framework for software services. We take *situation* to mean the states of the entities and their relationships that are relevant to the purpose of a resource access request. Our framework includes a *situation model* specific to access control, identifying the relevant purpose-oriented situation information. Using the situation model, the *policy model* of the framework provides support for specifying and enforcing situation-aware access control policies. A software prototype has been developed to demonstrate the practical applicability of the framework. In addition, we demonstrate the general applicability of our framework through two case studies from different domains. Experiments are conducted to quantify the performance overhead of providing such situation-aware access control for software services.

1. Introduction

A security policy normally states that a particular service can be invoked based on *the states of the relevant context entities* and *the specific purpose*, which describes the reason for which the organizational resource is used [2]. By identifying the user's intention in accessing software services, we can achieve purpose-oriented control over access to such services. In dynamic and context-aware environments, therefore, situation-aware access control (SAAC) applications need to take into account the relevant *context information* [3] and *user intention or purpose*, to provide situation-specific access to software services. In such environments, users demand access to software services in an anytime and anywhere fashion, as described by Weiser [4], and yet not to compromise the relevant privacy and security requirements of the stakeholders.

For example, a doctor's request to access a patient's medical records through a healthcare application may be

☆ An earlier version of this paper has been published in the Proceedings of the 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014) [1].

* Corresponding author. Tel.: +61452441830.
E-mail addresses: akayes@swin.edu.au (A.S.M. Kayes), jhan@swin.edu.au (J. Han), acolman@swin.edu.au (A. Colman).

http://dx.doi.org/10.1016/j.is.2015.03.011
0306-4379/

appropriate and allowed from the inside of the hospital when the patient is assessed by the doctor, but may not be from a public bus for reviewing patient cases. However, such service access request may also be granted for the emergency treatment purpose even if it is on a bus. In the medical domain the American Health Information Management Association (AHIMA) identifies 18 health care scenarios across 11 purposes (treatment, payment, research, etc.) for health information exchange [5]. Therefore, in order to specify *situations* for SAAC applications, on the one hand, it is required to capture the states of the relevant situation-specific context entities (e.g., user, resource, resource owner) and their relevant relationships (e.g., the interpersonal relationships between the user and the resource owner); on the other hand, it is required to identify the purpose or user's intention in accessing the software services.

Access control is one of the fundamental security mechanisms needed to protect information resources and software services. It determines whether a request to access the resources and services provided by a system should be permitted or denied. A well-accepted traditional access control model based on the roles of the users is role-based access control (RBAC), which has been introduced to tackle the problem of identity-based access control for managing and enforcing security in large-scale domains [6,7]. However, the basic RBAC approach does not provide adequate functionality to incorporate and adapt to dynamically changing contextual information. On the other hand, the basic attribute-based access control (ABAC) approach grants accesses to resources and services based on attributes of entities (e.g., the attributes possessed by the requester) [8]. The ABAC approach has similar drawbacks in supporting the context-awareness. Besides, the attribute-based approaches are not suitable in large-scale domains because they do not scale well in large open systems [9].

Context-aware access control is one of the security mechanisms needed to provide flexible control for users' access to resources and services according to the currently available contextual information [10]. During the past decades, a number of research efforts have extended the basic RBAC approach [7] by incorporating some specific types of contextual information: temporal information (e.g., [11,12]), spatial information (e.g., [13]), and both the time and location (e.g., [14]). Recently, Kulkarni et al. [15], He et al. [16], Huang et al. [17], and Schefer-Wenzl et al. [18] have adopted and extended the basic RBAC approach with some further contextual information from other than the temporal and spatial dimensions, including the resource and environment dimensions as well as the user dimension. Several research efforts (e.g., [19–22]) extend the basic ABAC approach with context-awareness by modelling contextual aspects of the user, resource and environment dimensions as attributes. These extended role-based and attribute-based access control approaches are however still limited in identifying situations.

In the field of situation representation models, there have been several research works for modelling situation information. Endsley [23] defines the basic components to achieve situation-awareness, *"the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"*. Some other research works describe situation as the states of the specific kinds of entities (e.g., [24–26]). Though these situation modelling approaches can specify conventional situations (e.g., the user's state), they are not adequate to specify purpose-oriented situations.

Some situation-aware access control approaches have been proposed in the access control literature (e.g., [27,28]), with each of them having different origins, pursuing different goals and often, by nature, being highly domain-specific. They consider the specific types of context information (e.g., the user's state) as policy constraints to control access to software services or resources. However, other than the relevant entity states, the states of the relevant relationships between entities are not considered. In addition, the purpose or user's intention in accessing the services is not considered in these works. In this paper, we consider the basic elements of situation-aware access control: the relevant *entity states*, the relevant *relationship states* and the *purpose or user's intention*, and their combination.

In general, the existing access control approaches for information resources and software services have only considered specific types of context information as policy constraints. As such, a *new situation model* for access control is required to identify purpose-oriented situation information. Furthermore, a *policy model* for access control that incorporates purpose-oriented situations into the access control process is also required, in order to provide more targeted service access permissions to users.

1.1. Our contributions

The above identified research issues and challenges motivate us to develop a new situation-aware access control framework for software services. In this paper, we present a novel framework, Purpose-Oriented Situation-Aware Access Control (*PO-SAAC*), to provide the capability to make access control decisions for software services by taking into account the purpose-oriented situation information. It makes the following key contributions:

(C1) *Purpose-oriented situation model*: Our framework uses the *purpose-oriented situation* information in determining situation-specific access to software services (authorization), where we present a *situation model* to represent and reason about the different types of situations. The *purpose-oriented situation* can be composed of the *relevant states of the entity and states of the relationships between entities* and the *user's intention or purpose*.

(C2) *Situation-aware access control policy model*: Our framework presents a *SAAC policy model* to specify situation-aware access control policies. The policy model supports access control to the appropriate software services based on the relevant situations.

(C3) *Ontology-based framework implementation*: Based on