A Comparative Analysis of Access Control Policy Modeling Approaches

K. Shantha Kumari, Department of Banking Technology, Pondicherry University, Kalapet, Pondicherry, India
T. Chithraleka, Department of Computer Science, Pondicherry University, Kalapet, Pondicherry, India

ABSTRACT

Access control policies (ACPs) characterize the high-level rules according to which the access control of a system is regulated. Generally they are defined separately from the functional requirements (FRs) of an application and added to the system as an afterthought after being built. But, many problems arose during the integration of ACPs and FRs. Hence, over the past years, researchers have suggested for the modifying the design phase to include an earlier focus on access control issues through various modeling techniques. This paper reviews the important approaches in ACP modeling and makes a comparative analysis of the advantages and limitations of those techniques especially in addressing complex ACPs. Based on the comparative analysis, this paper presents directions for further work needed in handling the intricate nature of today’s ACPs.

Keywords: Access Control Policies (ACPs), Aspect Oriented Modeling, Feature Modeling, Functional Requirements (FRs), Model-Based Approaches, Process-Based Approaches, Unified Modeling Language

1. INTRODUCTION

The objective of access control is to protect resources from unauthorized access, whilst ensuring authorized access. An Access control policy defines the (high-level) rules according to which access control must be regulated. An ACP may express conditions that must be satisfied before an access request can be granted. ACPs are derived from requirements as well as high-level security and privacy policies of the organization. Due to changes of organizational policies, new ACPs emerge, thus leading to new complexities in the application domain that have to be addressed. Due to this there may be a chance for inconsistency between FRs and ACPs. The consistency between the FRs and ACPs have to be considered, as the interference of the access control functionality with application behaviour issues makes it com-

DOI: 10.4018/jsse.2012100104
complicated for understanding and analyzing. This inconsistency issues are not addressed in the traditional approach of software development and this would result in a design with errors as given by Georg, Ray, and France (2002). Late analysis of ACP can generate conflicts between security needs and functional requirements of the system. Georg, G et al states that this leads to security failures, violations from the access control rules, leakage of vital information etc.

This problem was sorted out by externalizing the ACPs as presented by Saltzer and Schroeder (1975), Sirer, Grimm, Gregory, Anderson, and Bershad (1996), and Grimm and Bershad (1996). An additional advantage is that changes to policies can then be performed without the need to modify applications. Since they ACPs are separated from applications, they can be rebuilt, shared, and thus reused. But the independent specification of security policies presents a problem—how to integrate the policies in an application design.

Another solution was proposed by the research community for dealing with the intricate ACPs and also to avoid the inconsistency issues - to address the ACPs from the initial phase of the system development. Addressing the ACPs in the earlier phases is one of today’s challenges in software and requirements engineering research. But Access control mechanisms cannot be blindly inserted into a security-critical system. The Access control requirements can be formulated and integrated into system designs using a high level of abstraction. In this way, it becomes possible to develop security aware applications that are designed with the goal to have consistent access control even during complex situations. To properly address security risks and vulnerabilities without jeopardizing speed or cost, organizations must bring security into the development process and this proved to be effective. Hence, the process of integrating the ACPs with the FRs was recommended in many research works. As a case of representation, the works by Kim, Ray, France, and Li (2004), Priebe, Fernandez, Mehlau, and Pernul (2004), Lodderstedt, Basin, and Doser (2002), Jurjens (2002), and Doan, Michel, and Demurjian (2006) can be taken.

From a software design perspective, access control policies are requirements that must be addressed in a design phase. This also provides an insight to the various kinds of threats, violations that can be well handled in the design phase of the software development. The overall system development process is fruitful when the design phase supports integrated modeling of ACPs and FRs. Hence defining the ACPs in the same way as the FRs in the Design phase is considered as a prudent way. The modeling process should be expressive and flexible enough to accommodate all the different requirements that may need to be expressed, while at the same time be simple both in terms of use and implementation (so that it can be verified with ease).

The integrated development of ACPs with the FRs is also applicable from the requirements phase. This paper focuses on the research work done in combined presentation of ACP with FR at Design phase, as this phase talks more about solution space. Existing research works in this area presents various modeling approaches that defined ACPs as like FRs at design level. These approaches have taken care of the proper abstraction of ACPs and the process of modeling without losing the inconsistency of the FRs.

This paper focuses on the premier research works available in ACP modeling at design level either using Model-based approaches or Process -based approaches. Other methods include – Formal representation of ACPs and High level languages [HLLs] for policy specification. Logic-based languages have proved to be attractive for the specification of security policy, as they have a well understood formalism. The Formal representation of Logic based approaches have the advantage of clear representation and also they are amenable to analysis. However they can be difficult to use and are not always directly translatable into efficient implementation. HLLs are also used for representing the access control functionality at the design level. They are easy to express but not amenable for analysis. This review is based on Ray, Li, France, and Kim (2004). But Model-based and Process-based approaches are both simple in expressing and also effective in analyzing the ACPs when compared to Formal
Event Pattern Discovery in Multi-Cloud Service-Based Applications
Chrysostomos Zeginis, Kyriakos Kritikos and Dimitris Plexousakis (2015).
*International Journal of Systems and Service-Oriented Engineering* (pp. 78-103).
www.igi-global.com/article/event-pattern-discovery-in-multi-cloud-service-
based-applications/137071?camid=4v1a