Future Generation Computer Systems 55 (2016) 336–343

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Role mining using answer set programming

Wei Ye, Ruixuan Li, Xiwu Gu *, Yuhua Li, Kunmei Wen

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, PR China

HIGHLIGHTS

- We propose a novel role mining approach using ASP.
- This novel role mining approach can comply with various kinds of constraints.
- This novel role mining approach meets multi-objective optimization at the same time.

ARTICLE INFO

Article history:
Received 21 November 2013
Received in revised form
23 August 2014
Accepted 9 October 2014
Available online 9 December 2014

Keywords:
RBAC
Constraint
Role mining
Answer set programming

ABSTRACT

With the increasing adoption of role-based access control (RBAC) in business security, role mining technology has been widely applied to aid the process of migrating a non-RBAC system to an RBAC system. However, because it is hard to deal with a variety of constraint conflicts at the same time, none of existing role mining algorithms can simultaneously satisfy various constraints that usually describe organizations' security and business requirements. To extend the ability of role mining technology, this paper proposes a novel role mining approach using answer set programming (ASP) that complies with constraints and meets various optimization objectives, named constrained role miner (CRM). Essentially, the idea is that ASP is an approach to declarative problem solving. Thus, either to discover RBAC configurations or to deal with conflicts between constraints, ASP programs do not need to specify how answers are computed. Finally, we demonstrate the effectiveness and efficiency of our approach through experimental results.

1. Introduction

Currently, role-based access control (RBAC) [1,2] has become the predominant access control model because it greatly simplifies the security management. The key feature of RBAC is that each role is a collection of permissions, and all users acquire permissions only through the roles. However, it is costly to develop and maintain an RBAC system though RBAC reduces the management cost.

In order to build high quality RBAC system, researchers have proposed two important approaches: the top-down approach and the bottom-up approach. The top-down approach [3,4] often starts with expert analysis of business processes and builds RBAC system from such analysis. However, the top-down approach is time consuming since it is human-intensive [5]. The bottom-up approach can discover roles from existing user-permission assignments automatically. Such a computing-intensive approach is called role mining.

Given the same user-permission assignments, different role mining algorithms will build different RBAC systems. Therefore, it requires a measurement to evaluate how good an RBAC state is. The measurement used in Vaidya et al. [6] is the number of roles while the measurement used in Zhang et al. [7] and Ene et al. [8] is the total number of edges when an RBAC state is represented by a graph visually. Guo et al. [9] aim to minimize the number of roles and the edges in role hierarchy graph. Molloy et al. [10] summarized the previous multiple ways of measures and proposed the notion of weighted structural complexity.

Constraint is a defined relationship among roles or a condition related to roles. One of the most common constraints is a separation-of-duty policy. For instance, a user cannot be a member of both mutually exclusion roles. In addition, constraints can be used to reflect business requirements. For example, there is only one person in the role of CEO in a company. As an essential part of the RBAC models, constraints play an important part in defining the security requirements of the system [11–14].

Nonetheless, one main limitation of existing role mining methods is that, the construction process of an RBAC system cannot simultaneously meet various constraints. For example, two constraints are required to be satisfied. The role mining algorithm meets the first constraint but fails to satisfy the second constraint. That means there is a conflict between the two. Then, we should add an algorithm to resolve the conflict in the role mining algorithm. The next step in the example is to add the third constraint.

* Corresponding author.
E-mail addresses: csio@hust.edu.cn (W. Ye), rxli@hust.edu.cn (R. Li), guxiwu@hust.edu.cn (X. Gu), idcliyuhua@hust.edu.cn (Y. Li), kmwen@hust.edu.cn (K. Wen).

http://dx.doi.org/10.1016/j.future.2014.10.018
0167-739X/© 2014 Elsevier B.V. All rights reserved.

The third constraint may have conflicts with the first two constraints. Thus, we may need to implement more different algorithms to resolve these two conflicts. Meanwhile, the first conflict resolution algorithm probably needs to be modified so as to ensure that the first two constraints still do not conflict. Obviously, with the increasing number of constraints, these would become impossible tasks. What is more, you cannot combine various conflict resolution algorithms with the role mining algorithms in many cases.

Leveraging the approach of answer set programming in artificial intelligence, we propose an ASP-based novel approach to construct an RBAC system that can comply with constraints and meet multi-objective optimization at the same time, namely constrained role miner (CRM). In the field of artificial intelligence, ASP has been viewed as an effective programming language for knowledge representation and declarative problem solving [15]. Different from traditional imperative programming languages (C++, Java, etc.), it is about "what to do", without considering many details of "how to do", for solving a problem. ASP allows us to adopt mature ASP solvers that have been proved to work well in practice. Moreover, its rich modeling language eases the understanding and explanation of the problem. With the advantage of ASP, we do not need to implement a variety of specific conflict resolution algorithm, only to describe the constraints problem with ASP modeling language. The case of role mining problem is the same, and the problem can also be solved with ASP approach. Finally, we compute an answer set of the ASP program with ASP solver, and extract the solution if the problem is solvable.

The main contributions of this paper are as follows.

- This paper proposes a novel role mining approach using ASP that can comply with various kinds of constraints and meet multi-objective optimization at the same time, namely constrained role miner (CRM).
- This paper presents experiments to demonstrate the effectiveness of our approach. According to experimental evaluation, CRM is also better than the existing role mining approaches in case of no constraints.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3, we review the notions of role mining problem and the main concepts of ASP. In Section 4, we describe constrained role miner and demonstrate how CRM works by using ASP. In Section 5, we show the results of experiment. Finally, we conclude the paper and discuss future works in Section 6.

2. Related work

There are two basic role engineering approaches: top-down and bottom-up. While the top-down approach defines roles by examining the business processes, the bottom-up approach has been proposed to use data mining techniques to build RBAC system.

Coyne [16] firstly defined the role engineering problem and proposed the concepts of the top-down approach. Kuhlmann et al. [17] proposed the concepts of role mining and how to use data mining techniques for finding roles from user-permission assignments. The *ORCA* algorithm proposed by Schlegelmilch and Steffens [18] is a hierarchical clustering algorithm. However, it does not allow overlapping roles. RoleMiner[19] proposed by Vaidya et al. is a two-phase algorithm based on subset enumeration. *Pair Count (PC)* algorithm proposed by Molloy et al. [5] is based on a new idea for prioritizing roles. *HPr* algorithm [8] was proposed by a group of researchers from HP Labs, it aims to find a minimal set of roles. *HierarchicalMiner(HM)* algorithm [10] was proposed by Molloy et al. This approach is based on formal concept analysis and the semantics of roles.

Clearly, the main drawback of the above role mining algorithms is that they cannot deal with constraints. Therefore, they discover roles when the available information is limited to the user-permission relation. A close related work was proposed by Kumar et al. [20], their algorithm guarantees that no role contains more than a given number of permissions in the discovered configurations. the main drawback of this algorithm is that it only deals with one kind of the cardinality constraints by tradition imperative programming language when there are four kinds of cardinality constraints in all. This algorithm cannot deal with more constraints at the same time.

Lu et al. [21] defined the role mining problem with negative authorizations and proposed an approach to discover underlying constraints from the extended Boolean matrix decomposition. An assumption is proposed that the user-permission assignments imply the information of constraints. However, a lot of the constraints may not be embodied by the user-permission assignments since constraints usually describe high-level security and business requirements. In contrast, our work can deal with all of the constraints whether are embodied by the user-permission assignments or not.

Hu et al. [22] propose an ASP-Based approach to constraint-enhanced role engineering. Role engineering has a two-phase process. The first stage is role mining. The goal of this stage is to construct the RBAC system from a non-RBAC system. The second stage is role updating. It is a post-maintenance for the existing RBAC system. Their work is just for the second stage. Their method tweaks the existing RBAC system in order to satisfy the constraints via answer set programming. The optimization goal of their method is to minimize the change between the initial RBAC system and result RBAC state. Our work realizes the synchronization of constructing RBAC system from scratch and meeting the constraints.

3. Preliminaries

In this section, we will review the main concepts of ASP and the notions of role mining problem and constraint.

3.1. ASP preliminaries

ASP is an approach to declarative problem solving. Rather than solving a problem by telling a computer how to solve the problem, the idea is simply to describe what the problem is and leave its solution to the computer. By comparison with other approaches such as SAT (Satisfiability Checking) and CP (Constraint Programming), ASP is an expressive nonmonotonic language based on stable model semantics, which allows elegant knowledge representation such as causality, defaults, and incomplete information. Then, we review the main concepts of ASP. More details can be seen in Ref. [23].

An answer set program is a finite set of rules of the form

$$h \leftarrow b_1, \ldots, b_m, \mathit{not}\ b_{m+1}, \ldots, \mathit{not}\ b_n \qquad (1)$$

where $0 \leq m \leq n, h$ and b_i are atoms, and *not* denotes default negation. In addition, h is called the head of the rule and $\{b_1, \ldots, b_m, \mathit{not}\ b_{m+1}, \ldots, \mathit{not}\ b_n\}$ is called the body of the rule. When the rule body is empty, the rule is called a fact.

The ground logic program Π is denoted as $P(\Pi)$, which is obtained by all possible substitutions of elements of the Herbrand universe for the variables. Let M be a subset of the Herbrand base of Π. We say that M is called an answer set of a program Π if M is the minimal set of the program $P(\Pi)^M$, which is obtained from $P(\Pi)$ by

- removing all rules having a negative literal *not* b_i in its body where $b_i \in M$ and $i \in [m+1, n]$,
- and then eliminating *not* b_i in the bodies of the remaining rules.

3.2. Role mining problem

In this paper, we will review the basic definitions in RBAC and role mining problems which are the foundation of our work.