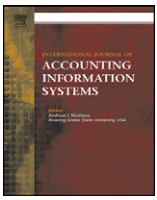
International Journal of Accounting Information Systems 15 (2014) 323–334

Contents lists available at ScienceDirect

International Journal of Accounting Information Systems



# An approach to correctness of security and operational business policies ☆

V.R. Karimi, D.D. Cowan *, P.S.C. Alencar

*David R. Cheriton School of Computer Science, University of Waterloo*

ARTICLE INFO

*Article history:*
Received 1 June 2013
Received in revised form 22 January 2014
Accepted 15 May 2014
Available online 12 July 2014

*Keywords:*
Security policies
REA
Correctness of security policies

ABSTRACT

In this paper we have proposed an approach to describing security and operational business policies and verifying their correctness with respect to a set of properties. The method is based on the REA business modeling language to construct definitions of security and operational business rules. Once the rules are created their representations are combined into policies and policy sets using state machines.



## 1. Introduction

One of the fundamental goals of software engineering is to provide systematic and disciplined approaches for the development of real-world software systems. In contrast with ad hoc approaches, these methods can benefit modern organizations in many ways since they offer techniques that can, for example, be used to guarantee that the software meets specific organizational requirements and works correctly with respect to the expectation of organizational stakeholders.

Providing such guarantees becomes a significant challenge when we consider large and complex modern software such as enterprise resource planning (ERP) systems, which have thousands of control requirements that need to be managed (Gal et al.). These controls involve a number of different aspects, some of which are related to the organization's business processes involving access control and proper business operation. Although frameworks such as COSO and CoBIT have been proposed for the evaluation of an organization's internal controls and emphasize the need to manage these controls properly, there is a need for approaches

☆ The authors thank Graham Gal and Malik Datardina for their valuable comments and suggestions, as discussants of an earlier version of this paper presented at the 2013 University of Waterloo Research Symposium on Information Integrity & Information Systems Assurance.

* Corresponding author.
*E-mail addresses:* vrkarimi@uwaterloo.ca (V.R. Karimi), dcowan@uwaterloo.ca (D.D. Cowan), palencar@uwaterloo.ca (P.S.C. Alencar).

http://dx.doi.org/10.1016/j.accinf.2014.05.004
1467-0895/

that can make the specification of policy-related controls more systematic and support automated control guarantees. In this paper we provide an approach that can be used to guarantee the correctness of business policies involving access control security and organizational process operation. In general terms, this work helps to bridge the gap between the software world supported by formal mathematical models and the corporate accounting world.

Overall, the proposed approach provides a formal method of specifying and evaluating business policies in ERP and similar automated environments. From the standpoint of a real world business perspective, the approach can also benefit practitioners that often need to make sure their access control and operational business policies work as expected. Further, the proposed approach can also help to alleviate problems such as unauthorized access control and dissemination, which are increasingly becoming a threat to modern information systems and can lead to costly and disastrous consequences. In addition, the complexity and large number of policies in real world applications make the use of manual checking not feasible in most cases.

We refer to a policy as a statement that guides decision making and indicates the general direction of an enterprise and is usually in the form of a procedure or protocol. Policies that assist in objective decision-making are usually operational in nature and can be objectively tested. For example a password policy is an objective operational policy.

Corporate operational policies that describe who can perform what actions on what objects are problematic especially when they are implemented in software. Policies incorporated into an enterprise resource planning (ERP) system or across a financial institution can be especially troublesome as a large percentage of policies are buried in software and hidden from human scrutiny. Mergers and acquisitions are also a problem as firms now need to confirm that policies are correct as they are amalgamated.

Testing is one way to have limited confidence in the correctness of a policy's software implementation. However, testing shows that the software passes the test; it does not show that the implementation has overall correct behavior in a given situation. To quote Dijkstra (Dijkstra, 1972) “testing shows the presence of bugs (errors) not their absence.” Even policies that are primarily implemented by people can have incorrect behaviour, although there is the basic safeguard of individuals deciding whether what a policy permits makes sense.

Software engineering practitioners have developed mathematical approaches and tools (sometimes called formal methods) over the last four decades that have allowed them to determine the correctness of portions of a software system. These methods have been applied most often in safety critical systems such as ones controlling aircraft or nuclear power plants. For example, the description of the program or software system is translated into mathematical logic, and then this version of the software is checked to see if it satisfies certain properties. However, these techniques are not readily accessible to the business community as they require advanced mathematical knowledge to apply them and are quite expensive to use in terms of time and expertise.

Recent work by Karimi (Karimi, 2012) on policies related to access control has indicated a direction that looks quite promising in its ability to make a significant subset of these formal methods available to the accounting and business community to check the correctness of operational policies in general. Although there is still some mathematical logic involved its presence in the methods has been substantially reduced through the use of patterns (Karimi, 2012).

This paper outlines the approach by describing the overall model and then applying it to role-based access control (RBAC) (Ferraiolo and Kuhn, 1992). RBAC is a model underlying operational security policies frequently used in business software systems to control access and updates to specific business information. In this case RBAC or similar access control models are used to create the access control rules for a business, and then these rules are combined into business security policies, which are further made into policy sets. For example, a rule is: “a teller may deposit a customer's money into the customer's account” or another rule is: “an account representative may deposit money and may also change personal information.” In contrast a simple policy would be “a teller or account representative may deposit money into a customer's account and an account representative may change personal account information.” A policy set may be: “a manager has all the privileges of an account representative and may also open accounts” along with the previously mentioned policies about tellers and account representatives. Of course policies and policy sets are significantly more complex than these examples.

In this paper we have deliberately kept the examples simple so as not to overcomplicate the methods being demonstrated. For more complex examples the reader can refer to (Karimi, 2012).