Computer Standards & Interfaces 42 (2015) 171–181

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

A twofold model for the analysis of access control policies in industrial networked systems

Ivan Cibrario Bertolotti, Luca Durante, Lucia Seno *, Adriano Valenzano

CNR-IEIIT, c.so Duca degli Abruzzi 24, I-10129 Torino, Italy

ARTICLE INFO

Article history:
Received 16 June 2014
Received in revised form 31 March 2015
Accepted 25 May 2015
Available online 11 June 2015

Keywords:
Industrial networked systems
Access control
RBAC
Security
Modeling of distributed systems

ABSTRACT

Requirements concerning the specification and correct implementation of access control policies have become more and more popular in industrial networked systems during the last years. Unfortunately, the peculiar characteristics of industrial systems often prevent the designer from taking full advantage of technologies and techniques already developed and profitably employed in other application areas. In particular, the unavailability and/or impossibility of adopting hardware (h/w) and software (s/w) mechanisms able to automatically enforce the policies defined at a high level of abstraction, often results in checking the correctness of policy implementation in the real system manually. The first step towards carrying out this cumbersome task in an automated way is the development of a model able to capture both the high level policy specification as well as the details and low-level mechanisms characterizing the actual system implementation. This paper introduces a twofold model for the description of access control policies in industrial environments aimed at coping with this requirement and which can be profitably adopted in several kinds of automated analysis.

1. Introduction

In the last years, the security of industrial networked systems (INS) and, especially, those aspects regarding the protection against threats carried out from either their inside or their outside have been receiving increasing attention [1–7]. Prevention/detection of attacks as well as reaction triggering have been also significantly considered in the scientific literature [8–10]. Unfortunately, relatively little work has been done about techniques for the coherent, error-free design of access control policies and, in particular, for the verification of correctness of policy implementation.

The specification and implementation of access control policies is one of the fundamental steps in the process of securing any kind of system. Indeed, in almost all situations where the access to resources has to be managed in a secure way, a basic need is guaranteeing that the specification of access control policies and their implementation match correctly to some extent.

This aspect is remarkably true for INS, such as those adopted in distributed process control/automation applications and critical infrastructures. Because of technological progresses and enhancements, these systems look more and more similar, today, to general-purpose IT solutions that are popular in other application areas such as, for instance, office and enterprise networks. The other side of the coin is that they are exposed to the same cyber-threats experienced by their IT counterparts.

Different performance and functional requirements, which make INS different from general-purpose IT systems [11], deeply affect the way the matching between the specification and implementation of policies can be guaranteed. In particular, in most real situations the only viable option is an *a posteriori* verification of correctness of policy implementation. Moreover, this difficult and cumbersome task has often to be performed by hand, which makes it even more time-consuming and error-prone.

In these conditions a model, designed to both specify access control policies and capture descriptive elements about the low-level access control mechanisms used for their implementation, is of valuable help in enabling correctness and coherence analyses. Our approach keeps a neat separation between the description of policies following the Role Based Access Control (RBAC) paradigm [12,13], and the description of the system structure including its low-level access control mechanisms and architectural details.

In the following we will refer to the RBAC-based portion of the model as the *Specification model*, while the part dealing with the actual system description will be called *Implementation model*. The *Implementation* view of the model, in particular, takes into account a wide variety of access control mechanisms, which are usually deployed in industrial distributed systems, and is the basis needed to provide a detailed description of the sequences of actions which can be performed by any user to access resources in the considered system. Specifically, the *Implementation model* allows for the description of both the system spatial topology (*i.e.* rooms and gates, cabinets, and so on) and its protection

* Corresponding author.
E-mail addresses: ivan.cibrario@ieiit.cnr.it (I. Cibrario Bertolotti), luca.durante@ieiit.cnr.it (L. Durante), lucia.seno@ieiit.cnr.it (L. Seno), adriano.valenzano@ieiit.cnr.it (A. Valenzano).

http://dx.doi.org/10.1016/j.csi.2015.05.002
0920-5489/

(locks and keys, electronic access control mechanisms, *etc.*). Devices and their physical locations and interconnections are also modeled together with services offered by any of them. Preconditions necessary for accessing services are considered as well (*i.e.* the need for either the physical access to the host where the service is located or the host accessibility through the network) and network device settings (*e.g.* firewall configurations) are also taken into account.

The main idea is schematically represented in Fig. 1 which puts into evidence how information about high-level policies is collected in the *Specification* part of the model, while low-level mechanisms and fine-grained details, concerning the real system implementation, build up the *Implementation* part. The two portions of the model are then processed by an automated analysis s/w tool, which is able compute all the possible actions a user should be allowed to perform according to the policy definition (*Specification set*), on the one hand, and all the operations the user is actually able to carry out in the actual system implementation (*Implementation set*), on the other hand. In particular, to compute the *Implementation set*, the tools builds a Labeled Transition System (LTS) for each user in the system, recording all the possible sequences of actions he/she can perform in the real systems. Elements in both *Specification* and *Implementation* sets are triples in the form (*user*, *operation*, *object*). By (automatically) comparing the resulting sets, the designer can check whether the system is correctly supporting the policies as expected (*verification*) or discrepancies are discovered. Clearly, in practical situation this process is iterative and, in principle, it should be repeated any time anything changes in the system. This occurs, for instance, whenever some policy implementation error is discovered and corrected, so as to keep the *Implementation* model aligned with the physical system characteristics and guarantee the consistency of the *verification* procedure. Note that in Fig. 1 dashed lines indicate tasks that need to be carried out manually by the designer, while solid ones represent actions/computations automatically performed by the tool.

Even though our final goal is checking whether or not the *Implementation* matches the *Specification* correctly, the focus of this paper is on the ability of our model to describe policies and access mechanisms in industrial systems in detail, thus the characteristics of the analysis and automated s/w tool will not be discussed further. The interested reader can find more details on these aspects in [14–17].

It must be pointed out that, while the *Specification set* directly comes from the policy definition, deriving the *Implementation model* from the real system could be a cumbersome task requiring careful consideration. Our future activities will address such a critical point by trying to use already available system information (network equipment configuration files, cabling and network schema, and so on) to automatically feed parts of the model.

The paper is then organized as follows: Section 2 discusses some related works and Section 3 introduces the model structure. Section 4 describes the RBAC-based *Specification model* in more detail, while Section 5 deals with the *Implementation model* and, in particular, presents the description of the physical system and the LTS used to compute the *Implementation set*. Finally, some conclusions are drawn in Section 6.

2. Related works

Access control in general, and RBAC in particular [18–21], have been hot topics for many years: a lot of research has been done in the past,

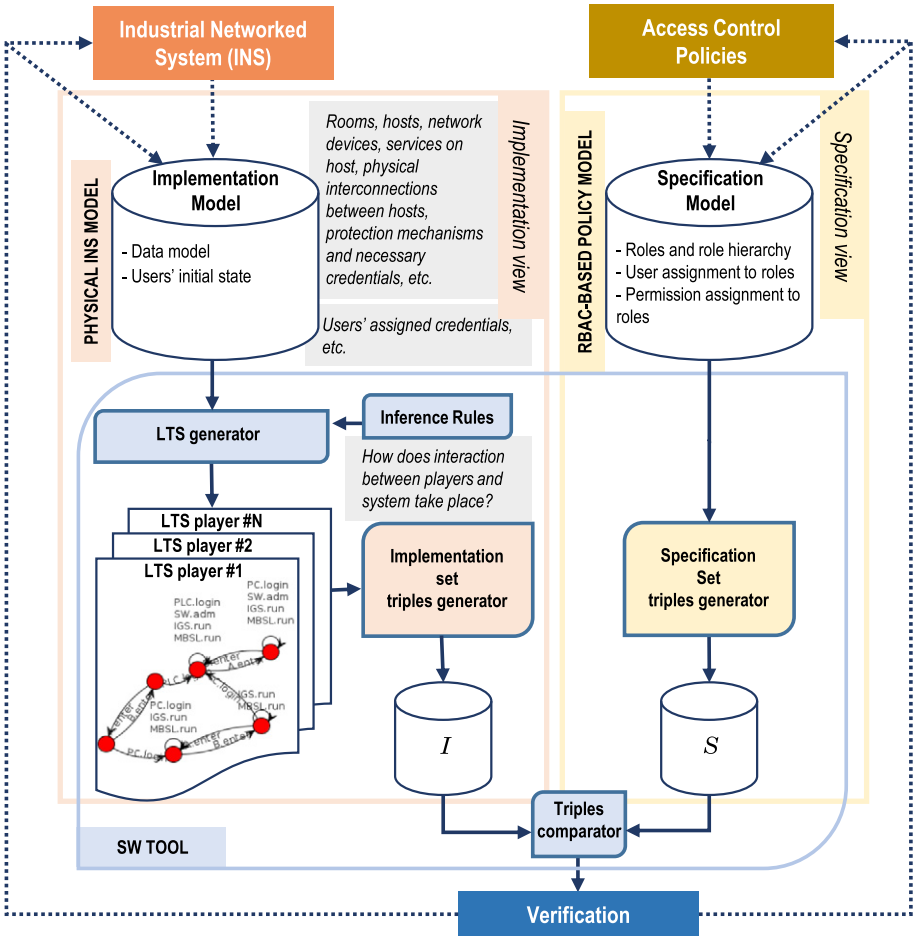

Fig. 1. Logical schema of the verification process.