Policy Engineering for Security Management of Organization Information Systems

Yi Zhang [1] Yong Zhang [2] and Weinong Wang [3]

1, 2, 3: Department of Computer Science and Engineering
Shanghai Jiaotong University, Shanghai, 200030, P. R. China
{zhangyi, zhangyong2001, wnwang}@sjtu.edu.cn

Abstract Security management of information systems is a complex and daunting task in most organizations. Management policies are then introduced to guide and control the management of information systems, but the management of management policies is also a complex task. Therefore, we propose a new concept—policy engineering, which uses the philosophy and paradigms of established engineering disciplines to address these problems. The introduction of policy engineering will provide a systematic methodology that can be used to guide and control the management of information systems.

1 Introduction

It is a common sense that policies play an important role in the management activities within an organization. A policy is a definite goal, course, or method of action specified by the management of an organization to guide and determine present and future decisions. Among these management policies, an organization's security management policies for information systems can be specifically defined as a set of rules derived from the security objectives to administrate, manage, and control accesses to the organization's information assets and related resources.

In order for policies to be active in the management activities, managers and administrators need to define them, and users or management mechanisms need to enforce them. None of these tasks can be complete without great efforts, and they are always interdependent on each other. The deployment of policy in the security management of information systems is therefore a complex and integrated process.

Policy-based management (PBM) provides a promising solution to the problem of managing complex systems. It provides a means by which the management procedure can be simplified and largely automated. The main point of PBM is the notion of using policies to drive the management procedure [1]. In PBM, managers model and define the desirable behavior of a system using high-level abstraction policies. These high-level policies are then automatically or semi-automatically translated into device-specific commands and actions for their enforcement.

While most researchers in this area concentrate on the mechanisms of specifying, distributing, and enforcing security management policies, this paper focuses on the management of policies themselves. We propose a new concept—policy engineering,

which provides a systematic way of managing the management policies themselves and gives a global view of the whole policy evolution process.

2 Policy engineering

Policy-based management is an integral process consisting of several stages, from management requirements analysis and assets evaluation to policy enforcement and maintenance. In this process, policies act as glue that put all the management activities, their participants, management tools, and managed targets together. Nevertheless, the gaps between these stages are always inevitable. To bridge these gaps, a systematic methodology must be in place to ensure the integrity, completeness, consistency, applicability, and maintainability of management policies.

2.1 Policy engineering

Since the complexity of the management policies and management activities, the policy management is not a tiny job but rather an engineering. Therefore, we propose a new concept in the literature—policy engineering—to provide a systematic methodology that can be used to guide and control the whole evolution process of management policies.

Policy engineering uses the philosophy and paradigms of established engineering disciplines to address the problems in the policy driven management activities. It studies the whole evolution process of the management policies. This process goes through a series of phases, covering from the problem domain to the resolution domain, including requirements analysis, policy definition and specification, policy analysis and translation, policy distribution and enforcement, policy monitoring and maintenance, and ultimately, policy retirement. Policy reverse engineering is also an important part of policy engineering. Moreover, it also includes the tools and techniques used to define, enforce, analyze, and maintain management policies, as well as the people involved in this process.

The core of policy engineering is a policy lifecycle model that gives a clear definition of the objective, tasks/activities, input and output, and involved people for each phase of the policy evolution process.

2.2 Relationship with the software engineering

We can see the above definition of policy engineering is very similar to that of software engineering. In order to develop quality software products, software engineers have devised many methods, procedures, and tools, such as analysis and design methodologies, project management, quality assurance, and so on. Software engineering is then applied for the systematic application of these techniques. It defines the activities and artifacts during the software development and how these activities are organized. The software process also goes through a series of phases

such as requirements analysis, specification, planning, design, implementation, integration, maintenance, and retirement [2].

With their similarity, policy engineering and software engineering are mutually supporting rather than contradict to each other. On one hand, policy engineering supports the management of organizational information and related resources, including applications that are the products of software engineering, and even the software process itself. On the other hand, since their similarity, we are encouraged to borrow concept, methods, tools, and even products of the software engineering when we study the policy engineering.

One difference between policy engineering and software engineering is that policy engineering is always built upon an existent system. Its main task is to effectively organize the existent management system or activities, efficiently allocate resources, fully utilize the available management tools and mechanism, find problems of current adopted policies and procedures, define new policies and delete useless ones if necessary, monitor the enforcement of policies, and so on.

Besides software engineering, it will be necessary for policy engineering to draw together important contributions from many other disciplines, including organizational behavior, human resource management, business management, requirement engineering, computer science and engineering, etc., in order to deal in a systematic way with the complexity posed by the management of the organizational information.

3 The phases of the policy evolution process

The policy evolution process is an integral process covering from the problem domain to the resolution domain. Although different organizations may have different policy evolution processes according to their specific requirements and environments, it broadly follows the phases introduced in the following.

(1). Enterprise organizational structure and business process analysis

This work may have been done within the software engineering process and enterprise's organizational structure and business process models may have already existed. The corresponding models in policy lifecycle from which we can derive management objectives are not just a subset of those in software engineering because system management concerns not only what actions or functions the systems should perform or provide, but also what they should not do or provide.

(2). System requirements analysis

In order for policies to be active in the management activities, managers and administrators need to define them. At present, however, the policy definition process is mostly an intuition one based on their experiences. Therefore requirement engineering is needed to provide a systematic way to define management policies and identify constraints

(3). Policy definition and specification

High-level policies are descriptions of what are to be achieved and /or what should be prevented. They are mostly written in natural language and therefore not

enforceable. They should first be specified in a more formal and precise format rather than in natural language before further processing.

(4). Policy analysis and translation

Policy specification must be validated to ensure their satisfaction of certain criteria. Necessary validations include policy syntactic and semantic validation, conflicting detection and resolution, completeness and feasibility validation.

Policy translation is a stepwise refinement process. In order to refine those policies, extensive management information on the management environment is essential. It can be done all by manual or be automated. In most situations, the process is indeed a computer aided-intuition guided one, i.e., done by the experts with the help of computer technologies.

(5). Policy distribution and enforcement

Policy distribution is to transfer the refined policies to the entities that finally enforce them. An important aspect of policy distribution is how to select proper enforcement elements within the system. The policy distribution mechanism must ensure that all policies are transferred to all involved enforcement elements correctly and punctually.

After having received related management policies, the policy enforcement element would begin the action of placing the managed targets in a desired state complying with the management policies, e.g., configuring the access control list, adding an obligation to an administrator's task list, configuring the information services and devices, etc.

(6). Policy monitoring and maintenance

Policy monitoring has mainly two tasks. One task is to check whether management policies are being actually enforced and desired result is achieved. The other task is to monitor the changes of the environment and notify these changes to managers and administrators. The change of the environment requires the adjustment of relevant policies to the new situation. Possible changes include the deployment of new devices or services, the change of the network topology, the introduction of new subject or object, etc.

Policy monitoring system is also responsible for logging those events that are interesting to the managers and administrators. Such information can help them to evaluate the policy definition and deployment, and to make improvement.

Policy maintenance is also an important task in policy-base management. During the policy evolution process, policies are defined, enforced, and at last, abandoned or revised. Policy maintenance mechanism will record the history of all these activities relevant to management policies for two purposes: audit and policy recovery.

(7). Reverse engineering

Information technology nowadays is not just an aid tool for the business of an enterprise but the integral part of the business process. The task of system management should contain the management of both the IT systems and the environment within which IT systems run. As a matter of fact, some of the management goals can be achieved either by the computer systems, or by the proper organization procedure, or by them both. There is a trade-off between these different realizations. A well-designed organizational structure can considerably reduce the complexity of information management. Well-engineered business processes are often the basis of successful systems management. The application of many well-known

management principles, like separation of duties, least privileges, well-formed transactions, etc., largely depends on the well-engineered business processes. As a result, system requirements may lead to the organizational structure redesign and business process reengineering with more and more adoptions of information technologies.

In policy analysis and translation phase, improper defined policies can be found and lead to the re-definition of policies. In policy monitoring and maintenance phase, if the underlying management mechanism cannot meet the requirements of the management objectives, it should be replaced by new management mechanism and policies should be re-translated in accordance with the changes. Also in this phase, the management system should be able to tell which policies are applied to a specific object, or what is the objective that a management procedure or configuration is to support.

Information end-users are the ones most affected by management policies. On one hand, their uses of the enterprise's information and related resources are constrained by security management policies. On the other hand, successful management of information system needs their cooperation and supports.

4. Policy lifecycle model

Policy lifecycle model has been mentioned in several papers [3, 4], but they failed to give a clear and complete definition. According to their abstraction levels, policies are structured from the top-level policies used by a human manager to specify information management objectives through to the low-level policies enforced by management elements. Policy lifecycle model defines not only what activities are involved in the definition and deployment of policies, and how these activities are organized, but also the input and output of each activity, and involved people. Here, we give a complete information management policy lifecycle model in figure 2.

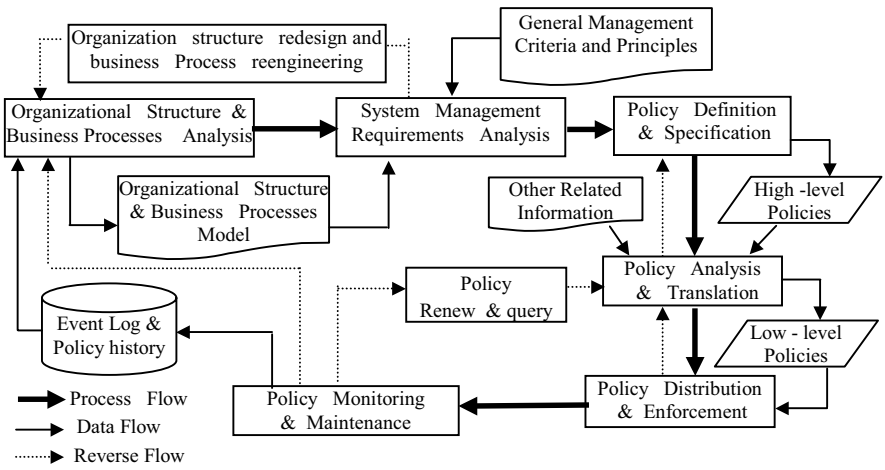

Figure 1. The policy lifecycle model

In figure 1, the process flow represents the main steps and related activities in the policy lifecycle. They have been introduced in detail in the previous section. The data flow illustrates the input and output of each activity. The reverse process illustrates possible reverse engineering in the policy lifecycle.

There are three types of people involved in the policy evolution process (for purpose of simplicity, they are not illustrated in figure1), including policy makers and auditors, administrators and various end-users.

5. Conclusions

In this paper, we propose a new concept, policy engineering, to provide a systematic methodology that can be used to guide and control the various activities and decisions relevant to the management of an organization's information assets and the management of management policies. We discuss in detail each phase in the policy evolution process. We also discuss the relationship between the policy engineering and software engineering, and suggest that many of the techniques and tools can be borrowed from the software engineering to support the definition, analysis and refinement of the information. At last, we give a complete policy lifecycle model that illustrates the activities involved in the policy evolution process and their organizations.

References

[1] S. Sloman, Policy Driven Management for Distributed Systems, Journal of Network and Systems management, Vol.2(4), pp.333-360, 1994.

[2] S. Schach, Software Engineering with JAVA, forth edition, the McGraw-Hill Companies, Inc., 1998.

[3] R. Wies, Using a Classification of Management Policies for Policy Specification and Policy Transformation, Proceedings of the IEIP/IEEE International Symposium on Integrated Network Management, Santa, Barbara, CA, USA, May 1995.

[4] R. Wies, Policy in Network and Systems Management—Formal Definitions and Architecture, Journal of Network and Systems Management, Vol.2, pp. 63-83, Mar. 1994.