Verification of Information Lifecycle Management Policy

TETSUO TANAKA
Research & Development Center, Hitachi Systems & Services, Ltd.
4-11-4 Ohmorikita, Ohta, Tokyo, 143-8545, JAPAN

RYOICHI UEDA
Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji, Asao, Kawasaki, Kanagawa, 244-0817, JAPAN

NORIHISA KOMODA
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, JAPAN

Abstract: Data retention, regulatory compliance, and security requirements are increasing by explosive information growth and government regulations. Information Lifecycle Management (ILM) has gotten a lot of attention as a solution for these requirements recently. In this paper we proposed a policy-based ILM system and a policy verification method. The main feature of the method is 'obligation policy' as approximation of a policy description person's intention, and 'knowledge' about conflicts among policy rules.

Key-Words: Information Lifecycle Management, Policy, Automation, Intension, Knowledge, Correctness, Verification

1 Introduction

The financial and medical sectors are looking for information management technology that will enable them to comply in a cost-effective way with increasingly stringent rules governing the retention of records [1][2]. The U.S. Securities and Exchange Commission, for example, has revised its rule governing the retention of records and now requires exchange members, brokers, and dealers to retain records of electronic communication (e-mails, instant messages, etc.) in an unalterable indexed format that can be readily searched. The Health Insurance Portability and Accountability ACT (HIPPA) of 1996 also requires the retention of medical records such as x-ray photographs.

Furthermore, companies using RFID tags and sensing technology are collecting and sharing the information needed to trace individual items, increasing their need for information technology facilitating the rapid registration, retrieval, and long-term storage of huge amounts of trace information.

Most of the data a company manages is unstructured data, such as that in e-mails and documents [3], and in many cases this data is stored and managed on the employees' desktop and laptop PCs. Rather than leaving such matters as data integrity, disclosure, the prevention of leaks to the discretion of the individual employees, companies want to manage this distributed information in a systematic and comprehensive way [4].

Information lifecycle management (ILM) has therefore been attracting attention [5][6][7][8] because it enables information to be dealt with at a service level appropriate to its value throughout its lifecycle from creation to deletion. Mission-critical data accessed frequently, for example, is saved on reliable high-end storage from which it can be retrieved instantly, whereas important data not accessed frequently is saved on a storage medium with sufficient cost performance at a lower access speed. Data discarded after several months is saved on a low-cost medium.

We have also proposed the method for policy description[9]. The ILM system we propose enables a policy to be described with a business person's vocabulary by offering the knowledge of a domain or a system.

In the method, a policy is described as a list of policy rules, and each rule consists of condition and action. Since each policy rule has a high degree of independence, it is easy to create and to change policy rules, and they are well-suited to expression of fragmentary thinking of human.

However, since the interaction between policy rules is not expressed explicitly, there is a fault, like the conflict between policy rules, difficulty to grasp of what kind of influence by an addition and change of a policy rule. Action that a policy description person does not expect may be performed.

The purpose of our research is verification of a policy realizing an intention of a policy description person. In this paper, "the approximation-expression of a policy description person's intention" and "a verification method of a policy filling an intention" are proposed. Hereafter, Section 2 outlines a target ILM system, and Section 3 describes a definition of a policy and its correctness with examples. The proposed verification method of the policy in the ILM system is described in Section 4.

2 ILM System

2.1 Information Lifecycle Management

Information generally has a lifecycles consisting of stages such as creation, referencing/updating, publication, archive, and deletion. The performance and reliability required and the cost permitted depend on the phase of this lifecycle and on the type of information, as also do the required authenticity, integrity, and confidentiality.

Information managers have the following needs with regard to information lifecycle management.

- Cost reduction: They want to move the information in the archive phase, which is not accessed frequently, into a low-price and low-speed storage.
- Improvement in convenience: They want to move the information in the referencing/updating phase and in the publication phase, both of which are phases in which information is accessed frequently, into a high-speed storage.
- Compliance: They want to prevent the falsification of accounting data, such as a cash book, in the archive phase. They also want to ensure that archived data persists for 5–10 years.
- Prevention of information leakage: They want to encrypt trade secrets and to record access history, and they want to obliterate discarded information (erase files and shred documents).
- Audit: They want to record the identity of a document's creator and record the time and date of creation.

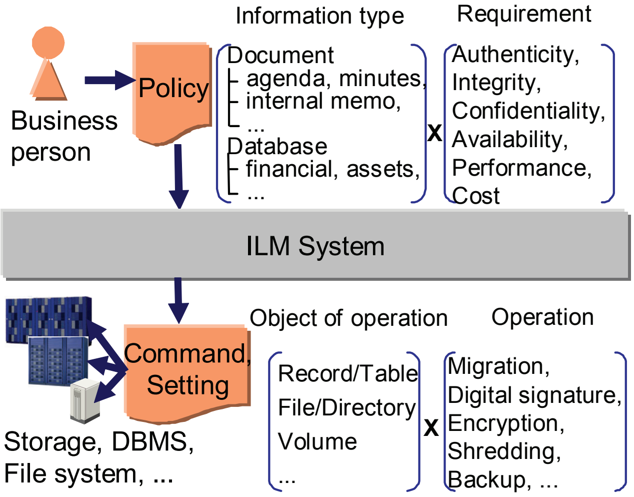

Figure 1. ILM system

ILM meets these needs by moving data between disks, changing access permission, backup data periodically, etc.

2.2 ILM System

To reduce the time and effort required for the operations described in part 2.1 of this section, our information lifecycle management system (ILM system) automates the administration of information.

As shown in Figure 1, an information administrator (i.e., a business person) gives the information administration policy to the ILM system, which will then operate on files and databases accordingly. Policies are the requirements according to the type of information and the phase of the lifecycle.

Policies reflect the business-related logic that is the domain of a business person. They should therefore be specified not by a system administrator but by an information manager (i.e. a business person). While operations on information are performed as commands to a storage system, a file system, DBMS, etc., the policies indicating the information administrator's intension should be described in a way easy for administrators to understand.

The ILM system we propose enables a policy to be described with a business person's vocabulary by offering the knowledge of a domain or a system. When a policy is interpreted, those vocabularies are interpreted in the context of that knowledge, transformed into storage and file-system commands.

3 Policy and its correctness

3.1 Definition of policy

In Section 2 we described policies as "the requirements according to the type of information and the phase of the lifecycle." Since there are, however, millions of ways for satisfying requirements, deriving suitable operation from requirements described declaratively is not efficient. So let us instead say that a policy is a "description of the operations needed for the information to satisfy the above-mentioned requirements when a certain type of information is in a certain phase of its lifecycle."

The Definition 1 is thus the definition of "policy" in this paper.

Definition 1. Policy

```
policy ::= policy_rule*
policy_rule ::= condition action
condition ::= information_type transition_condition
transition_condition ::= period (‘before’|‘after’) event
period ::= digit (years | months | weeks | dates | hours)
event ::= ‘creation’ | ‘reference’ | ‘update’ | ‘deletion’ |
          business_event | date_and_time
action ::= migration | deletion | shredding | backup |
           change_attribute
migration ::= ‘move_to’ destination
change_attribute ::= ‘change’ attribute parameter
attribute ::= ‘access_permission’ | ‘encryption’ |
              ’compression’
```

An "information_type" describes the position of the information in an informational-management system. For example in the office-document domain, it is the type of the document in the document system decided under the document-management rule of the office. In the operational-data domain, it is the type of the ledger in the ledger system.

A "transition_condition" is described by the "event" used to trigger state transition and by the period from an event, or period to an event. And an "event" is a business event, a date and time, or any creation, reference to, updating of, or deletion of information. In the example of an office document, a "business event" can be the activity on the start/end of a project, company events, various deadlines, and an activity in a workflow (approval, etc.).

To enable a business person to describe a policy with his/her vocabulary, the knowledge of a domain is offered as metadata of the information itself and a related thing. For example, since office activity and business are closely related with an organization or a business process, and "the type of information and the phase of the lifecycle" are dependent on the roll of the member of an organization or its organization, and the document system. Therefore, policy description is made easy by offering the metadata about projects such as the date and time of the start/end of a project or the relation between a project and a employee (such as an employee's affiliation)—and the state in business processes (such as approval by a section chief).

3.2 Definition of correctness of policy

A list of policy rules that fulfills the following conditions is defined as a correct policy.

(1) There is no error in each policy rule.
(2) There is no conflict among two or more policy rules.
(3) There is no omission in policy rules (the policy rule that should be described is described).

Hereafter, it explains "error of a policy rule", "conflict between policy rules" and "omission in policy rules" with examples.

3.2.1 Error of policy rule

A error of a policy rule is generated, when the description person made a simple mistake, and when a description person describes policy without following statutes and a corporate rules in ignorance of them (or with malice). By the definition of a policy rule, the error of a policy rule is an error of an information type, an event, a period, or action.

The example of a description mistake is as follows.

- Information type: the error that mistakes the preliminary discussion minutes inside the section for the executive meeting minutes.
- Event: the error that mistakes creation for reference.
- Period: the error that mistakes the period of 'after' for 'before', or 'three months' for 'three years'.
- Action: the error of the destination of migration, or the error of the attribute to set up e.g. ACL.

Moreover, the following is the example of description that follows neither a statute nor a corporate rule.

- The mistake that will shred the document with which an obligation of the storage for 15 years is imposed by law in ten years.

3.2.2 Conflict among policy rules

By the definition of a policy rule, the conflict among policy rules is generated in the following case.

- Inconsistent action that is different from each others performed to the same object simultaneously (or within a fixed period), i.e. information type is the same or in an inclusive relation, the same is said of the transition condition, and action is different from each other and contradictory, e.g. migration to the opposite direction, deletion and archiving. The policy description person who described it as 'archiving' may have an intention to want to keep it. On the other hand, the policy description person who described it as 'deletion' may have an intention that information must not be kept.
- The same action is redundantly performed to the same object sequentially, i.e. information type is the same or in an inclusive relation, the transition conditions differ, and action is the same, e.g. shredding after three years pass and five years pass. A policy description person who described it as "shredding after five years pass" may have the intention like "a document should be kept for an internal audit once in four years".
- Inconsistent action that is different from each other is sequentially performed to the same object, i.e. information type is the same or in an inclusive relation, the transition conditions differ, and actions are contradictory, e.g. shredding after three years pass and migration after five years pass.

3.2.3 Omission in policy rules

The omission in policy rules is generated, when the description person does not describe a policy as he/she intended, or according to statutes and a corporate rules in ignorance of them (or with malice).

The example of the omission in policy rules intended is as follows.

- Although a policy description person intends to save data not accessed frequently on a low-cost medium, there are no policies about migration to the medium.
- Although a policy description person intends to save mission-critical data accessed frequently on reliable high-end storage from which it can be retrieved instantly, there are no policies about migration to the storage.

The example of the omission in policy rules according to statutes and corporate rules is as follows.

- Individual information obliged by corporate rules to be shred is filed for five years, i.e. there are no policy rules about shredding.
- Individual information obligated by corporate rules to be encrypted is saved as plain text, i.e. there are no policy rules about encryption.
- There is not a backup of financial database obligated by corporate rules to be backed up.

4 Policy verification method

Section 3.2 describes three conditions for the correctness of a policy, i.e. no error, no conflict, and no omission. In this section, methods to detect error, conflict, and omission are described.

4.1 Detection of error and omission in policy rules

To exam whether there are errors or omissions in policy description or not i.e. whether policy is described as description person intends, it is necessary to express the intention in the form that can be processed by the computer, and to confront the intention with the policy. However, it is impossible. In this research, an obligation policy is used as approximation of a policy description person's intention.

An obligation policy is a policy which all the sub organizations under organization using an ILM system have to follow, and is described as a list of tuples of condition and status. Obligations explicitly assumed by statutes or corporate rules or implicitly imposed by unwritten rule are described as an obligation policy.

An obligation policy is defined as Definition 2.

Definition 2. Obligation policy

```
obligation_policy ::= obligation_rule*
obligation_ rule ::= condition status
condition ::= information_type obligation_condition
obligation_condition ::=
    obligation_period ('before'|'after') event
obligation_period ::=
    digit ('years' | 'months' | 'weeks' | 'dates' | 'hours')
event ::= 'creation' | 'reference' | 'update' | 'deletion' |
    business_event | date_and_time
obligation_status ::= 'existing' | 'nonexistent' | 'saved on
        high-speed storage' | 'saved on low-cost storage' |
        'encrypted' | 'backed up' | ...
```

The examples of obligation rules are as follows.

- Information must be kept for three months (don't delete).

- The mail from/to a customer must be kept for 15 years after reception/transmission (don't delete).
- Customer's individual information used in a project must not exist after the time of a project end.
- Customer's individual information must be encrypted.
- Ledger data of an accounting section must be backed up.
- The information accessed frequently must be put on high-speed storage (don't migrate to low-speed storage).

The “obligation_status” is the status what the information ought to be, such as ‘existing’, ‘nonexistent’, ‘saved on reliable high-speed storage’, ‘saved on low-cost storage’, ‘encrypted’, ‘backed up’. Since an intention of a policy description person is approximated by the “obligation_policy”, In order to verify that there is neither an error nor an omission in a policy, what is necessary is to check that information that has type described as "information type" in obligation rule is in the status described as "obligation_status" in the period described as "obligation_period". Namely, what is necessary is to check that the policy rule containing action that changes into the obligation status is described (there is no omission), and that the policy rule containing action which stops being in a obligation status is not described (there is no error). For the detection of errors and omissions, the relation between a state and action is offered. Table 1 shows examples of the relation. In Table 1, "mandatory action" is action required in order to change information into "obligation status", and "banned action" is action that changes information into the status where other than "obligation status."

The following procedure performs detection using Table 1.

Table 1. Relationship between status and action

	obligation status	mandatory action	banned action
1	existing	back up	delete, shred
2	nonexistent	shred	-
3	saved on reliable high-speed storage	-	migrate to low-cost storage
4	saved on low-cost storage	migrate to low-cost storage	-
5	encrypted	encript	-
6	backed up	back up	-
	…	…	…

(1) Repeat (2) about each obligation rule of a obligation policy.
(2) Repeat (3) and (4) about each policy rule in the policy with the same (or inclusive) information type as the information type of the obligation rule.
(3) Judge whether the time to perform action of a policy rule (i.e. transition conditions (event + period)) is within the obligation period. If so, judge whether action of the policy rule is banned action corresponding to the obligation status. If so, the obligation rule is violated.
(4) If obligation action corresponding to the obligation status is not action of the policy rule, go to the following policy rule in step (2). If it is action of the obligation rule, judge whether the transition conditions of the policy rule should before or not. If so the policy rule observes the obligation rule.

4.2 Detection of conflict among policy rules

By the definition of conflict among the policy rules in Section 3, to exam whether there are conflicts in policy description or not, it is necessary to exam whether there are following case:

- Inconsistent actions that are different from each other are simultaneously (or to the inside of a fixed period) performed to the same object.
- The same action is redundantly performed to the same object sequentially.
- Inconsistent actions that are different from each other are sequentially performed to the same object.

In this research, conflicts among the policy rules are detected using the following knowledge provided by ILM system.

Knowledge 1: A list of pairs of actions and a period outlined in Table 2
- It is impossible or impermissible that Action 1 and Action 2 are performed within Period.
- Action 1 and Action 2 are in no particular order

Knowledge 2: A list of actions outlined in Table 3
- It is impossible or impermissible that Action is performed to the same object regardless of a period of performing interval.

Knowledge 3: a list of pairs of actions outlined in Table 4
- It is impossible or impermissible that Action 1 and Action 2 are sequentially performed to the same object in this order regardless of a period of performing interval.

Table 2. Example of knowledge 2.

	Action 1	Action 2	Period
1	migration to high-end storage	Migration to low-cost storage	10days
2	delete	*(arbitrary action)	0
3	shred	*(arbitrary action)	0
	…	…	…

Table 3. Example of knowledge 3.

	Action
1	Delete
2	Shred
...	…

Table 4. Example of knowledge 4.

	Action1	Action2
1	delete	*(arbitrary action)
2	shred	*(arbitrary action)
3	migrate to high-speed storage	compress
4	migrate to high-speed storage	archive
	…	…

The following procedure performs detection using Table 2-4.

(1) Repeat (2) about each policy rule of a policy. Label the selected policy rule as RULE1.

(2) Repeat (3) - (5) about each policy rule in the policy with the same (or inclusive) information type as the information type of RULE1. Label the selected policy rule as RULE2.

(3) Judge whether there is the same pair of actions in Table 2 as actions of RULE1 and RULE2 and whether the interval between time when action of RULE1 is performed and time when action of RULE2 is performed is within Period in Table 2. If so, RULE1 conflicts with RULE2.

(4) Judge whether the action of RULE1 is the same as the action of RULE2 and the action is in Table 3, and time when action of RULE1 is performed is different from time when action of RULE2 is performed. If so, RULE1 conflicts with RULE2.

(5) Judge whether there is the same pair of actions in Table 4 as actions of RULE1 and RULE2, and time when the action of RULE1 is performed before the action of RULE2 is performed. If so, RULE1 conflicts RULE2.

5 Conclusion

In this paper we proposed an ILM system and policy description method. We also proposed the definition of the correctness of a policy and a policy verification method. The main feature of the method is an obligation policy as approximation of a policy description person's intention, and knowledge of conflicts among the policy rules.

Experiment of policy description, verification, and investigation for conflict prevention method are issues in the future.

References:

[1] K. Reilly, “AMR Research Predicts Compliance Is an $80B Issue,” *AMR Research,* 14 March 2005 http://www.amrresearch.com/Content/View.asp?pmillid=18086

[2] S. Marlin, “The Cost Of Compliance,” *Security Pipeline,* 6 October 2003 http://www.securitypipeline.com/story/showArticle.jhtml?articleID=15201368

[3] C. T. Chudnow, “File systems and storage,” *Computer Technology Review,* Vol. 22, No. 7, p. 30, 2003

[4] C. Doyle and D. Tapper, “Meeting the Enterprise Data Protection Challenge,” *IDC White Paper,* 2002.

[5] B. Francis, “SNIA nails down ILM definition,” *InfoWorld,* Vol. 26, No. 44, p. 3, 2004.

[6] T. Rief, “Information lifecycle management,” *Computer Technology Review,* Vol. 23, No. 8, pp. 38–39, 2003.

[7] M. Palermo, “Policy-based data management in ILM,” *Computer Technology Review,* Vol. 24, No. 8, pp. 20–21, 2004.

[8] M. Beigi, M. Devarakonda, R. Jain, M. Kaplan, D. Pease, J. Rubas, U. Sharma, and A. Verma, “Policy-Based Information Lifecycle Management in a Large-Scale File System,” *Proc. of the 6th IEEE International Workshop on Policies for Distributed Systems and Networks,* pp. 139–148, 2005.

[9] T. Tanaka, R. Ueda, T. Aizono, K. Ushijima, I. Naitoh, N. Komoda: “Proposal and Evaluation of Policy Description for Information Lifecycle Management”, *Proc. of CIMCA05 & IAWTIC05* Vol. 1, pp.261-268, 2005.