RZ 3667 (# 99667) 05/29/2006
Computer Science 14 pages

Research Report

A Dependability Perspective on Enterprise Compliance

Samuel Müller

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

A Dependability Perspective on Enterprise Compliance

Samuel Müller
IBM Zurich Research Lab
Säumerstrasse 4, 8803 Rüschlikon
+41 44 724 8275 (phone), +41 44 724 8953 (fax)
sml@zurich.ibm.com

Abstract

Large enterprises are confronted with an increasing amount of new and constantly changing regulatory requirements. In the light of this development, successfully becoming and effectively remaining compliant with all relevant regulations poses a big challenge to affected companies. In this paper, we delineate how dependability research can contribute to successful compliance management and help to sustain enterprise compliance. Specifically, we identify concepts and methods from the field of dependability which can be applied to address enterprise compliance. We also demonstrate how the classical dependability and security taxonomy can be generalized and extended to suit the pressing needs of this evolving field.

Keywords: Enterprise Compliance, Compliance Management, Compliance Failure, Auditing, Dependability Taxonomy

1 Introduction

Attaining and sustaining compliance with an increasing amount of new and constantly changing regulations poses a big challenge to affected enterprises. Against this background, dependability research, which investigates the ability of systems to avoid service failures more frequent and severe than acceptable, offers a number of helpful concepts and methods that can be reused in a compliance context. In this paper, we identify concepts and methods used in the field of dependability research, which can contribute to the effective management of enterprise compliance. In addition, we demonstrate how the classical dependability and security taxonomy [15, 19] can be extended to be reused in a compliance context.

1.1 Dependability and Compliance?

While the technical benefit and the respective merits of dependable systems are undoubted in research, investments in dependability are often hard to justify in practice. Unless a system is absolutely critical to a business, or a system failure may have catastrophic consequences, the importance of dependable and secure systems is frequently overlooked or simply ignored.

The advent and importance of enterprise compliance obligations has the potential to change this conception. As we have pointed out in earlier work, organizations are confronted with an growing amount of increasingly complex and constantly evolving regulatory requirements [9]. For instance, as a result of the Sarbanes-Oxely Act [23], chief executive officers and chief financial officers of corporations listed in the USA now face personal liability for the occurrence of material weaknesses in their internal control systems for financial reporting. Furthermore, companies risk paying considerable implicit (e.g., decrease in market valuation or customer base) and explicit (e.g., monetary fine) penalties if they fail to attain and demonstrate compliance with relevant regulations, provisions, and standards. Hence, large enterprises in particular are well-advised to carefully check their regulatory exposure and to ensure overall compliance.

Given the complexity of today's business operations, attaining overall enterprise compliance is not an easy task. To ensure continual compliance with relevant regulations, companies need a well-defined and comprehensive approach to compliance management. In an attempt to address this need, we have proposed a structured compliance management process and a metamodel called REALM (Regulations Expressed As Logical Models), which allows for the formalization of regulatory requirements using a real-time temporal object logic and enables automated transformation of respective regulation models into software artifacts [9].

However, there are also a number of strands of existing research that are in a position to contribute to comprehensive approach and to specific problems. Specifically, dependability and security have a large potential to address many issues central to achieving enterprise compliance. Often, both security and dependability are necessary preconditions to overall enterprise compliance. Moreover, some means classically applied to tackle specific dependability concerns may be reused to address general aspects of enterprise compliance. As a result, regulations and the need to attain compliance drive the adoption of dependability methods.

Not all techniques used in the context of dependability can be adopted to address corresponding compliance concerns. Likewise, not all aspects of enterprise compliance can be solved satisfactorily using the means of dependability. In this paper, we present a number of requirements categories that often occur in current regulations, and confront them with the established dependability and security taxonomy. The resulting matrices improve the understanding of individual compliance categories and identify adequate methods to address them.

1.2 The Role of Information Technology

Due to the growing importance of information technology (IT) to support vital business functions, compliant IT operations are a necessary precondition for overall enterprise compliance. However, modern companies striving to reach compliance with relevant regulations must recognize the ambivalent role of information technology. As they increasingly use IT to support, execute, manage, and monitor vital parts of their business, compliance of business processes which are supported or provided using IT must also be addressed on the IT level. Hence, services provided by IT systems may also need be compliant with specific non-IT regulations. Moreover, certain standards and regulatory requirements directly guide particular IT themes. For example, reaching compliance with standards such as COSO [4], ITIL [21], COBIT [13], or ISO 17799 [11] necessitates setting up and operating according to standardized processes and procedures. Furthermore, also provisions about the security and privacy of data directly impact IT systems, as nowadays most enterprises collect, store, and manage their data electronically.

Independent of whether compliance with a certain regulation has a direct impact on IT or whether IT compliance must be addressed in order to satisfy IT-agnostic regulations, it is important to realize that compliant IT systems are at best a necessary but never a sufficient ingredient for overall enterprise compliance. While compliant IT systems are important, comprehensive enterprise compliance can only be achieved if also employees and their behavior are taken into consideration.

1.3 Concepts from Dependability

As defined by Avižienis et al. [2], dependability refers to "the ability of a system to avoid service failures that are more frequent and more severe than is acceptable". Towards this end, dependability is concerned with reliability, availability, maintainability, and safety (RAMS), and also includes a number of other system properties, such as security and quality of service.

According to Laprie [15] and more recently Avižienis et al. [2], everything that might go wrong in a system can be characterized using one of the following threat types: *fault*, *error*, or *failure*. We speak of a failure when the delivered service no longer complies with its specification. We think of an error as that part of the system state that is liable to lead to a subsequent failure. And we use fault to refer to the hypothesized cause of the error.

Dependability research has also come up with a taxonomy that characterizes the possible approaches to the prevention of failures through preventing and neutralizing errors and faults. In particular, four basic categories have been identified and agreed-upon: *fault prevention*, *fault tolerance*, *fault forecasting*, and *fault removal* [15, 2, 19].

Fault tolerance can be further subdivided into *error processing* and *fault treatment*. Error processing can be done using error detection and recovery, i.e., the substitution of the erroneous state by an error-free one, or error compensation, i.e., delivering an error-free service using available redundancy. Fault treatment is done by first determining the cause of the errors at hand (fault diagnosis), and by then preventing faults from occurring again (fault passivation).

Fault forecasting, possible using both *probabilistic* and *non-probabilistic* techniques, denotes the ex-ante evaluation of system behavior with respect to future fault occurrence or activation.

Finally, fault removal involves three steps: first one needs to check whether faults are present (*verification*), then the causes of identified faults need to be determined (*diagnosis*), and finally, identified faults need to be adequately addressed and possibly removed (*correction*).

1.4 Content and Structure

The remainder of this paper is structured as follows. In the subsequent section, we introduce enterprise compliance and the associated problems. This includes a discussion of the different requirements categories one often finds in current regulations. In Section 3, we combine these categories with the classical dependability and security taxonomy and discuss how the concepts and methods of dependability may help to address enterprise compliance in a systematic way. Besides proposing some extensions to the existing taxonomy, we stress the important role of auditing in the context of enterprise compliance. Along the way, we also provide numerous examples of how specific compliance requirements can be addressed.

2 Enterprise Compliance

Before we can illustrate the application of dependability concepts to enterprise compliance, we have to understand compliance and the related challenges. We start with some definitions. Then, we provide a categorization of regulatory requirements and meta-requirements. And we look at the role that standards and metrics play in the context of enterprise compliance.

2.1 Definitions

Regulatory compliance refers to the ability of an entity (enterprise, organization, or individual) to operate and interact with other entities while adhering to a set of relevant regulatory requirements. Accordingly, an enterprise, organization, or individual is said to be *compliant* if it adheres to and does not violate any relevant provisions. In a narrow sense, a *regulatory requirement* is a piece of legislation, law, a provision, or any other guiding directive enacted by a sovereign regulatory authority imposed onto a well-defined group of individuals, enterprises or organizations. In a broader sense, a *regulatory requirement* may also include standards, business policies, etc., that is, also guiding directives not provided by a legislating authority. Talking about *enterprise compliance*, one generally adopts this broader definition of a regulatory requirement but confines one's perspective to corporate enterprises and their employees. Finally, *compliance management* in an enterprise denotes the proactive process of becoming and remaining compliant. Whether or not an enterprise (i.e., its systems, services, processes, products, etc.) complies with a set of relevant regulatory requirements is normally assessed in an *audit*. Increasingly, in audits of certain regulations such as the mentioned Sarbanes-Oxley Act or the Gramm-Leach-Bliley Act [10], enterprises are required to provide evidence that they tested the effectiveness of the controls that address, implement, and enforce individual requirements. A *compliance violation* or *compliance failure* occurs if a *component* or *service function* of the enterprise fails to comply with a regulatory requirement.[1]

As already defined earlier, dependability denotes "the ability of a system to avoid service failures that are more frequent and more severe than is acceptable". Recognizing that an enterprise is nothing else than a complex system, this definition can also be accommodated to suit the area of compliance management. In particular, we can understand regulatory requirements as part of the enterprise's system specification. In this sense, a violation of enterprise compliance occurs if an arbitrary component or service fails to comply with any regulatory requirement being part of the system specification of the enterprise.[2]

Turning back to the earlier definitions of fault, error, and failure, in the context of enterprise compliance, we can introduce a *functional compliance failure* as a specific subtype of a service failure. In addition to service failures, compliance failures may also occur in the sense of *structural compliance failures*. This type of failure, for which there is no corresponding concept in dependability, occurs if the structure of a system (i.e., the enterprise), or a composite or atomic component of it, fails to comply with a regulatory requirement.[3] Compliance failures can be characterized according to the four viewpoints described by Avižienis et al. [2]: *domain*, *detectability*, *consistency*, and *consequences*. The definitions of 'fault' and 'error' keep their original semantics also in the compliance context. They can be characterized using the eight viewpoints of the fault taxonomy and the elaborations on errors presented by Avižienis et al. [2].

We have introduced and characterized enterprise compliance using existing concepts from dependability. Moreover, we have introduced two new failure types, namely functional and structural compliance failures.

2.2 Requirements Categories

Regulatory requirements with a potential impact on the IT of an enterprise can be subdivided into three main categories: *data-centric*, *process-centric*, and *people-centric* requirements. As shown in the following, each of these categories can be split up into further subcategories.

Data-centric Requirements. Data-centric requirements focus on the proper collection, handling, reproduction, security, and privacy of data. The following subcategories occur frequently:

- *Content:* Explicitly require or forbid the collection of specific data.

[1] We use the terms *structure*, *component*, *behavior*, etc. using the semantics defined in the dependability and security taxonomy established in [2]. Whenever we deviate from the semantics used in this taxonomy, we will clearly state this.

[2] Further parts of the enterprise's "system specification" are other guiding artifacts such as the business strategy, specific business policies, company values, etc.

[3] As an example of a structural compliance failure consider the requirement that the chief executive officer of an enterprise must be supervised by a board of directors. If an enterprise's internal organization does not conform to this requirement, we have an instance of a structural compliance failure. Note that the provided services of the enterprise do not need to be affected by this failure and can per se still remain compliant. The example also involves a functional compliance failure on the part of the original service that constitutively created the enterprise's structure.

- *Retention:* Prescribe how long particular data must at least be retained or may at most be retained.
- *Reproduction:* Require the reliable reproduction of stored data records.
- *Security:* Mandate the adequate protection of critical data against unauthorized access and modification by deploying adequate security measures.
- *Privacy:* Include specific requirements regarding the treatment of personally identifiable data (e.g., use only for a specific purpose, fulfil certain obligations upon collection, etc.).

Process-centric Requirements. Process-centric requirements address the activities, structure, execution behavior, and documentation of business and IT processes. Relevant categories are:

- *Activity:* Require the execution of explicitly stated activities.
- *Ordering:* Require the structuring of certain activities in a predefined order.
- *Timing:* Refer to temporal requirements of an *absolute*, *relative*, or *periodic* type. Absolute temporal requirements necessitate the execution of specific activities at some specific time. Relative temporal requirements use another event or activity as their reference point. Periodic temporal requirements rely on recurring points in time to specify when certain activities need to be executed.
- *Documentation:* Mandate the comprehensive documentation of processes, activities and their results.

People-centric Requirements. People-centric requirements represent another prominent category. They address specific characteristics of an enterprise's employees. Relevant and often found categories are:

- *Accountability and Responsibility:* Denote requirements that associate individuals, professions, or roles with specific responsibility and accountability.
- *Knowledge and Belief:* Comprise specific skills and precise information on particular events, which people in certain positions are obliged to have.
- *Education and Training:* Require that persons in certain positions must have completed a certain education or a specific training.

2.3 Meta-Requirements

Regulatory requirements may themselves be subject to certain meta-requirements that are implicitly implied by many regulations. Such meta-requirements emerge because compliance often needs to be audited by independent auditors. Conformance with these meta-requirements helps auditors better understand how specific requirements have been implemented and whether they are being followed. This normally increases the likelihood of successfully passing an audit. Frequently, requirements are constrained by the following principles:

Documentation. The implementation of requirements should be appropriately documented. This includes the documentation of the original requirement, the enterprise-specific interpretation(s) of the requirement, specific implementation decisions, and the change history.

Traceability. Requirements should be traceable over their entire life cycle. This starts with the original source (i.e., the legal text), linking it to its interpretation and implementation in specific artifacts through transformation, and also includes linking the original requirement to all affected systems, processes, and products.

Accountability. All requirements should have one or several well-defined *owners* who are responsible and accountable for their concrete interpretation, implementation decisions, their deployment, enforcement and their removal. In addition, an *assessor* should be assigned who is responsible to assess the effectiveness of the implemented requirement.

By adopting a dependability perspective on enterprise compliance, we stipulate the adoption of *compliance* as an additional attribute extending the established set of dependability and security attributes [2]. Furthermore, we suggest that the above set of meta-requirements may characterize how an enterprise addresses this supplemental aspect.

2.4 Standards and Metrics

The substantial complexity and impact of many regulations, and the associated non-compliance risk have resulted in a number of *standards* being recommended for adoption by many regulators. Using standards such as COSO, COBIT, ISO 17799, or ITIL has the advantage that auditors can better plan their audits and compare the results in a meaningful way. For affected enterprises, the adoption of established standards often means a reduction of uncertainty with respect to compliance audits and, in addition, helps them to better understand and manage their own processes. Most of the standards actually used in practice focus on establishing a so-called cycle of continuous improvement. After every inspection or audit of the enterprise's processes, systems, or products, there is an improvement phase wherein identified flaws and weaknesses are addressed and eventually removed. Like this, enterprises strive to achieve ever higher maturity levels in close analogy to the well-known CMMI[4] maturity levels [3].

When specifying the details of an IT process that supports or executes a particular business function, the selection of appropriate *metrics* is of major importance. Ex post, only a set of suitable metrics allows for assessing whether the process performs with the required efficiency, effectiveness, security, etc. Hence, according to many standards, the implementation of IT processes necessitates the definition of so-called key goal indicators (KGIs) and/or key performance indicators (KPIs). KGIs define measures that inform management whether an IT process has achieved its business requirements. They are usually expressed in terms of information criteria such as the absence of integrity and confidentiality risks, cost-efficiency, reliability, effectiveness and compliance. KPIs define measures that determine how well the IT process performs with respect to enabling the goal to be reached.[5]

3 Dependability for Compliance

As the importance of IT in supporting and handling business operations continues to grow, tools and techniques for the proper engineering of the respective systems are increasingly important. Paralleling this development, the rising scope and impact of regulations put a heavy burden on enterprises to appropriately address these requirements and to demonstrate compliance. Dependability concepts can be used to ensure the correct and compliant provisioning of IT services. As IT continues to be used to support and enable vital business functions, established means from dependability research may even be reused and applied to other aspects of enterprise compliance and is not constrained to IT.

In this section, we take on a unified perspective on both dependability and compliance by combining the means of the dependability and security taxonomy with the introduced require-

[4] Capability Maturity Model Integrated (CMMI), developed at the Software Engineering Institute of the Carnegie Mellon University, Pittsburgh.

[5] With respect to compliance, the choice of the appropriate metrics strongly depends on whether compliance is assessed with respect to a data-centric, a process-centric or a people-centric requirement.

			Fault Tolerance			
			Error Processing		Fault Treatment	
			Error Detection || Recovery	Error Compensation	Fault Diagnosis	Fault Passivation
Regulatory Requirements Categories	Data-centric	Content	Input Data Analysis || Enforcement (e.g. ex-post Request)	Redundancy (e.g. Four-Eyes Principle, Error-correct. Codes)	Code/Process/ Policy Review	Isolation, Reconfiguration (Process, Policy, Code)
		Retention	Policy/Data/Log Inspection || Ex-post Request; Backup; Deletion	Data Redundancy (Backup, Mirrors)	Code, Process, Policy Analysis	Isolation, Reconfiguration (Process, Policy, Code)
		Reproduction	Inspection, Test || Customized Transfer, Emulation	n/a	System Analysis	Reconfiguration (Migration)
		Security	Log/Event Analysis, Checksums, Intrusion Detection || Selective Blocking	Multi-layered Security, Firewalls, Fault-masking	System Security Analysis	Isolation, Reconfiguration (Process, Policy, Code)
		Privacy	Log/Event Analysis || n/a	n/a	Policy/Process/ Code Analysis	Isolation, Reconfiguration (Process, Policy, Code)
	Process-centric	Activity	Process Analysis || Rollback, Rollforward	Exception Handling	Code/Process Review	Isolation, Reinitialization (Process)
		Ordering	Event Correlation || Rollback, Rollforward	n/a	Process/Code Analysis	Isolation, Reinitialization (Process)
		Timing	Event Correlation || Rollback, Rollforward	n/a	Process/Code Analysis	Isolation, Reinitialization (Process)
		Documentation	Documentation Review || Update Documentation	Redundancy in Documentation	Review assigned Responsibilities for Documentation	Reconfiguration (Update Documentation, Responsibilities)
	People-centric	Accountability	Event/Log Analysis || Responsibility Assignment	Delegation	Review of Responsibilities	Implementation of RACI Matrix
		Knowledge/ Belief	Situation/Skill Analysis || Information Collection	Knowledge Bases, Documentation	Interviews, Skill Analysis	Training, Implementation of Processes, Databases etc.
		Education/ Training	Skill Review || Coaching	n/a	Examination, Check-up, Skill Profile	Training, Coaching, Education

Table 1: Fault tolerance techniques for specific requirements categories.

ments categories. The resulting tables identify possible solutions to addressing the various categories of enterprise compliance. Specifically, the tables pinpoint possible areas where techniques from dependability research can be feasibly applied to compliance management. Moreover, the tables also identify areas, where solutions are either impractical or still missing and first have to be developed.

3.1 Overview

We present three tables, each one applying well-known means of dependability to the regulatory requirements categories introduced earlier. Table 1 identifies techniques from fault tolerance which can be utilized to ensuring compliance with the presented requirements categories. Table 2 summarizes which fault forecasting methods can be applied to the management of general compliance categories. Finally, Table 3 outlines which means of fault prevention and fault removal support compliance with individual requirements categories.

3.2 Fault Tolerance for Compliance

In a compliance context, fault tolerance refers to the ability of delivering a compliant service also in the presence of faults that potentially cause compliance failures. The concepts of dependability contain two strategies to address them: error processing and fault treatment.[6]

Table 1 presents an overview of methods from the area of fault tolerance, which are appropriate for handling data-centric, process-centric, and people-centric regulatory requirements categories.

[6] What we and Meadows and McLean [19] call ‘error processing’ is referred to as ‘error detection’ and ‘error handling’ by Avižienis et al. [2]. Likewise, our term ‘fault treatment’ includes the various concepts of ‘fault handling’ defined by Avižienis et al. In contrast to these authors, we also do not use the notion of ‘recovery’ as a super-concept to ‘error handling’ and ‘fault handling’ but we regard it as a specific means of ‘error processing’.

3.2.1 Error Processing

With respect to achieving and sustaining a status of continuous compliance, error processing refers to the monitoring of compliance events, and aims at detecting and properly handling errors that might otherwise lead to compliance failures.

Error Detection and Recovery In the presence of data-centric requirements, errors potentially leading to compliance failures can be detected through thorough analyses of input data, system logs, and real-time events as well as by intrusion detection techniques [5] and inspections of individual policies and code [17].

Detecting errors associated with process-centric requirements may necessitate the instrumentation of systems to emit compliance events reporting on relevant business situations. Emitted events can then be monitored and correlated by correlation systems. Errors related to process-centric requirements of the activity type can be detected by conducting manual or automated process analyses.

Errors related to people-centric requirements are generally hard to detect. If accountability and responsibility requirements are implemented using IT, errors can be detected using dynamic event or static log analysis. For other people-centric requirements, where IT support is not necessarily available, the situation and skills of the involved people may also have to analyzed manually.

Recovery from erroneous states can be achieved by defining and executing exception activities, when non-compliant erroneous states with the potential to lead to non-compliance are detected. Such states must be replaced by error-free (i.e., compliant) states. Assuming an error related to a retention requirement has been detected and data were deleted too early, in some cases recovery may be possible by re-collecting the data or by regenerating the needed data from a backup system. In other cases, data which should have been deleted earlier can be removed at the point of detection. Furthermore, recovery from malfunctioning access control policies can be achieved by implementing exception mechanisms allowing for selective blocking.

With respect to process-centric requirements, whenever some activity should have been executed and was not, or when a process was not executed in the correct order, or respecting certain timing constraints, different notions of rollback (to an earlier, saved state during process execution) or rollforward (to a new, error-free state) may be possible to recover from the erroneous state. For example, during the opening of a new bank account, the corresponding process may suddenly enter an erroneous state, because the responsible account manager tentatively opened the account and failed to compare the name of the customer with a list of known terrorists. In such a case, a rollback to the last state before the required activity would have to be performed by setting all data changed by the erroneous process activities back to their original value. In the case of missing documentation, it may be possible to update missing or erroneous documentation through a rollforward.

Recovery from errors of people-centric requirements can generally only be addressed using rollforward, specifically through the clear assignment of responsibilities, the implementation of systematic information collection and reporting, or through coaching.

Error Compensation In a compliance context, error compensation means that compliant business operations can also be provided in the presence of accidental or malicious errors [5]. For example, a fraudulent manager may try to value a traded asset higher than the current market value and enter this figure into the accounting system. An error-compensating compliance management system would cross-check the actual market value and realize that the value entered lies above it (e.g., by invoking the notorious stock quote web service). Consequently, the system could autonomously decrease the value or raise an error signal resulting in an external compensation action.

Other methods for addressing error compensation with respect to data-centric requirements include using error-correcting codes (ECC) and the four-eyes principle. Also the use of redundant backup systems, firewalls, fault-masking (e.g., self-checking components, k-out-of-n threshold schemes) or multi-layered security concepts can help sustain compliance and the integrity of data [19]. Process-centric requirements may be addressed by implementing default exception handling procedures, or by including redundancy into the process documentation. Finally, error compensation with regard to people-centric requirements comprises the use of delegation chains for addressing errors with respect to responsibility, and the use of knowledge bases and comprehensive documentation to support knowledge-specific requirements.

3.2.2 Fault Treatment

The categories of fault treatment, fault diagnosis and fault passivation, may be applied to compliance management as follows.

Fault Diagnosis Fault diagnosis is geared towards identifying the fault, i.e., the cause of the error, which leads to a compliance failure. For instance, let us assume an unauthorized user of an accounting system has modified his travel expense account, leading to a failure in that the company's balance sheet is not compliant to generally accepted accounting principles. We suppose the failure occurred because of an erroneous access control system. Fault diagnosis might then reveal that the error was due to misconfiguration, with the specific fault being that the respective user was mistakenly defined as a member of a group with access to the system.

Fault diagnosis of both data-centric and process-centric requirements requires manual or partially automated analysis of policies, process definitions, or source code (code review and code inspection). With respect to process documentation and people-centric accountability requirements, the correct assignment and documentation of responsibilities should be reviewed. Further diagnosis methods for people-specific requirements include interviews, skill analyses and examinations.

Fault Passivation Fault passivation for data-centric regulatory requirements can be addressed using isolation and reconfiguration [16]. Through isolation, the faulty physical or logical components are excluded from further delivering the service that failed to satisfy some requirement. Thus the fault is made dormant. Using reconfiguration, faulty components can be replaced by spare components (e.g., a new access control policy or system), or tasks are reassigned among non-faulty components (e.g., delegate access to a particular resource to some other compliant access control system).

Process-centric requirements can be addressed using isolation and reinitialization. A conceivable reinitialization method is, for example, to check and update an existing process definition and to run the faulty process again. Faulty documentation must be updated and the responsible fault is to be passivated by reconfiguring the responsible documentation procedure.

Faults in the context of people-centric requirements can be addressed by isolation (e.g., isolation and replacement of the faulty person) or reconfiguration (e.g., by reassigning tasks among compliant employees). Further means to addressing fault passivation with respect to people-centric requirements include the implementation of RACI matrices (Responsible, Accountable, Consulted, keep Informed) [21] and various types of education.

3.3 Fault Forecasting for Compliance

Fault forecasting is done by performing an evaluation of the system behavior with respect to the occurrence or activation of faults. The probabilistic and non-probabilistic methods of fault

		Fault Forecasting	
		Non-prob. Methods	Prob. Methods
Requirements Categories	Data-centric	Failure Mode and Effect Analysis, Reliability Block Diagrams, Fault Trees	Probabilistic Fault Trees, Markov Chains, Stochastic Petri Nets, Stochastic Processes, Modeling and Evaluation Testing
	Process-centric	Failure Mode and Effect Analysis, Reliability Block Diagrams, Fault Trees	Probabilistic Fault Trees, Markov Chains, Stochastic Petri Nets, Stochastic Processes, Modeling and Evaluation Testing
	People-centric	Failure Mode and Effect Analysis, Reliability Block Diagrams, Fault Trees, Assessment of Control Systems	Probabilistic Fault Trees, Stochastic Petri Nets, Modeling using Game Theory and Principal-Agent Theory

Table 2: Fault forecasting methods for addressing requirements categories.

forecasting summarized in Table 2 may be reused to address particular requirements categories.

3.3.1 Non-probabilistic Methods

Non-probabilistic methods try to identify and analyze possible faults that occur with respect to the intended or proper behavior of the system. The qualitative evaluation aims at the identification, classification, and ranking of the failure modes or event combinations that may lead to compliance failures. It also includes the appropriate management of identified fault risk (i.e., tolerance, prevention, removal, or transfer).

While faults relating to both data-centric and process-centric requirements can be forecast using failure mode and effect analysis, reliability block diagrams, and fault trees [17], fault forecasting methods with respect to people-centric may additionally necessitate a qualitative assessment of internal control systems. For example, when forecasting the likelihood of a fault relating to the effectiveness of an enterprise's access control system to protect the confidentiality and integrity of financial data, a fault tree analysis can be conducted to better understand which combination of faults and additional conditions may lead to a compliance failure. One may also identify varying degrees of severity with respect to the likely consequences of such a potential failure.[7] Forecasting of faults with respect to a people-centric requirement addressing responsibility and accountability of certain job roles may also require a thorough assessment of the systems that actually communicate and enforce these responsibilities.

3.3.2 Probabilistic Methods

Probabilistic fault forecasting means evaluating enterprise compliance in terms of the likelihood of fault occurrence and the implied failure probabilities. This entails assessing the likely consequences of the related risk of non-compliance (e.g., expected penalties) that the company may face for being non-compliant. It further includes the specific risk-appetite of the company, the cost to implement suitable control measures, as well as the forecast cost of non-compliance. Useful techniques with respect to data-centric and process-centric requirements categories are probabilistic fault trees [17], Markov chains [22], stochastic petri nets and stochastic processes [20], and modeling and evaluation testing. In addition, (malicious) faults concerning people-centric requirements can be analyzed ex ante and forecast using game theory [8] and formal models inspired by economics of information, in particular principal-agent theory [18, 14].

3.4 Fault Prevention and Removal for Compliance

As in the case of dependability and security also in the context of enterprise compliance fault prevention is a crucial part of general engineering (e.g., sound system design, formal verification, systematic testing, etc.) [19, 2, 17]. Fault prevention is also an obvious aim of development,

[7] In the context of the Sarbanes-Oxley Act, this would translate to an assessment of which faults in what exact combination will eventually produce *deficiencies*, *material weaknesses* and so on [23, 12].

			Fault Prevention and Removal			
			Verification	Diagnosis	Correction	Auditing
Regulatory Requirements Categories	Data-centric	Content	Data Collection and Flow Analysis	Static Fault Analysis	Enforce/Deny Data Collection	Data Inspection
		Retention	Code, Process, Policy Inspection	Policy, Code Analysis	Policy, Code Modification	Data Availability Analysis, Policy Inspection
		Reproduction	Random Sampling	System Compatibility Assessment	Pro-active Migration (Plan)	Random Testing, Demonstration
		Security	Vulnerability Testing and Correctness Proofs	Static Fault analysis	Policy, Process, Code Modification	Audit Trail/Log Analysis
		Privacy	Policy, Process, Code Inspection	Static Fault Analysis	Policy/Code/Process Modification	Random Testing
	Process-centric	Activity	Static Analysis (Inspection, Walk-through)	Static Fault Analysis	Process/Code Modification	Walk-through, Process Inspection
		Ordering	Model Checking, Theorem Proving	Static Process, Code Inspection	Process/Code Modification, Tests	Evidence/Audit Trail/ Log Analysis
		Timing	Simulation (Symbolic Execution), Testing	Static Process, Code Inspection	Process/Code Modification, Tests	Evidence/Audit Trail/ Log Analysis
		Documentation	Inspection, Random Testing	Process/ Responsibility Analysis	Update Documentation	Evidence (Random Testing)
	People-centric	Accountability	Job Description/ Responsibility Assignment Inspection	Process and Responsibility Assessment	Procedure change	Random Testing, Evidence (Audit Trails, Logs, Testimonies)
		Knowledge/ Belief	Interview, Control System Assessment	Analysis, Testing	Coaching, Update Processes, Control, Reporting System	Interviews, Assess Control and Reporting System, Evidence
		Education/ Training	Skill Assessment, Examination	Analysis, Testing	Coaching, Training, Education, Replacement	Interviews, Certification, Evidence

Table 3: Fault prevention and removal techniques for regulatory compliance.

control and process methodologies such as the earlier mentioned CMMI, COSO, COBIT, ITIL, or ISO 17799. By adopting and adapting such standards, enterprises get well-defined processes for most of their operational functions. Thus, they are less likely to ignore important aspects of running their business and may prevent many faults that would have occurred otherwise. The continuous improvement cycles mentioned earlier, which are inherent in most methodologies, are a further step to preventing or removing faults related to compliance a priori. Fault removal can be addressed both during the engineering of system components (development) and during their use. Table 3 presents an overview of the most suitable means known from dependability research to address specific regulatory requirements categories.

3.4.1 Verification

Dependability research differentiates between static and dynamic verification. Static verification refers to verifying a system without actually executing it. Verifying a system by executing it (or parts of it) constitutes dynamic verification.

In a compliance context, useful techniques to address data-centric categories are static analysis, in particular code and process inspections, data collection and flow analysis, and vulnerability search [17, 5]. Process-centric categories can also be verified using static analysis (process review, random testing and inspection of documentation), model checking [7] and theorem proving [6]. In this context, we use the term compliance checking to refer to the application of model-checking methods for verifying compliance of, e.g., a business process definition against temporal regulatory requirements expressed in linear temporal logic. As human behavior cannot easily be simulated, people-centric requirements are generally hard to verify a priori. Possible means are static analyses of job descriptions, responsibility assignments, control systems, and documentation. Further means are to conduct skill assessments, interviews and examinations.

With respect to proactive compliance management, the verification of controls and processes should always provide evidence of their effectiveness. Such evidence can be used in later audits.

3.4.2 Diagnosis

Diagnosis refers to understanding the exact nature of a fault, i.e., the specific cause of an error that potentially leads to a compliance failure. In many cases, the results of the earlier verification step constitute the starting point for the diagnosis and need to be carefully analyzed. Often, however, additional tests or in-depth analyses are needed in order to understand the detailed nature of an identified fault. For example, when a symbolic execution of a process definition results in a compliance failure with respect to a timing requirement, often a manual follow-up inspection or a walk-through of the processes definition needs to be performed. In an enterprise compliance context, diagnosing a fault also means carefully assessing the severity of the fault. This is important to evaluate the likely consequences of the occurred fault and to prioritize corrective measures.

3.4.3 Correction

Correction refers to the removal of possible error sources, i.e., flawed code, policies, processes or procedures, which were diagnosed to have some failure potential. Naturally, this requires that the respective faults have been correctly identified and understood in the earlier diagnosis step. In general, whenever standardized IT processes are in place, the correction of an identified fault requires a formal change request, which leads to the prioritization of the change and to the decision on how to optimally remove the fault (e.g, modification of faulty process, policy, or code; education; replacement of responsible system, person, etc.).

3.4.4 Auditing

Meadows and McLean [19] proposed extending the earlier taxonomy of Laprie [15] with a forth step: *convincing*. While we would agree that this constitutes an important task for dependability and security, in the enterprise compliance context this fourth step is crucial. In a compliance management setting, convincing translates to *auditing*, which denotes the task of (external or internal) auditors assessing the conformance with regulatory requirements. Failure to pass external audits may result in both explicit (e.g., fine and the obligation to correct the identified deficiency within a certain time) and implicit (e.g., loss of investor confidence and a decrease in the share price) penalties. Because of this, audits per se and methods to support assessing and demonstrating compliance in audits are very important. Auditors often use standardized frameworks and random tests to assess whether an enterprise reached the necessary degree of compliance. Increasingly, they require the enterprise to demonstrate compliance (e.g., by requiring them to provide evidence that they tested the effectiveness of the control measures in place as in the case of the Sarbanes-Oxley Act or the Gramm-Leach-Bliley Act). As a result, traceability and the proper documentation of regulatory requirements (e.g., justifying how and why a certain requirement has been implemented in a specific manner) is increasingly important.

Auditing of compliance with data-centric requirements can be done using manual data and policy inspection based on random testing of audit trails. Process-centric requirements with respect to the existence, ordering, timing, and documentation of particular activities, can be done using either manual process walk-throughs and inspections, or by applying the earlier-mentioned automated verification methods to specific audit trails. Auditing can also be done by querying activity log databases and sometimes processes may even be reconstructed and compared to required work flows [1]. Finally, auditing compliance with people-centric requirements generally requires conducting interviews, interrogations, and assessing control systems and audit trails.

4 Conclusion

In this paper, we have stressed the important role of dependability and security in the context of compliance management. Specifically, we have described why and how dependability can help to attain and sustain enterprise compliance. Understanding an enterprise as a complex system striving to be compliant with relevant regulatory requirements, we have proposed two new failure types, namely functional and structural compliance failures. Moreover, we have presented a categorization of regulatory requirements and a set of important meta-requirements. In this context, we have proposed compliance as an additional attribute to extend the existing set of attributes characterizing dependability and security. By combining the requirements categories and the classical dependability and security taxonomy, we have demonstrated how enterprise compliance can be understood in terms of the well-defined concepts of dependability. With the resulting classification framework, we have identified existing and new methods to attain compliance with individual compliance categories and, additionally, we have pinpointed areas where feasible solutions are either impractical or still missing.

Acknowledgements

I thank Christian Cachin, Birgit Pfitzmann, HariGovind V. Ramasamy, and Dieter Sommer for informal discussions and helpful comments on an earlier version of this paper. The work reported in this paper is supported by the European Commission through the IST Programme under Contract IST-4-026764-NOE ReSIST.

References

[1] R. Agrawal, C. Johnson, J. Kiernan, and F. Leymann. Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In *ICDE*, page 92, 2006.

[2] A. Avižienis, J.-C. Laprie, B. Randell, and C. E. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[3] Carnegie Mellon Software Engineering Institute. Capability Maturity Model Integration (CMMI), 2005. `http://www.sei.cmu.edu/cmmi/`.

[4] Committee of Sponsoring Organizations of the Treadway Commission (COSO). Internal Control – Integrated Framework, 1992. Jersey City, NJ: AICPA/COSO.

[5] Conceptual Model and Architecture of MAFTIA. Technical report, Jan. 2003. Also available as Research Report RZ 3473, IBM Research.

[6] D. A. Duffy. *Principles of Automated Theorem Proving.* John Wiley & Sons, Inc., New York, NY, USA, 1991.

[7] J. Edmund M. Clarke, O. Grumberg, and D. A. Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 1999.

[8] R. Gibbons. *A Primer in Game Theory.* Pearson Education Limited, 1992.

[9] C. Giblin, A. Y. Liu, S. Müller, B. Pfitzmann, and X. Zhou. Regulations Expressed As Logical Models (REALM). In M.-F. Moens and P. Spyns, editors, *Proceedings of the 18th*

Annual Conference on Legal Knowledge and Information Systems (JURIX 2005), volume 134, pages 37–48. IOS Press, December 2005.

[10] Gramm-Leach-Bliley Act of 1999 (GLBA), 1999. Public Law 106-102, 113 Stat. 1338.

[11] ISO/IEC. Information Technology - Security Techniques - Code of Practice for Information Security Management, 2005. ISO/IEC 17799.

[12] IT Governance Institute. IT Control Objectives for Sarbanes-Oxley, 2004.

[13] IT Governance Institute. Control Objectives for Information and Related Technology (COBIT), 2005. Version 4.0.

[14] J.-J. Laffont and D. Martimort. *The Theory of Incentives: The Principal-Agent Model.* Princeton University Press, March 2002.

[15] J.-C. Laprie. Dependability: Basic Concepts and Terminology. *Dependable Computing and Fault Tolerant Systems*, 5, 1992.

[16] P. A. Lee and T. Anderson. *Fault Tolerance: Principles and Practice.* Springer Verlag, 1990.

[17] N. G. Leveson. *Safeware: System Safety and Computers.* ACM Press, New York, NY, USA, 1995.

[18] I. Macho-Stadler and D. D. Perez-Castrillo. *An Introduction to the Economics of Information: Incentives and Contracts.* Oxford University Press, 2nd edition, February 2001.

[19] C. Meadows and J. McLean. Security and Dependability: Then and Now. In *Computer Security, Dependability, and Assurance: From Needs to Solutions*, pages 166–170. IEEE Computer Society, 1999.

[20] J. Medhi. *Stochastic Processes.* Wiley Eastern, New Delhi, 1994.

[21] Office of Government Commerce (OGC). IT Infrastructure Library (ITIL), 2006.

[22] M. L. Puterman. *Markov Decision Processes.* John Wiley & Sons, New York, NY, 1994.

[23] Sarbanes-Oxley Act of 2002, 2002. Public Law 107-204, 116 Stat. 745.