Compliance Assessments of Projects Adhering to Enterprise Architecture

Ralph Foorthuis, UWV Business Services, The Netherlands

Frank Hofman, Statistics Netherlands, The Netherlands

Sjaak Brinkkemper, Utrecht University, The Netherlands

Rik Bos, Utrecht University, The Netherlands

ABSTRACT

This article examines how to assess projects, which implement business processes and IT systems, on compliance with an Enterprise Architecture (EA) that provides them with constraints and high-level solutions. The authors begin by presenting the core elements of EA compliance testing. Next, the authors discuss the testing process and four types of compliance checks (i.e., correctness check, justification check, consistency check, and completeness check). Finally, an empirical case is reported in which a real-life project has been tested on conformance, demonstrating and evaluating the authors' approach. The results indicate that objective compliance testing cannot be taken for granted. Therefore, several suggestions are presented to decrease the subjectivity of assessments, such as operationalization of EA prescriptions.

Keywords: *Compliance Assessment, Compliance Checks, Conformance, Enterprise Architecture, Projects, Subjectivity*

INTRODUCTION

When studying the literature, Enterprise Architecture (EA) can be said to have two major ideal type functions. One function is to provide decision-makers with a clear and comprehensive *descriptive overview* of the organization, or relevant aspects thereof. Such insights into the enterprise form the basis for making high-level management decisions (cf. Johnson et al., 2004; Riempp & Gieffers-Ankel, 2007; Gammelgård et al., 2007), determining, e.g., which programs or projects to initiate. This reflective function of EA targets mainly managers as its users. The EA can be expected to demonstrate a heavy focus on depicting the (problematic) as-is situation. A second ideal type function of EA is to provide a *prescriptive framework* that guides and constraints subsequent development of business and IT solutions (cf. Kaisler et al., 2005; Boh & Yellin, 2007; Op 't Land & Proper, 2007; van Bommel et al., 2007; Foor-

DOI: 10.4018/jdm.2012040103

thuis et al., 2008; Hoogervorst & Dietz, 2008; Meschke & Baumoel, 2010). This normative approach, focusing strongly on the to-be situation, should ensure that both enterprise-level and local initiatives within the organization are consistent with the overall strategy, and enable a coherent and integrated development of business, information and IT. This directive function of EA targets not only managers as its users, but also business analysts, system analysts, software architects and other roles in projects (re)designing the business and its IT support. In this article, we focus mainly on this latter function, a prescriptive EA providing constraints and high-level solutions to which business and IT systems – and in particular the projects implementing them – should conform. Prescriptive EAs prove to be common in practice. One example is the Enterprise Architecture of a manufacturing company, which uses principles, policies and models to ensure that business and IT initiatives are consistent with the business strategy (Bruls et al., 2010). Another example is a national statistical institute's architecture, consisting of principles and models to which projects much adhere in order to save costs and increase the quality of statistical products (Foorthuis & Brinkkemper, 2008).

An EA's norms or prescriptions are often applied in projects. Although EA typically focuses on the entire enterprise and compliance is indeed demanded at this level, in practice it is unrealistic for an entire organization to become EA-compliant at short notice. It can therefore be expected that conformance will be achieved incrementally at the local level, step by step – or rather, project by project (cf. Ross et al., 2006). However, philosophers have acknowledged for hundreds of years that, although compliance with 'contracts' might be better for the group as a whole and it might also be in an individual actor's best interest to agree to contracts, it may not be in his interest to actually comply with them. In contractarian ethics this is one of the issues of the so-called *compliance problem* (cf. Gauthier, 1991; Hartman, 1996). Because of this potential conflict of interest, it should be tested whether actors actually conform to the contract. If we consider a specific project to be the actor, then an EA could be seen as the contract that needs to be complied with. In other words, although conformance is required for obtaining EA benefits, it cannot be expected to occur automatically (Boh & Yellin, 2007). This is especially relevant here as compliance with EA norms may be in the best interest of the organization as a whole, but not optimal per se to the local projects and departments that actually have to comply. Assessments should therefore be carried out at the level at which EA is applied, i.e., the project level. Testing at this level also allows for correcting non-compliant aspects, at least if it is performed while EA is being applied. Assessing projects on conformance is crucial, as a large survey study (n=293) has shown not only that project compliance with EA is positively associated with various strategic benefits, but also that the most important determinant of conformance is in fact conducting compliance assessments of projects (Foorthuis et al., 2010).

Emmerich et al. (1999) define compliance in the context of IT projects as "the extent to which software developers have acted in accordance with the 'practices' set down in the standard." Kim (2007) defines compliance in this context as "an accordance of corporate IT systems with predefined policies, procedures, standards, guidelines, specifications, or legislation." In the context of EA we define *compliance* as corporate business and IT systems being in accordance with predefined Enterprise Architecture prescriptions. We will use the terms "compliance" and "conformance" interchangeably. Likewise, "assessing compliance" and "testing on conformance" are considered equivalent. A "project" in this article refers to the regular projects that need to comply with Enterprise Architecture, which, by and large, have a localized scope (e.g., delivering a new business process and related IT applications for a department).

In this article, we aim to answer the following research question:

How can projects, and the business and IT solutions they deliver, be assessed on compliance with a prescriptive EA?

To answer the main research question, we will divide it into several sub-questions:

1. *What concepts play a key role in assessing compliance with EA?*
2. *By what process can EA compliance testing be carried out?*
3. *What kind of compliance checks can be utilized in the EA compliance test process, and what are their respective evaluation criteria?*

The underlying goal of our research is to identify and explore core aspects of testing projects on EA compliance. It is our intention to stimulate additional research into the topic. A second, more practical goal is that the results should provide organizations with a working model that can be used to develop their approach for testing their change initiatives on EA conformance.

This article will proceed as follows. In the next section, related topics and work are discussed. Following that, we position our study in the context of EA and describe the research approach. The subsequent sections aim to find answers to the respective sub-questions and present our empirical case. The final section is for discussion and conclusions.

RELATED TOPICS AND WORK

Although we did not find any academic work dedicated to assessing compliance with EA at the time of our research, the topic can nonetheless be linked to other work. In particular, EA conformance testing is related to the fields of *compliance management, software testing* and *auditing*. In terms of compliance management, several areas relevant to our discussion can be acknowledged. First, due to *legislation*, organizations are required to comply with regulations that have consequences for their business processes and information systems. Non-compliance here may even have penal consequences for an organization's management (El Kharbili et al., 2008). In Europe, important drivers are Directive 95/46/EC, i.e., the Data Protection Directive, and Directive 2002/58/EC, i.e., the Privacy and Electronic Communications Directive (Massacci et al., 2005; Nouwt, 2008). Examples of laws in the United States which demand compliance are the Sarbanes-Oxley Act, the Gramm-Leach-Bliley Act and the Health Insurance Portability and Accountability Act (Kim, 2007; Lankhorst, 2005; zur Muehlen et al., 2007). The Basel Accords, featuring regulations for capital adequacy of the banking sector, form an example of a global regulatory framework (Barr & Miller, 2006).

A second area in compliance management is consistency with international and industry-wide *standards* for processes and products, such as ISO 9001 for quality management and IEC 61508 for safety. There are several reasons for conforming to such best practices, for example clients or strategic partners demanding certification for assurance reasons, and using best practices to improve the organization's processes and products. Conformance to standards is especially important in large and critical systems engineering projects in, e.g., the defense, aerospace and telecommunications sectors. See Emmerich et al. (1999), Pfleeger et al. (1994), and Chung et al. (2008) for more about compliance with standards. We will employ some of the concepts in these publications in our own research.

A third relevant area is *security and risk management*, which aims to protect the organization's assets, such as valuable information. Compliance here has an important role to play in preventing both deliberate and unintentional harm to the organization, e.g., by imposing access restrictions. See for example von Solms (2005), Drew (2007), and Vroom and von Solms (2004) for more on this topic.

All three areas are relevant to our discussion, as an EA can feature constraints and high-level solutions based on any of the above. Needless to say, they are not mutually exclusive.

For example, security and risk management are principal concerns of the Basel framework and of international standards such as ISO/IEC 27000.

Assessing projects and their products on compliance with EA can also be related to software testing. Several core elements can be distinguished in this discipline (Baresi & Pezzè, 2006; Binder, 2000). First, *test items* refer to the items that need to be tested, e.g., a document or a version of an application. Secondly, the *features* are the specified properties that the test item is required to possess. Thirdly, *acceptance criteria* are needed to decide whether the software is ready for successful usage in the business setting. This is relevant because features are not sufficient for testing, as not every feature is equally important and features may be only partially implemented. Finally, a *test approach* is needed to define the testing techniques to be used in determining whether the test item possesses the features to an acceptable degree. In this article we will translate these software testing concepts to the domain of EA conformance testing.

Finally, it is interesting to mention the similarities between the compliance test discussed here and an audit. According to IEEE (1990), an *audit* is "an independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria." Similar to software testing, an audit has several elements (IFAC, 2003), such as the *subject matter* (a work product) that is evaluated against the *criteria* (benchmarks), leading to an *assurance report* (containing a conclusion on whether the subject matter conforms to the criteria). If the goal of an audit is to assess compliance of designs with an Enterprise Architecture, then an audit and the EA assessment discussed in this article should be very similar. However, if an audit is to assess whether a business unit does in practice what is intended, then an audit may be a compliance assessment after *run-time* (also taking into account actual execution traces such as process logs). This differs from this article's assessment of a project, which is conducted in *design-time* (taking into account artifacts that describe designs of processes and systems) and as such offers preventative potential (Sadiq et al., 2007). Another difference is the fact that the approach described in this article is specialized for EA compliance testing. This results not only in several EA-specific concepts being used, but also recognizes the strategic and abstract nature of EA.

POSITIONING THE RESEARCH

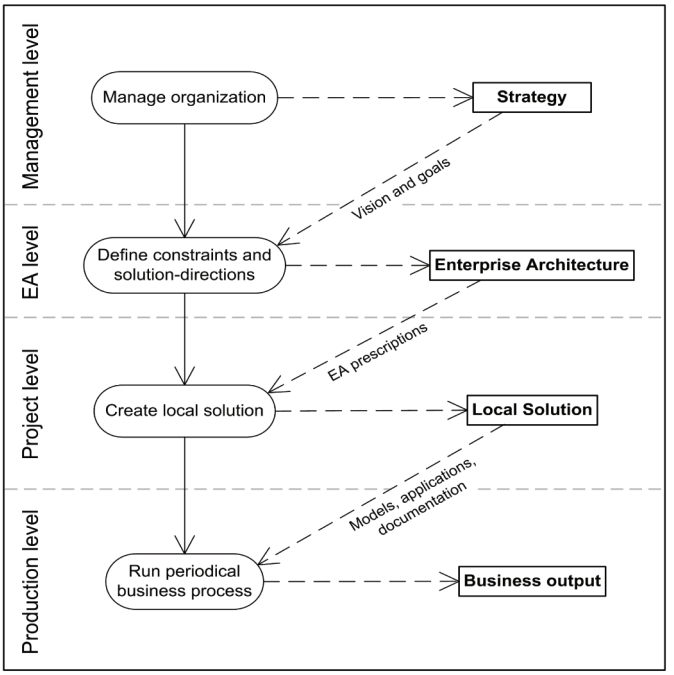
Figure 1 depicts the different levels involved in working with EA and the relationships between the processes at these levels. The output of each level is input for the lower level. A rounded rectangle represents a process, whereas a square rectangle represents the input and output of a process. In addition, a continuous line denotes the process flow; a dashed line denotes an information or product flow.

The diagram should not be regarded as modeling one single process, but rather as identifying four distinct processes, each at a different level. The model explicitly shows that the output of each process is the input for the level below. Feedback can certainly flow from lower to higher levels, but in order to focus on the essence we have abstracted from that in this diagram. The output of each of the processes will be described in more detail in the next section.

This article focuses on testing whether the Local Solution does indeed conform to EA. In other words, on assessing whether the project has correctly applied the EA prescriptions in creating the solution. We will therefore focus primarily on the project level, as we expect a Strategy and Enterprise Architecture to be given, and the production process generating Business Output can only be run after the Local Solution has been delivered and adjudged compliant with EA.

As a final remark, please note that, in addition to an EA, an organization can also have

Figure 1. High-level overview of the processes related to working with EA



one or more Domain Architectures. We will not discuss this here, however, since we consider assessing compliance with Domain Architectures to be very similar to EA compliance testing.

RESEARCH APPROACH

We adopt a design science approach for this study, as methods for assessing projects on compliance with EA is a relevant topic that has not yet received much attention. Design science seeks to create innovative artifacts with the underlying goal to make the analysis, design, implementation, management, and use of information systems more effective and efficient (Hevner et al., 2004).

A distinction can be made between several types of research outputs or artifacts (March & Smith, 1995; Hevner et al., 2004). First, *constructs* form the formal or informal vocabulary or language of a discipline. An example is the rules for creating a class diagram. Secondly, a *model* is a set of propositions expressing relationships among constructs, representing for example problem and solution statements. A *method* is a set of steps used to perform a task in order to obtain a certain result. A method is based on underlying constructs and models, for example because the steps take parts of a model as input. It can take the form of algorithms or guidelines. An example is a systems development method. An *instantiation* demonstrates the feasibility and effectiveness of the models and methods they contain, and thereby provides the empirical part of the study.

In order to answer sub-question 1, we will present in the next section a model describing the key concepts in EA compliance testing. In answering sub-questions 2 and 3, we will subsequently present as a method a set of steps and compliance checks that allow a tester to assess the degree of compliance. This method is the design science artifact that is evaluated and demonstrated in this article. This is done

by instantiating (putting to practice in a real-life situation) the steps and checks, and by providing relevant statistical metrics.

FUNDAMENTAL CONCEPTS IN EA COMPLIANCE TESTING

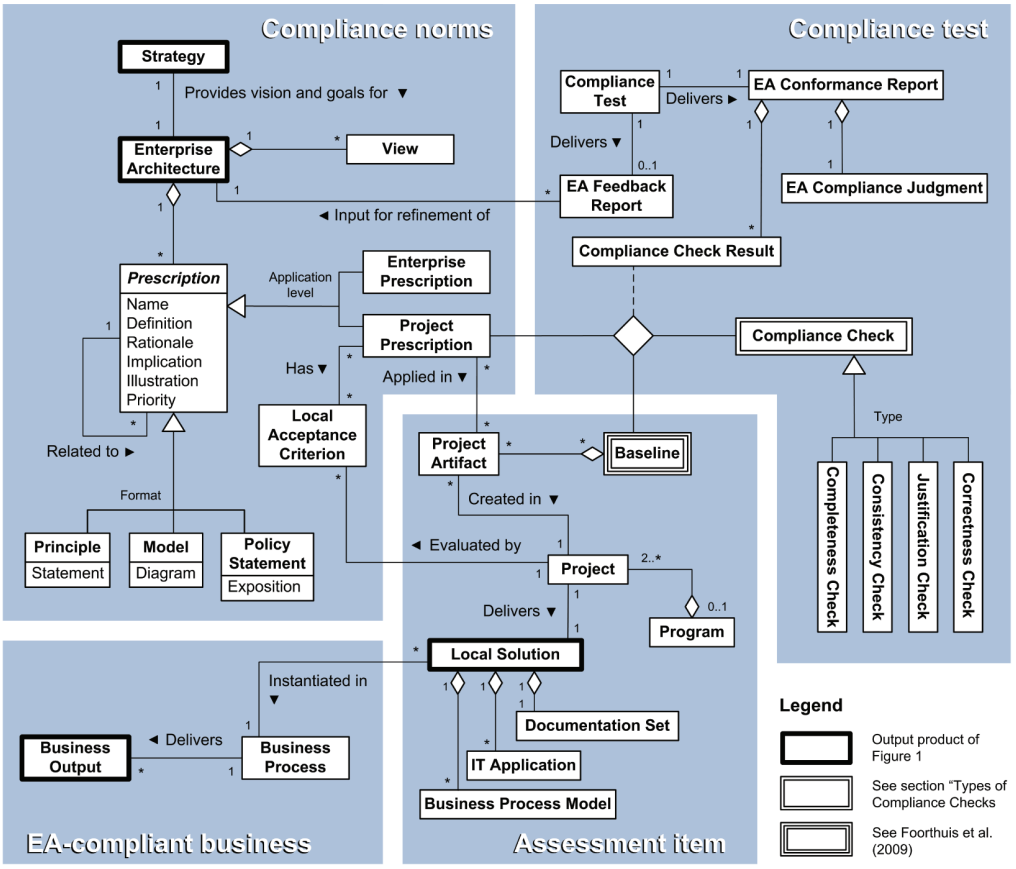
This section presents an overview of the core elements of EA compliance testing, represented in the EA Compliance Model of Figure 2 as a UML class diagram. The bold-outlined classes are the four output products of Figure 1. The double-lined class (the Compliance Check) will be described in more detail in the section "Types of Compliance Checks". The triple-lined class (the Baseline) is described in more detail in the referred paper. Since the model will function as the basis for the remainder of our article, its contents will, where relevant, be supported by literature. We have used the model of Emmerich et al. (1997, 1999) as inspiration since it aims at testing on compliance with standards. It also takes as input documents (similar to our project artifacts). Furthermore, it subdivides the model in various parts (similar to the four high-level concepts or grey areas in Figure 2). A difference is that Emmerich et al. focus on automated compliance checking (whereas we perform manual checks) and on the field of software development (whereas we focus on the broader and more strategic field of EA).

Four high-level concepts can be acknowledged in compliance testing, represented by the grey areas. These are inspired by the aforementioned core elements of software testing and auditing. First, analogous to software testing there is an *assessment item*, which needs to be tested. This is the set of project artifacts, in which the EA prescriptions should have been applied. An artifact here is a deliverable or intermediate work product, such as a software architecture document (note that this is different from the design science artifact evaluated here). Secondly, a set of *compliance norms* functioning as an evaluation frame is required. These are the EA's prescriptions, possibly complemented with local acceptance criteria. Thirdly, an approach or *compliance test* will be used to establish (non-)compliance of the items. This comprises several types of compliance checks. Finally, the *EA-compliant business* represents the desired result. We will discuss the model in more detail below. The individual classes of Figure 2 will be directly referred to using Capitalized Names, while properties will be referenced with *Italic Capitalized Names*.

An enterprise's Strategy will provide the input for the Enterprise Architecture, as an EA is a governance instrument intended to facilitate the translation from corporate strategy to daily operations (Jonkers et al., 2006). The resulting EA consists of Views and Prescriptions (Foorthuis et al., 2008). A View typically provides insight into the context and meaning of a system (e.g., an entire enterprise, an IT system or a business service), and its fundamental organization, components and their relationships. As such, a View can depict both the as-is and the to-be situation. It can be utilized as a cognitive aid, in the form of an overview (e.g., a context model), a frame of reference (e.g., a structuring mechanism for analysis purposes), or a shared vocabulary (for communication purposes). A Prescription, focusing solely on the to-be situation, has an explicit guiding function and is required to take the form of a Principle (textual statement), Model (visual diagram) or Policy Statement (exposition containing text and possibly diagrams). These types of Prescriptions explicitly provide constraints or directions and are therefore more directly related to compliance than a View. An example of a Prescription is the principle "Data suitable for re-use shall be identified and stored in enterprise-wide data stores."

A Prescription is a relatively stable fundamental norm or guide that has to be complied with. As the Prescription is the central element in the model, it is presented along with its properties. These properties will be used in the section "Types of Compliance Checks" to identify and define types of checks. They are based in part on the template for describing principles, as defined by Richardson et al. (1990) and The Open Group

Figure 2. The EA compliance model



(2009). The first property is the *Name*, which should succinctly and identifiably refer to the essence of the Prescription. Secondly, the explicit *Definition* is the compliance requirement, presented as clearly as possible in the form of a Statement, Diagram or Exposition[1]. A third important property is the *Rationale*, providing the reasons behind the Prescription and thereby elaborating on the business benefits achieved by adhering to it. It should make clear why and when the prescription can be effective, and could as such motivate compliance (Emmerich et al., 1999). Fourthly, the *Implication* describes the (potential) impact and consequences of applying the Prescription in terms of costs, resources and activities. This is input for a cost-benefit analysis when deciding whether or not to apply it and can provide information on how to apply it in practice. The fifth property, the *Illustration*, is valuable because examples can clarify Prescriptions that are inherently ambiguous as a result of their generic nature (Foorthuis & Brinkkemper, 2008). Finally, the *Priority* indicates the importance of the prescription, stating whether it is mandatory or merely recommended.

A Prescription can be related to other Prescriptions. For example, prescriptions can be ordered *hierarchically* (which is relevant if the EA framework features abstraction levels). The *counterpart prescriptions* described in Foorthuis and Brinkkemper (2008) are another

example, in which business Prescriptions with IT implications have closely related counterpart IT prescriptions, and vice versa. As such, they are a mechanism for improving business-IT alignment. In addition, a Prescription can be an Enterprise Prescription or a Project Prescription. The first provides generic constraints (boundaries) and directions (high-level solutions) for an entire enterprise. Prescriptions applied at this level can as such guide the outlining of the enterprise's policy or direct the development and evolution of enterprise-wide services. A Project Prescription, provides generic constraints and directions for localized Projects (or rather, their products). Projects and compliance testing may also need to take into account Local Acceptance Criteria. The reason for this is that the specific situation to be assessed might call for ad hoc variations, e.g., exempting the project from certain EA prescriptions in the case of urgency.

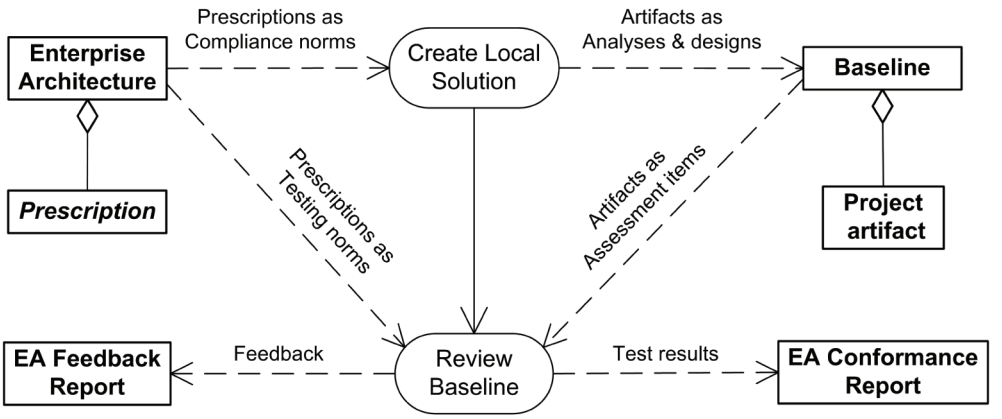
Project Prescriptions are applied by Projects in their Project Artifacts, i.e., deliverables such as software architecture documents. A Baseline collects several Project Artifacts that are reviewed and agreed on by their immediate stakeholders and which form the basis for further development (IEEE, 1990). Through their explicit or implicit application in Project Artifacts, Project Prescriptions can guide the development and evolution of local initiatives by providing constraints and directions which the Projects implementing the solutions should adhere to. A Project, which may be part of a larger program, delivers a Local Solution. This consists of a newly designed Business Process Model, a newly developed IT Application and a Documentation Set (i.e., manuals and the final Baseline). Generating Business Output means instantiating the Local Solution in a Business Process, which involves planning and running a real-life instantiation of the process and IT application delivered by the project. First, however, compliance with EA needs to be assessed.

Key elements in performing the Compliance Test are Compliance Checks, norms (i.e., the EA prescriptions and Local Acceptance Criteria functioning as an evaluation frame) and a resulting EA Conformance Report (cf. Emmerich et al., 1999; Baresi & Pezzè, 2006; Chung et al., 2008; Foorthuis et al., 2008; Farenhorst & De Boer, 2009). A Baseline provides an ideal opportunity for this compliance assessment, as it describes the agreed-upon basis for the remainder of the project and still allows for intervention in case of non-compliance. As the ternary association class shows, the Compliance Check Result is the product of the EA's Project Prescriptions, the Project's Baseline to be tested and the types of Compliance Checks. In other words, given a specific Baseline to be tested, several compliance checks are performed for each Prescription, resulting in a number of Compliance Check Results. See Table 1 in the section "Empirical Evaluation" for an example of such test results for a given Baseline. Each individual (non-conformant) Compliance Check Result will be an entry in the EA Conformance Report. Four types of Compliance Checks will be identified in the section "Types of Compliance Checks". The EA Conformance Report also contains a final EA Compliance Judgment, which is the test conclusion stating whether or not the assessed item (i.e., Baseline) complies or not. Finally, the Compliance Test may yield an EA Feedback Report, which provides valuable information to the enterprise architects for updating the EA.

THE PROCESS OF COMPLIANCE TESTING

In this section we will describe the process of compliance testing (i.e., the design science artifact of this study). We will start by presenting several requirements for such a process. A first requirement is the separation of duties (i.e., checks and balances). An actor testing himself on compliance cannot be expected to always produce true and objective results (von Solms, 2005). An EA compliance assessment or audit should therefore be performed by other individuals and preferably other organizational units rather than those carrying out the respective project. In the context of this article, this means that if an enterprise architect actively partici-

Figure 3. Role of EA and project artifacts in carrying out projects and compliance assessments



pates in a project, he or she should not be the tester performing the conformance assessment.

A second requirement is that assessing EA compliance should not be carried out solely at the end or in the latter stages of the project, since by that time the architectural decisions will already have been implemented. Because such decisions are fundamental, they will be difficult to reverse at a later stage. Compliance testing should therefore be done at stages in the project's lifetime when fundamental analysis and design decisions have matured and have been explicitly stated, but not yet implemented. In this way, deviations from the architecture can be identified while there are still opportunities to correct them. There should therefore be multiple baselines. Ideally, when creating these baselines, the project will have already consulted an enterprise architect (Foorthuis et al., 2008).

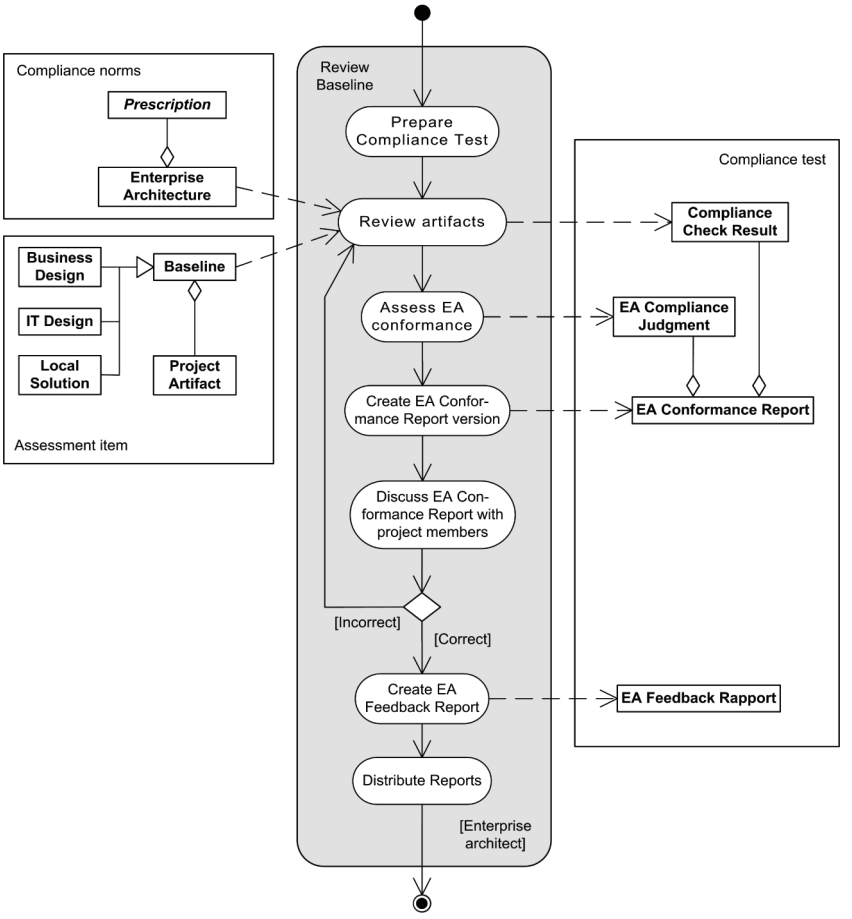
A third requirement is that, like in auditing, compliance testing should be part of a larger compliance initiative (cf. Hamilton, 1995; Burditt, 1996). To be more precise, the projects should be stimulated to comply from the start. Figure 3 shows the relationships between creating the project artifacts and assessing them on compliance. It shows prescriptions having two roles, those of steering norms and those of testing norms. The process "Create Local Solution" represents carrying out a project that is stimulated to conform to EA as described in Foorthuis et al. (2008). "Review Baseline" represents the compliance assessment process, which is based on that same study, quality aspects (Appendix B) and this study's empirical evaluation. We have modeled the steps of this testing process in detail in Figure 4, meeting the requirements mentioned above. The assessment is performed by the enterprise architect role. The reason for this is that this role is external to the project, has knowledge of the EA prescriptions (unlike regular auditors), has the interest to let projects conform, and (as EA often is not comprised of legal rules) is able to negotiate with the project.

The use of three baselines shows that the assessment can be carried out at three stages in the project's lifetime: after business analysis and design, after specification of functional requirements and software architecture, and after delivery of the final product.

In terms of notation, we used the technique presented in van de Weerd and Brinkkemper (2008). We will elaborate on the various steps of the model.

1. **Prepare Compliance Test**: The Enterprise Architect prepares the Compliance Test for use in the specific situation. This includes collecting the Baseline and obtaining the most recent versions of the Prescriptions present in the Enterprise Architecture docu-

Figure 4. Process model for compliance testing



mentation. In addition, the involved stakeholders should agree on a time planning.

2. **Review artifacts**: The Enterprise Architect reviews the Project Artifacts from the Baseline. Reviewing the artifacts means using Compliance Checks for assessing the EA Prescriptions that have been implicitly or explicitly applied in the Baseline's project artifacts. The four types of Compliance Checks (discussed in more detail in the next section) are the Correctness Check, the Justification Check, the Consistency Check and the Completeness Check. Applying them yields Compliance Check Results that (possibly only in the case of non-compliance) will be included in the EA Conformance Report.
3. **Assess EA conformance**: After reviewing the Project Artifacts, the Enterprise Architect passes an EA Compliance Judgment regarding the degree to which the project complies with the EA.
4. **Create EA Conformance Report version**: The Enterprise Architect creates a version of the EA Conformance Report.
5. **Discuss EA Conformance Report with project members**: The Enterprise Architect discusses the draft version of the EA Conformance Report with the authors of the assessed Baseline. The goal of this

step is twofold. First, to clarify the report, if needed. Secondly, to avoid Compliance Check Results (i.e., review comments) and an EA Compliance Judgment that are invalid due to an incorrect understanding of the Baseline and its knowledge domain. If changes in the EA Conformance Report are required, the Enterprise Architect goes back to the "Review artifacts" step.

6. **Create EA Feedback Report**: During the review process and the discussions with the project members, the Enterprise Architect may have discovered weak aspects of the EA. Furthermore, the test may have yielded ideas for additional or updated operationalizations of prescriptions (see the section "Empirical Evaluation"). These can be stated in an EA Feedback Report.
7. **Distribute Reports**: The Enterprise Architect distributes the EA Conformance Report to the relevant stakeholders. The EA Feedback Report is sent to the lead Enterprise Architect.

TYPES OF COMPLIANCE CHECKS

As shown in the EA Compliance Model we can distinguish between several types of compliance checks, which are used in the "Review artifacts" step of the process model. A *compliance check* is an analytical tool or mechanism to assess the current state of compliance (cf. Emmerich et al., 1997). When testing projects on EA conformance, several types of such checks can be distinguished, each assessing a specific aspect of compliance. Like the EA Compliance Model, the identified compliance checks are partly based on insights from the field of automated compliance testing (Chung et al., 2008; Emmerich et al., 1997, 1999). Examples of checks proposed there are the completeness and correctness check (Chung et al., 2008). As these are reminiscent of quality aspects of software engineering, data management and auditing (cf. van Zeist & Hendriks, 1996; Pipino et al., 2002; IFAC, 2003; Caballero et al., 2007), we have also studied whether some of these aspects might make for relevant EA compliance checks (Appendix B).

The resulting types of checks are described. For each type, the specific elements of the norms required for the assessment will also be mentioned (in terms of properties and relations of the classes of the EA Compliance Model depicted in Figure 2).

- *Correctness check*: verifies whether a given prescription is applied by the project in a way that is in accordance with its intended meaning, rationale and usage. In other words, this check verifies whether the application of the prescription deviates from the prescription as it was intended by the enterprise architects.
  In terms of the EA Compliance Model, the criteria needed for performing the correctness check can be found mainly in the Prescription's *Definition* and *Illustration* properties, as these serve to communicate its intended meaning. However, the *Rationale* and *Implication* may also be relevant here, as they elaborate on its value and usage.
- *Justification check*: verifies whether the (lack of) application of a given prescription is justified, depending on its relevance and priority in the specific situation. The justification check's actual execution is dependent upon certain conditions. First, if the application of a prescription *deviates* from its intended application (which is determined by the correctness check), it needs to be ascertained whether the alteration is justified. Secondly, if a prescription is *not applied*, it needs to be ascertained whether it is justified not to apply it. Thirdly, if a prescription is *applied correctly*, it needs to be checked whether it is indeed justified to apply it. This last sub-check aims to avoid 'blind' conformance which could unnecessarily harm project or enterprise goals in the specific situation. In short, the justification check verifies whether the project has made the appropriate choice

when deciding to either apply, alter or not apply a given prescription.

In the EA Compliance Model, the justification check's evaluation criteria can be found in the Prescription's *Rationale*. The rationale describes the prescription's benefits (which should be consistent with the local situation's objectives) and when it should be applied (which should be consistent with the nature of the local situation). In addition, the *Implication* may be relevant here, since the impact in terms of costs, resources and activities can play a role in the cost-benefit analysis. Furthermore, the *Priority* states whether prescriptions are mandatory or merely recommended guidelines.

- *Consistency check*: verifies whether, if a given prescription is applied, required related prescriptions are also applied. Some prescriptions, especially those at lower abstraction levels, might need to be implemented as a package. For example, the counterpart prescriptions mentioned in the previous section. Another focus of the check is to verify whether the prescriptions' applications do not contradict each other, but instead culminate in a consistent and balanced result.

  The consistency check's evaluation criteria can be found in the prescription's relationship with other prescriptions (i.e., the self-reference of the Prescription class).
- *Completeness check*: verifies whether all the prescriptions are applied. Minimally, the prescriptions that have been designated as mandatory (perhaps dependent on specific project situations) need to be applied, so as to avoid projects applying merely a convenient subset.

  The completeness check's evaluation criteria can be found in the Prescription's multiplicity with the Enterprise Architecture. It is the number of Prescriptions (that are of type Project Prescription) represented by the "*" symbol in a real-life instantiation of the aggregation between Enterprise Architecture and Prescription. Or put more simply: the total number of (mandatory) prescriptions relevant for projects. The *Priority* states whether prescriptions are mandatory or not.

The completeness and correctness check types are also mentioned in Chung et al. (2008) in their discussion of compliance with standards. We have adapted them here to fit the EA context. The justification and consistency check types are contributions of the current research (including the study of quality aspects; see Appendix B). We have added the justification check because the relevance of prescriptions can be conditional (cf. Pfleeger et al., 1994) and local acceptance criteria might need to be taken into account. The idea for the consistency check is supported by the respective quality aspect (Pipino et al., 2002). This check is especially relevant in the context of Enterprise Architecture, because EA aims for a coherent development of business, information and IT, but at the same time has to deal with potentially conflicting stakeholder interests and requirements (cf. The Open Group, 2009). Other quality aspects mentioned were not relevant in the EA context.

The correctness and justification checks are performed at the level of an individual Prescription. The completeness check is done at the level of the entire collection of Project Prescriptions. The consistency check is performed at the level of a group (package) of individual Prescriptions.[2] This is illustrated in Table 1 of the "Empirical Evaluation" section.

The checks can be applied to all Prescriptions, regardless of whether the Enterprise Architecture focuses on all the aspects often acknowledged, such as business, information, applications and infrastructure (cf. Boar, 1999; Foorthuis & Brinkkemper, 2007; The Open Group, 2009).

Given an applied Project Prescription, each individual check can have one of three outcomes:

- *Passed*: the applied prescription passed the respective compliance check.

- *Failed*: the applied prescription failed the respective compliance check.
- *Needs attention*: the applied prescription might be (or become) compliant. However, it is applied partially or its application is ambiguous (i.e., there is not sufficient information to determine the outcome of the check).

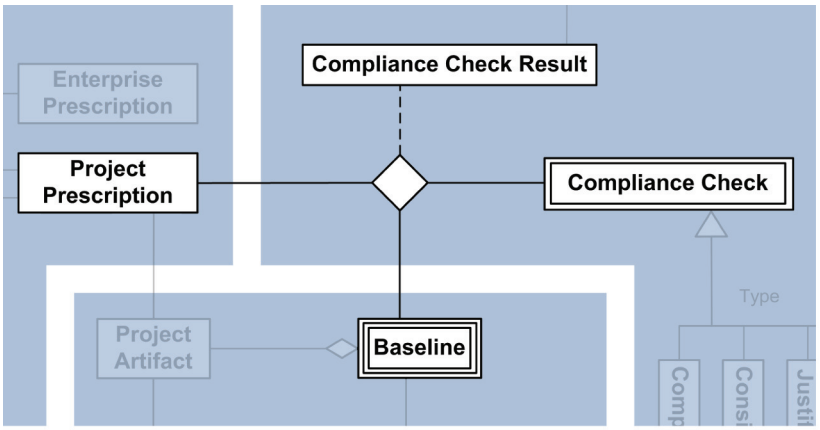
EMPIRICAL EVALUATION

To evaluate and illustrate the EA testing process and its compliance checks, we tested two real-life projects on compliance with actual EA prescriptions. The assessments were carried out at Statistics Netherlands (SN), the Dutch national statistical institute. SN had been developing its architectural practice for several years, since 2006. Conformance to its EA was relevant to projects, since the program responsible for developing it provided them with free IT resources (including an adjunct team of experienced redesign architects cooperating with the project members). Compliance testing was done regularly, albeit often in an informal fashion. Since the original number of prescriptions was considered too extensive, SN had – shortly before our tests were carried out – brought down the number of principles significantly. The project artifacts assessed here were created independently of the researchers and enterprise architects.

To be able to identify the arbitrary aspects of testing, both assessments were carried out independently by the two principal researchers, both working at SN at the time. Contact between the testers occurred only before and after a test (to compare results and clarify ambiguous prescriptions or checks), not during it.

To explain more fully what was required for our compliance assessment, we will refer to Figure 5, which is an excerpt from the EA Compliance Model (Figure 2). The ternary association presents three inputs for the test (the three highlighted classes connected with a continuous line). First, at the bottom of the diagram, the Baseline represents the object(s) being evaluated. In both projects, these *assessment items* consisted of a business analysis and design baseline. Secondly, the Project Prescriptions denote the *compliance norms*, to which the baseline must conform. The prescriptions here took the form of textual principles. Since the assessment items were business analysis and design documents, only the business and information principles were used as norms. At a later stage, the application and technology infrastructure prescriptions would have to provide the norms to test the baselines containing functional IT requirements and software architecture specifications. As a third input,

Figure 5. Elements required for a compliance assessment



the Compliance Check consists of four types of verification, each evaluating a different aspect of conformance. They can be used in every EA compliance assessment. The compliance check types are part of the overall *compliance test*.

In preparation of this assessment, the two testers discussed the principles in detail, which was needed since it was not always explicitly mentioned in the EA why or how they needed to be applied. This resulted in the clarification of these prescriptions' rationale and implication. For the consistency check, it was also determined which set of prescriptions formed a package. The first test was subsequently carried out, yielding various Compliance Check Results. As an example, Table 1 presents one of the tester's reports.

Using a binomial distribution and no empirical data, the expected number of randomly agreed-upon ratings can be calculated as: $E = n \cdot p = 21 \cdot 0.25 = 5.25$ expected identical scores[3]. However, despite the joint preparation, the first test yielded the surprising result that, with only 3 identical scores, there was even less agreement between the two testers than could be expected on the basis of chance alone. In addition, 6 scores showed extreme differences, i.e., "Passed" versus "Failed" values. For further analysis, Cohen's Kappa was calculated, which is a statistical measure for determining the agreement between two raters. It has a value of between -1 and 1, with the former representing perfect disagreement and the latter perfect agreement. Values near

Table 1. The compliance checks results per prescription (for a given baseline)

Prescription		Compliance Check Results			
		Correctness	Justification	Consistency	Completeness
1	The statistical production shall be output-focused and cost aware.	!	!		
2	A rigorous distinction shall be made between a) the actual data that are processed, and b) the metadata describing definitions, quality and process activities.	!	!	!	
3	There shall be no production before relevant metadata is fully and explicitly stated.	!	!		
4	Processes concerning the management function shall be distinguished from all other processes.	✘	✘		
5	When redesigning statistical processes, the benefits of re-use shall be exploited to the full.	✓	✓		✘
6	Re-usable data shall be stored in enterprise-wide steady state data stores belonging to one of four interface levels (i.e., Inputbase, Microbase, Statbase and Outputbase).	!	✓	!	
7	Metadata and (anonymized) data stored in steady state data stores shall be standardized, easily discovered and publicly accessible within SN.	✓	✓		
8	Processing of data shall occur between interface levels, in which data is collected from and stored in the Data Service Center.	✓	✓		
9	Quality versions of steady state data stores shall be identifiable as versions of one and the same data store.	✓	✓		
EA Compliance Judgment: Not passed yet. Especially regarding metadata, important elements are missing.					
Symbols: ✓ Passed ! Needs attention ✘ Failed Not applicable					

zero, associated with non-significant p-values, suggest that the observed (dis)agreement is attributable to chance (SPSS, 2008). See Landis and Koch (1977) for a more fine-grained interpretation of Kappa scores. With the first assessment's Kappa having a value of -0.086 and a p-value of 0.383, we have to conclude that the two testers agreed no more and no less than if they had performed the assessment randomly. Post-assessment discussions revealed that the inter-rater disagreements could be attributed to ambiguity in all three inputs of the ternary association, i.e., the prescriptions, the compliance checks and the business analysis baseline were all being interpreted differently. Although one conclusion was that strict operational definitions were necessary, the four types of compliance checks were deemed useful. No additional compliance check types were required in order to perform the assessment.

Following the first test, improved operationalizations of both the compliance checks and the prescriptions were created. The operational definitions of the (organization-independent) checks resulted in strict rules for the meaning and application of these checks. They are all included in Appendix A, as they are re-usable in other organizations. The operationalization of the (organization-dependent) prescriptions, which should be seen as separate from creating their rationale and implication, resulted in stricter and more detailed operational definitions. An example is provided in Appendix A. The second test consequently resulted in a significant increase of agreement, with 14 identical scores, no extreme differences, a Kappa value of 0.520 and a p-value of < 0.0005. Although statistically significant and thus not attributable to chance, this value for inter-rater reliability still represents only "moderate" agreement (Landis & Koch, 1977). While discussing the results, it became clear that the deviating scores could still be attributed to the remaining subjectivity of the prescriptions and business analysis artifact, but no longer to a different interpretation of the compliance checks. The conclusions were discussed with the authors of the baselines and feedback remarks were e-mailed to the lead architect.

Discussion of Research Results

Our research sheds light on the aspects of compliance testing that are specific to Enterprise Architecture. The results indicate that assessing compliance with EA is inherently subjective and interpretive in nature, similar to judicial decisions and academic peer reviews (which often show inconsistent outcomes). There are several reasons for this. First, EA prescriptions often prove to be inherently *abstract*, which is a consequence of their strategic nature and of them aiming at a partially unknown future. This renders prescriptions open to interpretation. Creatively interpreting and translating EA prescriptions to fit them to the specific situation is inherent in working with EA. Secondly, since EA prescriptions and project artifacts have to be read and applied by human actors (analysts, testers, programmers, managers and other stakeholders), *natural language* is the most appropriate format. Natural language, however, is always open to interpretation. Thirdly, when discussing the tests we discovered that we (subconsciously) had used not only the information provided by the artifacts and the EA, but also *personal and contextual knowledge*, e.g., previous experiences with the domain in question which helped give understanding and meaning to the assessed baseline. In short, testing requires sense-making, intuition, experience and knowledge of the business context. Assessments cannot therefore be expected to result in total agreement between human testers (i.e., a Kappa value of 1.0). Take, for example, principle 5, "When redesigning statistical processes, the benefits of re-use shall be exploited to the full." Assessing this rather abstract principle not only requires knowledge of existing and potentially re-usable statistical data (inside SN) and IT systems (both inside and outside SN), but also of the goals and requirements of the project in question in order to make a match between potentially re-usable resources and project needs.

There are indications that the factors causing subjectivity in EA compliance testing are not solely present in the organization in which we did our empirical research. Take for example almost any of TOGAF's example set of architecture principles (The Open Group, 2009) to see the above-mentioned abstract and vague nature (e.g., "Data is an asset that has value to the enterprise and is managed accordingly"). In addition, research on EA has regularly found EA prescriptions to be ambiguous (Lindström, 2006; Op 't Land & Proper, 2007; Foorthuis & Brinkkemper, 2008). This is consistent with research in other fields, since, throughout the years, consensus studies have often demonstrated low or moderate agreement between auditors (e.g., Joyce, 1976; Srinidhi & Vasarhelyi, 1986; Amer et al., 1994; Lin et al., 2003). In addition, when considering law and international treaties, the legal rules therein often prove to be ambiguous, and thus call for subtle and subjective compliance evaluation (Chayes & Chayes, 1993; Zaelke et al., 2005). The results of our study, including the second test with moderate agreement is consistent with these findings.

Given the above, a Kappa value of 0.520, representing "moderate" agreement, is a satisfactory result, especially for a foundational study. However, this does not imply that research should not strive to improve the inter-tester agreement. Given the inherent subjectivity, research could perhaps aim at achieving Kappa values of between 0.61 and 0.80, i.e., "substantial" (Landis & Koch, 1977). What can be done to mitigate the effects of the subjective nature of EA compliance testing? First, our results suggest that prescriptions need to be as operationalized as possible, similar to rendering concepts in social science research measurable. This makes the testing of prescriptions less prone to individual interpretation. The pseudo-formalizations can be inspired by real-life situations, limiting operationalizations to relevant issues. In theory this needs to be done only once, but after an given assessment the pseudo-formalizations may need to be improved as a result of the new testing experience. The operationalizations and examples of their application can then be part of an in-house training on EA compliance testing aimed at improving the assessment process. However, note that in a real-life, non-academic setting operationalizing has its limits, since too many rules will likely result in testers not reading or remembering them. Furthermore, operational definitions may deal only with a limited set of well-known situations, and may be of less use to new and unknown situations. A second way to deal with the interpretive nature of EA compliance testing is assessing (important) projects by two testers and have their joint EA Compliance Report reviewed by the lead enterprise architect. The result should be increased consensus between testers (Trotman & Yetton, 1985; Joyce, 1976). This is therefore not only recommended to decrease the subjectivity of the assessment, but also to boost acceptance of its results by the project members (who will undoubtedly be aware of the interpretive nature of the prescriptions, since they have applied them). Finally, the EA Conformance Report itself should be reviewed and discussed with the authors of the baseline, in order to prevent erroneous check results and judgments. This empirically induced insight is the reason why this step has been added in the "Review Baseline" process model of Figure 4.

The results of our study also have ramifications for automated compliance testing. This is a popular topic in many publications on compliance with standards and legislation (cf. Emmerich et al., 1997, 1999; El Kharbili et al., 2008; Sadiq et al., 2007; Chung et al., 2008). Indeed, it is feasible to perform all kinds of checks on documents, models and datasets. Especially when the mere existence of properties can be objectively measured, e.g., compliance with the standard "each user requirement includes a measure of priority" (cf. Emmerich et al., 1999; Chung et al., 2008). However, our research leads us to suspect that an EA is less suitable for automated compliance testing, as the above-mentioned characteristics of EA prescriptions severely hinder automated checking. Prescriptions are written in natural language, they are often inherently abstract and have been

translated to a local situation. Furthermore, testing them often requires knowledge that is out-of-scope for machines, for example domain knowledge or information about the non-automated or non-modeled business or its context. Formalizing this might prove impossible or not worth the effort. Arguably, an inference engine capable of testing prescriptions with these characteristics is also sufficiently powerful to carry out the project itself. Tests that could be done automatically are likely to yield irrelevant and non-substantive outcomes. For the time being, knowledgeable human actors are key in this type of compliance assessment task, as they are capable of identifying and resolving interpretational differences.

We therefore consider it likely that tools (at least in the short-term) will not be able to meaningfully test a substantial part of business processes and IT systems on EA conformance automatically. However, there are definitely areas in compliance testing that could be supported by tools. For example, the operationalization of the compliance checks (Appendix A) defines strict constraints for the checks' values. These 'meta checks' can be carried out by a tool for recording the values. Furthermore, tools could provide valuable assistance for registering compliance issues. Structured recording would allow for automated calculation of 'compliance scores' of projects and departments, and for post-assessment analyses (e.g., identifying which prescriptions are the most important sources of non-compliance or which departments have a relatively low 'compliance score').

Another discussion altogether is the question of whether our proposed method is suitable for routine application. Since design science is concerned with innovations, this is both a relevant and a difficult question. However, some remarks can still be made, especially concerning the question of whether all four types of checks should always be performed and reported for each prescription (set). If an EA contains many prescriptions, then this can yield a large number of compliance check results. It may therefore be practical and more efficient to regard the checks as aspects to be kept in mind, and only report on an aspect if it has compliance issues. It might also be possible to perform the correctness and justification checks at the same aggregated level as the consistency check, thereby allowing for a more superficial test when time is an issue. However, when an organization is in the process of starting up its EA compliance assessment function, we advise to conduct detailed and full compliance assessments and to involve multiple testers in each of these assessments. This allows for the testers to develop a shared understanding regarding the prescriptions and checks, and collaboratively create the necessary operationalizations. When compliance testing is becoming routine and testers have received training, the above mentioned partial reporting and aggregated checks can be carried out, allowing for a more efficient process. This will make the assessment less precise and more vulnerable to subjectivity. However, since reporting is less detailed, the probability of not detecting disagreement also increases (i.e., using less detailed categories decreases the probability of observing different scores).

CONCLUSION

We set out to explore how projects, and the business and IT solutions they deliver, can be assessed on compliance with EA. Our research has yielded the following contributions.

- The artifact *EA Compliance Model* (Figure 2 and supporting text), which identifies the core elements of compliance testing in the context of Enterprise Architecture. In design science terms, this artifact can be categorized as a model. However, the model communicates a world view and is not empirically evaluated here.
- The artifact *Process model* (Figure 4 and supporting text). This offers process steps and detailed prescriptions for carrying out several EA-related checks. In design science terms, this artifact can be categorized as a method, taking several parts of the EA compliance model as input. The set

of *Compliance Checks* (section "Types of Compliance Checks" and Appendix A) used in the process model may in design science terms be seen as a construct. The process model – including the compliance checks – has been empirically evaluated and illustrated in this article.

- The *insights* that resulted from the empirical evaluation with real-life EA prescriptions and project documents. One insight is that our approach can be used to assess real-life projects, albeit the inter-rater agreement is still only moderate. This is related to another insight, that EA compliance testing is inherently subjective and interpretive by nature, due to EA prescriptions being strategic and abstract, the (justified) use of natural language, and the inevitable use of personal and contextual knowledge. This is similar to the inconsistent outcomes in, e.g., judicial decisions, academic peer reviews and audits. We therefore do not consider it realistic to expect much from formalized, objective and automated assessments, especially not in the short term. We expect more from operationalizing norms for human-based compliance tests, bearing in mind that perfectly objective tests will not be within reach. In design science terms, the empirical endeavor can be seen as an instantiation, yielding insights in EA compliance testing.

We have several suggestions for further research. First, as our empirical research employed architecture principles, another topic for further investigation is studying whether our conclusions for principles also hold true for models and policy statements. In fact, it can be expected that the compliance checks are not only useful for assessing the application of EA prescriptions, but also of other norms, such as legal rules and industrial standards. In addition, they may not only function as checks performed by testers after application, but also as aspects to take into account by implementers when in the process of conforming to or applying the norms. As a second suggestion, future research can study what kind of tool support is most valuable. Although we do not expect much from automated compliance testing, we have presented several options for tools supporting compliance assessments. A third topic that deserves further attention is how to arrive at optimal operationalizations for human-based compliance assessments. The operationalizations presented in Appendix A can also be subjected to scrutiny. A fourth topic would be to investigate the role of tacit knowledge in testing, which could focus on developing shared implicit meanings regarding prescriptions, rather than on explicit operational definitions. Whatever the topics of future studies, our research clearly shows that minimizing the subjectivity of assessments is something that has to be pursued actively, as objective compliance testing cannot be taken for granted.

ACKNOWLEDGMENTS

The authors wish to thank Abby Israëls, Lieneke Hoeksma, Marlies van Steenbergen, Wiel Bruls, Tjalling Gelsema, Robbert Renssen, Peter van Nederpelt, Marta Indulska, Shazia Sadiq, Michael zur Muehlen and the anonymous GRCIS and JDM reviewers for their valuable comments. A preliminary version of this article was presented at the GRCIS workshop of the CAISE 2009 conference in Amsterdam, the Netherlands.

REFERENCES

Amer, T., Hackenbrack, K., & Nelson, M. (1994). Between-auditor differences in the interpretation of probability phrases. *Auditing: A Journal of Practice and Theory, 13*(1), 126-136.

Baresi, L., & Pezzè, M. (2006). An introduction to software testing. *Electronic Notes in Theoretical Computer Science*, *148*(1), 89–111. doi:10.1016/j.entcs.2005.12.014

Barr, M. S., & Miller, G. P. (2006). Global Administrative Law: The view from Basel. *European Journal of International Law*, *17*(1), 15–46. doi:10.1093/ejil/chi167

Binder, R. V. (2000). *Testing object-oriented systems: Models, patterns, and tools*. Reading, MA: Addison-Wesley.

Boar, B. H. (1999). *Constructing blueprints for enterprise IT architectures*. New York, NY: John Wiley & Sons.

Boh, W. F., & Yellin, D. (2007). Using enterprise architecture standards in managing information technology. *Journal of Management Information Systems*, *23*(3), 163–207. doi:10.2753/MIS0742-1222230307

Bruls, W. A. G., van Steenbergen, M., Foorthuis, R. M., Bos, R., & Brinkkemper, S. (2010). Domain architectures as an instrument to refine enterprise architecture. *Communications of the Association for Information Systems*, *27*(1), 27.

Burditt, G. M. (1996). Corporate compliance audits. *Food and Drug Law Journal*, *51*, 217–220.

Caballero, I., Verbo, E., Calero, C., & Piattini, M. (2007). A data quality measurement information model based on ISO/IEC 15939. In *Proceedings of the 12th International Conference on Information Quality*, Cambridge, MA.

Chayes, A., & Chayes, A. H. (1993). On compliance. *International Organization*, *47*(2), 75–205. doi:10.1017/S0020818300027910

Chung, P. W. H., Cheung, L. Y. C., & Machin, C. H. C. (2008). Compliance flow: Managing the compliance of dynamic and complex processes. *Knowledge-Based Systems*, *21*(4), 332–354. doi:10.1016/j.knosys.2007.11.002

Drew, M. (2007). Information risk management and compliance: Expect the unexpected. *BT Technology Journal*, *25*(1), 19–29. doi:10.1007/s10550-007-0004-x

El Kharbili, M., Stein, S., Markovic, I., & Pulvermüller, E. (2008). Towards a framework for semantic business process compliance management. In *Proceedings of the GRCIS Caise Workshop on Governance, Risk and Compliance of Information Systems*.

Emmerich, W., Finkelstein, A., Montangero, C., Antonelli, S., Armitage, S., & Stevens, R. (1999). Managing standards compliance. *IEEE Transactions on Software Engineering*, *25*(6), 836–851. doi:10.1109/32.824413

Emmerich, W., Finkelstein, A., Montangero, C., & Stevens, R. (1997). Standards compliant software development. In *Proceedings of the ICSE Workshop on Living with Inconsistency*.

Farenhorst, R., & De Boer, R. C. (2009). *Architectural knowledge management: Supporting architects and auditors* (Unpublished doctoral dissertation). Vrije Universiteit Amsterdam, Amsterdam, The Netherlands.

Foorthuis, R. M., & Brinkkemper, S. (2007). A framework for local project architecture in the context of enterprise architecture. *Journal of Enterprise Architecture*, *3*(4), 51–63.

Foorthuis, R. M., & Brinkkemper, S. (2008). Best practices for business and systems analysis in projects conforming to enterprise architecture. *Enterprise Modelling and Information Systems Architectures*, *3*(1), 36–47.

Foorthuis, R. M., Brinkkemper, S., & Bos, R. (2008). An artifact model for projects conforming to enterprise architecture. In J. Stirna & A. Persson (Eds.), *Proceedings of the IFIP WG 8.1 Working Conference on Practice of Enterprise Modeling* (LNBIP 15, pp. 30-46).

Foorthuis, R. M., van Steenbergen, M., Mushkudiani, N., Bruls, W., Brinkkemper, S., & Bos, R. (2010). On course, but not there yet: Enterprise architecture conformance and benefits in systems development. In *Proceedings of the Thirty First International Conference on Information Systems*, St. Louis, MO.

Gammelgård, M., Simonsson, M., & Lindström, Å. (2007). An IT management assessment framework: evaluating enterprise architecture scenarios. *Information Systems and e-Business Management, 5*(4), 415-435.

Gauthier, D. (1991). Why contractarianism? In Vallentyne, P. (Ed.), *Contractarianism and rational choice: Essays on David Gauthier's Morals by Agreement* (pp. 15–30). Cambridge, UK: Cambridge University Press.

Hamilton, R. A. (1995). Compliance auditing. *Internal Auditor*, *52*(6), 42–45.

Hartman, E. M. (1996). *Organizational ethics and the good life*. New York, NY: Oxford University Press.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly*, *28*(1), 75–105.

Hoogervorst, J. A. P., & Dietz, J. L. G. (2008). Enterprise architecture in enterprise engineering. *Enterprise Modelling and Information Systems Architectures*, *3*(1), 3–13.

IEEE. (1990). *IEEE St. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology*. Washington, DC: Author.

IFAC. (2003). *International Framework for Assurance Engagements, Exposure Draft*. New York, NY: Author.

Johnson, P., Ekstedt, M., Silva, E., & Plazaola, L. (2004). Using enterprise architecture for CIO decision-making: On the importance of theory. In *Proceedings of the Second Annual Conference on Systems Engineering Research*.

Jonkers, H., Lankhorst, M. M., ter Doest, H. W. L., Arbab, F., Bosma, H., & Wieringa, R. J. (2006). Enterprise architecture: Management tool and blueprint for the organisation. *Information Systems Frontiers*, *8*(2), 63–66. doi:10.1007/s10796-006-7970-2

Joyce, E. J. (1974). Expert judgment in audit program planning. *Journal of Accounting Research*, *14*, 29–60. doi:10.2307/2490445

Kaisler, S. H., Armour, F., & Valivullah, M. (2005). Enterprise architecting: Critical problems. In *Proceedings of the 38th Hawaii International Conference on System Sciences*.

Kim, S. (2007). IT compliance of industrial information systems: Technology management and industrial engineering perspective. *Journal of Systems and Software*, *80*(10), 1590–1593. doi:10.1016/j.jss.2007.01.016

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. doi:10.2307/2529310

Lankhorst, M. (2005). *Enterprise architecture at work. Modelling, communication and analysis*. Berlin, Germany: Springer-Verlag.

Lin, K. Z., Fraser, A. M., & Hatherly, D. J. (2003). Auditor analytical review judgement: A performance evaluation. *The British Accounting Review*, *35*(1), 19–34. doi:10.1016/S0890-8389(02)00107-5

Lindström, Å. (2006). On the syntax and semantics of architectural principles. In *Proceedings of the 39th Hawaii International Conference on System Sciences*.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. doi:10.1016/0167-9236(94)00041-2

Massacci, F., Prest, M., & Zannone, N. (2005). Using a security requirements engineering methodology in practice: The compliance with the Italian data protection legislation. *Computer Standards & Interfaces*, *27*(5), 445–455. doi:10.1016/j.csi.2005.01.003

Meschke, M., & Baumoel, U. (2010). Architecture concepts for value networks in the service industry. In *Proceedings of the Thirty First International Conference on Information Systems*, St. Louis, MO.

Nouwt, S. (2008). Reasonable expectations of geoprivacy? *SCRIPTed*, *5*(2), 375–403. doi:10.2966/scrip.050208.375

Op 't Land, M., & Proper, H. A. (2007). Impact of principles on enterprise engineering. In *Proceedings of the 15th European Conference on Information Systems*.

Pfleeger, S. L., Fenton, N., & Page, S. (1994). Evaluating software engineering standards. *IEEE Computer*, *27*(9), 71–79. doi:10.1109/2.312041

Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM*, *45*(4), 211–218. doi:10.1145/505248.506010

Richardson, G. L., Jackson, B. M., & Dickson, G. W. (1990). A principles-based enterprise architecture: Lessons from Texaco and Star Enterprise. *Management Information Systems Quarterly*, *14*(4), 385–403. doi:10.2307/249787

Riempp, G., & Gieffers-Ankel, S. (2007). Application portfolio management: A decision-oriented view of enterprise architecture. *Information Systems and e-Business Management, 5*(4), 359-378.

Ross, J. W., Weill, P., & Robertson, D. (2006). *Enterprise architecture as strategy: Creating a foundation for business execution*. Boston, MA: Harvard Business School Press.

Sadiq, S., Governatori, G., & Naimiri, K. (2007). Modeling control objectives for business process compliance. In *Proceedings of the 5th International Conference on Business Process Management*.

SPSS. (2008). *SPSS 16.0 Help File*. Chicago, IL: Author.

Srinidhi, B. N., & Vasarhelyi, M. A. (1986). Auditor judgment concerning establishment of substantive tests based on Internal control reliability. *Auditing: A Journal of Practice & Theory, 5*(2).

Tewarie, W. N. B. (2010). *Model based development of audit terms of reference: A structured approach to IT auditing* (Unpublished doctoral dissertation). Vrije Universiteit Amsterdam, Amsterdam, The Netherlands.

The Open Group. (2009). *TOGAF Version 9: The Open Group Architecture Framework*. Reading, UK: Author.

Trotman, K. T., & Yetton, P. W. (1985). The effect of the review process on auditor judgments. *Journal of Accounting Research*, *23*(1), 256–267. doi:10.2307/2490918

van Bommel, P., Buitenhuis, P. G., Hoppenbrouwers, S. J. B. A., & Proper, H. A. (2007). Architecture principles: A regulative perspective on enterprise architecture. In *Proceedings of the Second Workshop on Enterprise Modelling and Information Systems Architectures* (Vol. 119, pp. 47-60).

van de Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In Syed, M. R., & Syed, S. N. (Eds.), *Handbook of research on modern systems analysis and design technologies and applications* (pp. 38–58). Hershey, PA: Idea Group. doi:10.4018/978-1-59904-887-1.ch003

van Zeist, R. H. J., & Hendriks, P. R. H. (1996). Specifying software quality with the extended ISO model. *Software Quality Journal*, *5*(4), 273–284. doi:10.1007/BF00209185

von Solms, S. H. (2005). Information security governance: Compliance management vs. operational management. *Computers & Security*, *24*, 443–447. doi:10.1016/j.cose.2005.07.003

Vroom, C., & von Solms, R. (2004). Towards information security behavioural compliance. *Computers & Security*, *23*, 191–198. doi:10.1016/j.cose.2004.01.012

Zaelke, D., Kaniaru, D., & Kružíková, E. (2005). *Making law work: Environmental compliance & sustainable development* (*Vol. 1-2*). London, UK: Cameron May.

zur Muehlen, M., Indulska, M., & Kamp, G. (2007). Business process and business rule modeling languages for compliance management: A representational analysis. In *Proceedings of the 26th International Conference on Conceptual Modeling*, Auckland, New Zealand.

ENDNOTES

1. For didactic purposes, we used overriding of a property here (i.e., *Definition*), which is unusual in OO.
2. Note that the EA Compliance Model depicted in Figure 2 has been simplified for didactic purposes, as it can only contain the checks at the level of an individual Prescription. To model the consistency and completeness checks technically correct, a Project Prescription Group, containing one or more Project Prescriptions, should be added between the Project Prescription and the ternary association.
3. We are not interested in the two raters both having a specific outcome (e.g., "passed"), but simply in them having an identical outcome. Therefore, one of the ratings should be seen as given, rendering its value irrelevant to the calculation of the expected number of identical scores. The probability of two raters having the same outcome is thus 0.25. This should be multiplied by the number of cells (i.e., 21).
4. Not all EA prescriptions are mandatory in practice, as some EAs also contain, e.g., recommended best practices. This is reflected by the property *Priority* in the EA Compliance Model of Figure 2. A mandatory prescription is also not necessarily relevant. This is due to its priority being determined at the Enterprise Architecture level, while relevancy is determined at a later stage at the application (project) level. In practice, general prescriptions may prove to be irrelevant in specific situations. Note that what exactly is "relevant" is determined here by the tester. It should also be noted that the authors had a discussion about whether priority should be included in the operationalization.

Ralph Foorthuis is currently a Senior IT Architect at UWV Business Services. He has worked at several organizations as a business and IT architect, systems analyst, database designer and developer. He has studied at the University of Amsterdam, where he obtained his masters in Informatics and Communication Science (both cum laude). As part of his PhD research he has published multiple articles. His research interests include enterprise and project architecture, compliance, business and systems analysis, and the value of architecture.

Frank Hofman is working as a Business Architect at Statistics Netherlands (CBS). Before this position he was a consultant at Atos Origin working at several customers including Shell, ING Bank, Royal Dutch Airlines (KLM) and Dutch Railways (NS). He holds a MSc in Information Systems Development at the university of Hertfordshere in collaboration with polytechnic of Arnhem and Nijmegen, and a BBA (Ing) in Business Administration of polytechnic of Rijswijk.

Sjaak Brinkkemper is Full Professor of Organization and Information at the Department of Information and Computing Sciences of Utrecht University, the Netherlands. He leads a group of about thirty researchers specialized in product software entrepreneurship. The main research themes of the group are methodology of product software development, implementation and adoption, and entrepreneurship in the product software industry.

Rik Bos is Assistant Professor at the Department of Information and Computing Sciences, Faculty of Science, Utrecht University. He obtained a PhD in Mathematics in 1984, also from Utrecht University and then switched to Computing Sciences at the Technical Unversity of Eindhoven. He has worked for fifteen years in several software companies in the field of software and architecture. His research interests include enterprise architecture, software modelling and patterns.

APPENDIX A

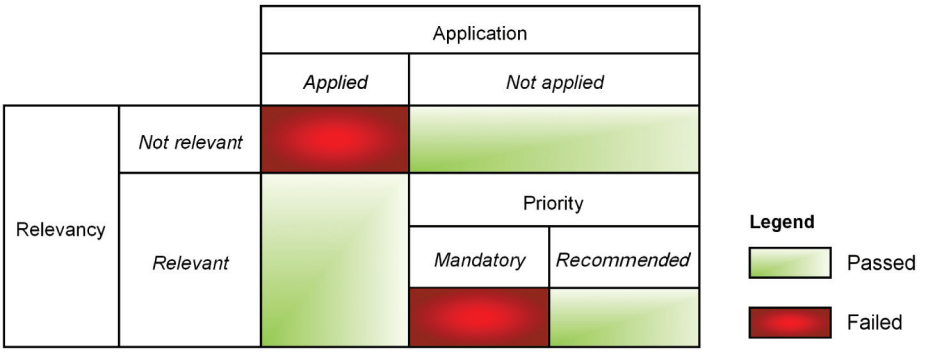
Operational Definitions

1. Operationalization of the Compliance Checks

This sub-section presents the operationalization of the compliance checks. Note that these are organization-independent and can thus be re-used in other settings.

1. The three values of the checks are ordinal by nature. From low to high, the order is "Failed", "Needs attention" and "Passed". The "Not applicable" value, in principle assigned up-front, is not considered an intrinsic part of this order.
2. The assessment is limited to testing the desired or future situation – be it short, medium or long term – since the objective is to test the compliance of the (design of the) new business and/or IT system that is to be delivered. The current situation is therefore not assessed when testing a project on conformance.
3. If a prescription is relevant (regardless of whether it is mandatory) and has indeed been applied (regardless of whether it has been applied correctly), the Justification Check results in "Passed". If a prescription is relevant (and mandatory) but has not been applied, the Justification Check results in "Failed".[4] If a prescription has been applied while it is not relevant in the specific local situation (regardless of whether it is mandatory), the Justification Check again results in "Failed". If a prescription has not been applied in a situation in which it is not relevant (regardless of whether it is mandatory), the Justification Check results in "Passed". See statement 4 for more information about the values of the Justification Check. Figure 6 summarizes this operationalization visually.
4. The instruction in statement 3 focuses on situations in which a strict distinction can be made between "Passed" and "Failed". However, as grey areas may exist, the meaning of the values for the Justification Check will be described in more detail below. Given a prescription:
   - The "Passed" value indicates that:
     - The project has applied (all the mandatory elements of) the prescription, regardless of whether this has been done correctly or not.

Figure 6. Operationalization table(a)



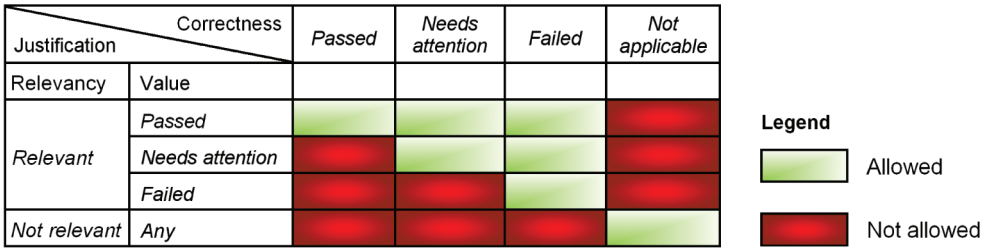
        - The project has not applied (all the elements of) the prescription, either because of non-relevance or because the project has taken the freedom that is inherent in the recommended nature of the prescription.
    - The "Needs attention" value indicates:
        - Partial conformance: the project has applied the prescription partially (e.g., only one or several of the mandatory elements, or one mandatory element only to a certain degree), regardless of whether this has been done correctly or not.
        - Insufficient information: there are indications that the project has applied the prescription (e.g., because it is claimed or implied in the Baseline), regardless whether this has been done correctly or not. However, it is not possible to test this on compliance (e.g., because references have been made or implied to additional documents, which are not included in the tested Baseline and are therefore not available for assessment).
    - The "Failed" value indicates that:
        - No information whatsoever is available about the application of the prescription, i.e., the prescription seems to have been totally ignored.
        - The project has stated that this (relevant) prescription is not considered relevant.

5. The value of the Correctness Check is dependent on the value of the Justification Check for the prescription in question. The value of the Correctness Check cannot be higher than that of the Justification Check. For the Correctness Check, no distinction is made between mandatory and recommended prescription elements; all elements are considered equal. In other words, if a prescription has been applied (regardless of whether it is mandatory), it should be applied correctly. Below, the value of the Correctness Check is discussed in relation to the Justification Check.
    - If the value of the Justification Check is "Passed" because the prescription is relevant and has been applied, the value of the Correctness Check can result in "Passed", "Needs attention" or "Failed". A value of "Not applicable" is not allowed.
    - If the value of the Justification Check is "Needs attention" because the prescription is relevant, but has been applied partially or there is insufficient information to test it, the value of the Correctness Check can only result in either "Failed" (if all elements are "Failed") or "Needs attention" (e.g., if one element is "Passed", one is "Failed" and one is "Needs attention"). The values "Passed" and "Not applicable" are not allowed.
    - If the value of the Justification Check is "Failed" because the prescription in question has not been applied and it was relevant to do so, the value of the Correctness Check per definition also results in "Failed".
    - If the value of the Justification Check is "Failed", "Passed", "Needs attention" or "Not applicable" and the prescription in question is not relevant, the value of the Correctness Check per definition results in "Not applicable" (regardless of whether the prescription in question has been applied correctly or not).

    Summing up, Figure 7 shows the combinations that are allowed and not allowed.

6. The value of the Consistency Check results in "Failed" if specific inconsistencies or off-balances can be found or expected. Therefore, the value does not automatically result in "Failed" if one or more of the underlying Correctness or Justification Checks is "Failed" (as this would in essence simply be equivalent to a Completeness Check on a subset of the prescriptions). However, one or several "Needs attention" values for underlying Correct-

Figure 7. Operationalization table(b)



ness or Justification Checks do automatically result in a value "Needs attention" for the Consistency Check, since it cannot be known whether consistency is maintained.

7. The Completeness Check only results in a "Passed" value if all prescriptions have a "Passed" value for the Justification Check. The Completeness Check assesses whether all relevant mandatory prescriptions have been taken into account, regardless of whether their application is correct. Therefore, the results of the Correctness Check and Consistency Check are not relevant here; these will be taken into account in the final judgment.
8. The final EA Compliance Judgment takes all of the compliance check results in account. A "Passed" value for this judgment indicates complete conformance, and thus a "Passed" value for all underlying checks.

2. Clarification and Operationalization of Prescriptions

This sub-section presents an example of the clarification of the prescriptions' rationale and implication that preceded the first test. In addition, the operationalization of the respective prescription, which was created between the first and second test, is also included. These operational definitions can, for example, prescribe which compliance check values to assign in which situation. Note that both the clarification and operationalization are organization-dependent.

Table 1A. Visual summarization of operationalization

Statement	8. Processing of data shall occur between interface levels, in which data is collected from and stored in the Data Service Center.
Rationale & implication	The interface levels are the Inputbase, Microbase, Statbase and Outputbase, in which steady state datasets are stored. A statistical process uses data from such a store as input and stores them after processing (e.g., enriching or aggregating) in a higher-level data store in the enterprise-wide Data Service Center. The *rationale* is that this stimulates re-use of data, as these stores are available throughout Statistics Netherlands. The *implication* is that the data stored in these enterprise-wide interface levels need to be relatively stable and of high quality (i.e., there should be no need to correct the data in the immediate future).
Operationalization	Datasets (i.e., statistical products or steady state data stores) should be related to interface levels. This means that each dataset should be explicitly linked to either the Inputbase, Microbase, Statbase or Outputbase (and possibly also the Pre-Inputbase and/or Post-Outputbase). If it is only mentioned that the Data Service Center is or will be used, the values of the Justification and Correctness Checks should be "Needs attention". If the Data Service Center is not mentioned at all, the values of the Justification and Correctness Checks should be "Failed". This prescription is not part of a package (i.e., it is not related to other prescriptions for the Consistency Check).

APPENDIX B

Quality Aspects

Quality aspects of data, software and auditing have served as theoretical support for the compliance checks and process steps. This Appendix lists the quality aspects and the rationale for (not) adapting them to our approach.

1. Software

The Quint model features several quality dimensions for software (van Zeist & Hendriks, 1996).

Functionality

- *Suitability*: dependent on one's interpretation, this might be seen as the justification check.
- *Accuracy*: this is the correctness check. However, "accurate information, measurements, and statistics are correct to a very detailed level" (Collins Cobuild Dictionary). Since EA checks are not detailed, due to the strategic and abstract nature of the prescriptions, we prefer the term "correctness".
- *Interoperability*: irrelevant for EA compliance checks. This is a quality aspect specifically for systems (although an EA prescription could be about interoperability, so it could be subjected to a check).
- *Security*: irrelevant.
- *Compliance*: irrelevant. If, e.g., "compliance to law" is an EA principle, then it will be tested as part of checking that specific prescription.
- *Traceability*: irrelevant as a separate check. However, it should be clear how the EA prescription is applied. If this is not the case, then a compliance check can yield the outcome "Needs attention".

Reliability

- *Maturity*: irrelevant.
- *Fault tolerance*: irrelevant.
- *Recoverability*: irrelevant.
- *Availability*: irrelevant, but might make for a good quality aspect for EA prescriptions.
- *Degradability*: irrelevant.

Usability

- *Understandability*: an EA prescription itself may be understandable or not, but the conformance check is on its application. It thus does not lead to a separate check. However, it is a relevant issue, since the application of a prescription should be understandable for it to be checked. This is why there is an outcome "Needs attention" (which amongst others can mean that the application is ambiguous).
- *Learnability*: like Understandability, this might be a good quality aspect of a prescription. However, it is irrelevant for its application.
- *Operability*: irrelevant (see Learnability).
- *Explicitness*: irrelevant. EA prescriptions can be applied implicitly or explicitly in project artifacts.

- *Customizability*: irrelevant.
- *Attractivity*: irrelevant.
- *Clarity*: regardless of whether EA prescriptions are applied explicitly or implicitly, it should be clear how they are applied. If this is not the case then a compliance check can yield the outcome "Needs attention".
- Helpfulness: irrelevant.
- *User-friendliness*: irrelevant.

Efficiency
- *Time behavior*: irrelevant.
- *Resource utilization*: this aspect is not entirely irrelevant for EA compliance assessments. If the application of an EA prescription costs more than is gained (from the perspective of the entire enterprise), there is no good reason to apply it. This is covered in the justification check.

Portability
- *Adaptability*: irrelevant. It might be a good quality aspect for an EA prescription: due to its strategic nature, a prescription needs to be translated (adapted) to the specific situation in which it is applied.
- *Installability*: irrelevant.
- *Conformance*: irrelevant. Assessing conformance is the whole point here, and it is tested on several aspects.
- *Replaceability*: irrelevant.

Maintainability
- *Analyzability*: irrelevant.
- *Changeability*: irrelevant.
- *Stability*: irrelevant. It might be a good quality aspect for an EA prescription.
- *Testability*: irrelevant. From the perspective of this research, this obviously is a good quality aspect for an EA prescription.
- *Manageability*: irrelevant as an explicit check. However, applying the prescriptions should be realistically possible, which can be verified as part of the justification check.
- *Reusability*: irrelevant.

2. Data

Several quality dimensions for data can be acknowledged (Pipino et al., 2002; Caballero et al., 2007).

- *Accessibility*: irrelevant.
- *Appropriate amount*: verified with the completeness check. A related issue is how detailed and comprehensive a compliance assessment and its reporting should be.
- *Believability*: irrelevant.
- *Completeness*: this is the completeness check.
- *Conciseness*: irrelevant as an explicit check. However, see Appropriate amount.
- *Consistency*: this is the consistency check.
- *Customer support*: irrelevant as an explicit check. However, in this regard it should be noted that both the testers and the project members should be available to explain their choices.

- *Documentation*: irrelevant as an explicit check. However, the judgments made by the testers should be documented in an EA Conformance Report.
- *Ease of manipulation*: irrelevant.
- *Free-of-error*: this is the correctness check.
- *Interpretability*: irrelevant as an explicit check. However, if the project artifacts feature, e.g., definitions that are not clear or diagrams with ambiguous symbols, a "Needs attention" value is assigned.
- *Objectivity*: irrelevant as an explicit check, although testers should indeed do their work objectively.
- *Price*: irrelevant as an explicit check. However, the involved stakeholders should agree on how much capacity will be put into the test.
- *Relevancy*: this is covered by the justification check, since the prescriptions applied should be relevant for the situation at hand.
- *Reliability*: irrelevant as an explicit check. However, testers should be able repeat their work or show inter-rater reliability (which, as we have seen, cannot be taken for granted).
- *Reputation*: irrelevant.
- *Security*: irrelevant as an explicit check. However, it should be considered whether the EA Conformance Report should be openly published within the organization.
- *Timeliness*: irrelevant. However, the involved stakeholders should agree on a time planning.
- *Understandability*: irrelevant as an explicit check. However, if the project artifacts are not sufficiently understandable, a "Needs attention" value is assigned.
- *Verifiability*: irrelevant as an explicit check. However, the project artifacts should be verifiable.
- *Value-added*: this is the justification check, since conforming should deliver value in the project situation.

3. Auditing

Principles and quality aspects of auditing (cf. IFAC, 2003; Tewarie, 2010) are also potentially relevant. However, these tend to focus on the quality of the prescriptions, audit process and auditors, leading to aspects such as *Integrity*, *Objectivity* and *Neutrality*. Although desirable, these aspects do not provide suitable templates for compliance checks. Other aspects in this context are already covered, such as *Completeness* and *Consistency*.