CE-SIB: A Modelling Method Plug-in for Managing Standards in Enterprise Architectures

Christoph Moser[1(✉)], Robert Andrei Buchmann[2], Wilfrid Utz[1], and Dimitris Karagiannis[1]

[1] Knowledge Engineering Research Group, University of Vienna, Vienna, Austria
{cmoser, wilfrid, dk}@dke.univie.ac.at

[2] Business Informatics Research Center, Babeş-Bolyai University, Cluj-Napoca, Romania
robert.buchmann@econ.ubbcluj.ro

Abstract. In Enterprise Architecture (EA) Management, adoption of standards brings essential benefits pertaining to compatibility and repeatability but also raises governance challenges. EA frameworks recommend placing architecture artifacts under strict governance to control technological diversity towards reduced costs of operation or business-IT alignment; however, they do not provide methodological guidance on how to support decision-making for standards management. Business process management, model-driven software engineering or IT service management do address such challenges, but fall short in covering all relevant architectural layers. Driven by industry experience, this paper proposes a modelling method plug-in ("function block") to support a model-based integration of practices for standards compliance management and their relevant model bases. It also aims for generality, as the proposal is pluggable through "semantic docking points" to arbitrary EA frameworks. A prototypical implementation in the form of a modelling tool is discussed as an expository instantiation, as well as basis for evaluation and learned lessons.

Keywords: TOGAF standards information base · Enterprise Architecture Management · Standardization · Compliance Evaluation · Metamodelling

1 Introduction

To overcome challenges pertaining to infrastructure heterogeneity and complexity management, Enterprise Architecture (EA) practitioners (see e.g., [1, 7]) and frameworks (see e.g., [13, 30, 35]) advocate the adoption and governance of standards. Generally, standards may be defined at any level of the organization: standard business processes, standard applications, standard technologies etc.; their relevance and alignment within EA must be assessed and governed, considering the management practices already in place.

For example, TOGAF [35] recommends the alignment of EA management activities with Portfolio, Solution Development or Operations Management methods but

H.C. Mayr et al. (Eds.): ER 2017, LNCS 10650, pp. 21–35, 2017.
https://doi.org/10.1007/978-3-319-69904-2_2

falls short in providing guidance on how to integrate the key information artifacts produced and exchanged within these management practices. With a focus on standards management, TOGAF recommends a collection of technical specifications within a "Standards Information Base" (SIB), as a reference for architecture conformance. Implementing a system to "ensure compliance with internal and external standards and regulatory obligations" is considered a key aspect of effective EA governance [7].

The work at hand is motivated by compliance requirements in industry cases. The contribution, labelled with the acronym **CE-SIB** (Compliance Evaluation for Standards Information Bases), is a *modelling method function block* that can extend EA frameworks with a model-based dashboard for defining, aligning and communicating organizational standards. It is designed to be pluggable to arbitrary model-driven management practices, both to their model bases and underlying modelling methods. This means that it provides extensions to all the building blocks of a modelling method and its deployment relies on existing methodologies for agile customization of modelling tools – e.g., the Agile Modelling Method Engineering methodology [22] - and its technological enablers (the ADOxx metamodelling platform [24]). However, the proposal will be abstracted in order to inspire adoption for other frameworks.

The remainder of the paper is organized as follows: Sect. 2 generalizes the problem statement from industry experience and contrasts the approach against related works. Section 3 uses a minimal yet representative example as an explanatory starting point, then it generalizes the CE-SIB building blocks with respect to the generic notion of a *modelling method*. Section 4 discusses an expository instantiation in the form of a modelling prototype. The paper closes with a summative SWOT evaluation.

2 Problem Statement and Background

2.1 Problem Statement

The SIB catalogue proposed by TOGAF [7, 36] holds descriptions of technology products and their versions (e.g., "Apache 2.4") and interoperability standards (e.g., "Web Service Definition Language 2.0") to be used as requirements for procurement. However, TOGAF does not explain (i) how the standards collude conceptually with the TOGAF meta-model; (ii) how architecture development and governance processes should ensure architecture compliance; (iii) what viewpoints should support the depiction and communication of standards compliance; and (iv) how evaluation can be ensured by model-based mechanisms or algorithms.

All these are pragmatic requirements identified in TOGAF-driven industry cases (in banking and public administration sectors) that motivated the work at hand. The major stakeholders in these cases were the technology/solution architects and the operations managers. Their commonly employed tools were Excel (for standards description), Visio and Powerpoint (for communication) with no semantic integration between contents, between standards and architecture elements, and no on-demand reporting mechanisms. To overcome limitations, a model-driven solution is hereby proposed for better maintainability of SIBs and technology portfolios.

EA frameworks like TOGAF [35], FEAF [13] and PEAF [30] propose architecture principles such as “Control Technical Diversity”, “Interoperability”, “Common Use of Applications”, and “Reuse” to govern the selection and implementation of IT solutions; however, they also do not explicitly recommend how governance could be exerted through these principles. The work at hand aims to fill this gap with a Design Science artifact – the CE-SIB *modelling method function block* providing extensions to all building blocks of a modelling method.

2.2 Related Works

Other examples of Standard Information Bases are SAGA, the governmental interoperability framework of the German federal administration, and comparable frameworks like EIF (the European Interoperability Framework) or NISP (NATO Interoperability Standards Profiles) (see [4] for an overview). Like TOGAF’s SIB, their main contribution is to recommend a catalogue of IT standards, with no explicit methodical support on how to monitor these standards and their involvement in other EA layers.

Besides the set of standards catalogued within a SIB, there is also a set of characteristics, namely “qualities” that apply across all architecture building blocks [7]: maintainability, security, reliability and efficiency. According to TOGAF, some of these qualities are easier to describe in the form of “standards” [7], rather than “metrics”. Buckl et al. [9] present an evaluation function using metrics based on probabilistic relations models (PRM), using the quality “availability” for illustration purposes. In contrast, the contribution at hand focuses on the description of required standards as semantically rich modelling objects whose alignment to requirements (or compliance, if we take the internal perspective) is quantified and color-coded in a model-based dashboard based on specific comparison assessment mechanisms.

In [8], an approach for controlling and measuring the degree of standardization of an IT landscape, utilizing fuzzy logic concepts and a basic metamodel for representing the IT landscape, are introduced. The approach allows the calculation of the compliance degree of service categories. In contrast, CE-SIB focuses on standardization degrees of architecture artifacts; however, the approach in [8] can be combined with the foundation provided by CE-SIB: it delivers those artifacts (within a service category) to be recommended as a standard.

The proposal may also be understood as having a more general scope than the EA monitoring approach proposed by [26] in the form of Archimate extensions. At the same time, it instantiates the “embrace pragmatics” theory developed by [6], as the proposal is motivated by case-based requirements of altering standard modelling methods to achieve a pragmatic goal – here, governance of standards adoption. Consequently, the work is also related to [17], which introduced its own notion of “method integration” by placing emphasis on socio-technical implications, whereas our proposal focuses on semantic and functional aspects (some aspects pertaining to the involved collaborative work will be discussed in Sect. 3.3).

3 Design Decisions

3.1 CE-SIB: A Design Science Artifact

EA is typically regarded as a holistic approach which serves as an “umbrella” for specialized management practices (see e.g. [5, 21, 34, 35]). EA Management viewpoints can be anchored in model-driven software engineering (MDSE), business process management (BPM), business planning methods (see e.g. [27] for IT-based scorecards), project portfolio management (see e.g. [10]) and IT service management (see e.g. [11] for integrating ITILs configuration management process with EA practices). In a complex environment with an extensive modelling culture, these viewpoints are supported by different modelling notations and languages (see [2]).

CE-SIB aims to be reusable and pluggable to any of these approaches (and their hybridizations), therefore it extends the *modelling method* building blocks defined by Karagiannis and Kühn [25]. As discussed in [37], a *modelling method function block* has the same components as modelling methods:

- A *modelling language* comprising a modelling notation and a metamodel that defines the language grammar and vocabulary. CE-SIB defines a metamodel fragment (see Sect. 3.2) for “semantic docking” to modelling methods that support the above mentioned management practices. The integration itself relies on the Agile Modelling Method Engineering methodology [22] which facilitates the agile tailoring of modelling methods/tools in response to pragmatic requirements (e.g., those derived from the problem statement in Sect. 2.1);
- A *modelling procedure* comprising the required processes for creating and maintaining a model base. CE-SIB defines a socio-technical procedure for monitoring the standards compliance of architecture building blocks, aiming to replace legacy procedures with a diagrammatic model analysis environment (see Sect. 3.3);
- Model-based *mechanisms/algorithms*: CE-SIB defines quantitative evaluation mechanisms for standards compliance criteria (see Sect. 3.4).

3.2 The CE-SIB Language Fragment: Semantic Docking

The CE-SIB metamodel relies on *semantic docking points* that can be identified in EA-supporting modelling languages or in language hybridizations (e.g., between models expressing EAM, BPM, MDSE, business planning or service management viewpoints). To ensure understandability, we will focus on a simplified yet representative example derived from experience with two viewpoints expressed through two popular modelling languages - Archimate and UML.

A “semantic docking point” is a recurring pattern identified in TOGAF as follows: TOGAF defines a *building block* as “a package of functionality defined to meet the business needs across an organization” [7] and differentiates between: (i) *Architecture Building Blocks* (ABBs) representing the required architecture capabilities (functional view); and (ii) *Solution Building Blocks* (SBBs) representing the concrete components that will be used to implement required capabilities – e.g. concrete application components and technology products. TOGAFs Architecture Development Method cycle refines ABBs into one or more SBBs (see [35], phase G).

An example is Archimate's recommendation to refine *application components* and application interfaces (both ABBs, mapped to "application co-operation" viewpoints and "infrastructure usage" viewpoints) into "UML components" (SBBs mapped on modular parts of a software system). Figure 1 shows an implementation and deployment viewpoint including the used application components, further refined in the corresponding UML (specification level) deployment diagram. The application components (an Archimate concept, here acting as ABB) are refined into technical artifacts which in turn are deployed on nodes (UML concepts, here acting as SBBs). In a modelling tool, this "refinement" will manifest in the form of a machine-readable relation (e.g., visual connectors or hyperlinks across models) subjected to constraints (e.g., domain, range, cardinality etc.) and possibly enriched with its own attributes (input for mechanisms and algorithms). In the same figure, at metamodel level the ABB-to-SBB relation forms the *semantic docking point* for the CE-SIB language block.

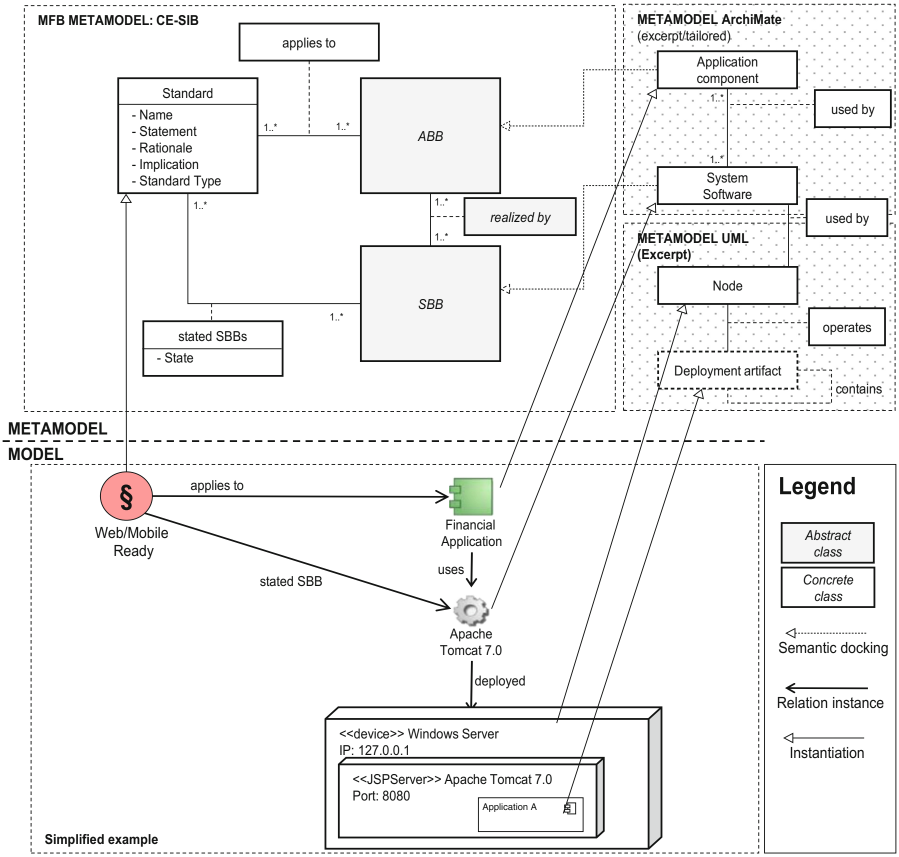


Fig. 1. CE-SIB metamodel block "plugged" to an Archimate-UML hybrid method.

Other examples of docking points to which this may be generalized are: the refinement of data objects into UML classes/objects, the refinement of business processes into UML activity diagrams, the refinement of Archimate nodes into UML nodes (see [3] for a more detailed discussion). In this particular example, CE-SIB enriches the Archimate-to-UML semantic docking point with a modelling concept "Standard" and two semantically-rich relations, namely "Stated SBB" and "Applies to".

The relation "Stated SBBs" specifies all valid and standard-conforming architecture elements specified by a certain standard. Contrasting from TOGAF, in CE-SIB the Standard Information Base is not only a list of SBBs serving as standards. A bundle of SBBs can be assigned to the "Standard" plus qualities such as "Statement" and "Rationale". An example instance for this could relate to "Web Application Server Technologies": the standard might specify the set of concrete web server technologies (SBBs such as Apache Tomcat 7.0, Java Glassfish 4.1) as the technology components (i.e., nodes) an organization's application components can use. These SBBs have their own lifecycle and pass through a series of status in the context of a standard. A status is defined via the attribute "State" of the relation class "Stated SBBs"; while an SBB might be stated as an active standard for the as-is architecture it might be non-conformant for future architectures. The CE-SIB metamodel recognizes this requirement – see the attribute "Standard Lifecycle" with values in conformity with TOGAF: Trial Standard, Active Standard, Deprecated Standard, and Obsolete Standard.

The relation "Applies to" assigns standards to those architecture building blocks which need to adhere to the standards – e.g., the application component "Financial Application" must adhere to the standards "Database Management Systems", "Web Server Technologies", and "Operating Systems". Standards will typically be defined on any level of the EA - they might be used to restrict the set of underlying technology products, utilized for developing, testing, and operating application components; or, if the architecture principle "Interoperability" must be described, the standard would define appropriate interoperability protocols (see architecture principle no. 21 in TOGAF 9, [35]). These standards do not necessarily have to be transferred to a technical level. An architecture principle on business architecture level can be formulated such as "Common Use Applications" (see architecture principle no. 5, [35]). Organizations adhering to this principle try to avoid the introduction of similar and duplicative applications supporting their business processes. In this case applications that are already in place are stated as the standard for certain capabilities or processes.

The modelling class "Standard" is oriented towards the structure of architecture principles (refer to TOGAF's content metamodel in [35]), supporting the formulation of a business case and/or business rationale for each standard in terms of some editable properties: *Name; Statement* (concise definition of the standard including the list of stated SBBs); *Rationale* (listing of the business benefits adhering to the standard); *Implication* (listing of the requirements, both for the business and IT, for adhering to the standard in terms of resources and costs); *Standard Type* (required level of conformance - may adhere to TOGAF's conformance schema [35]).

3.3 The CE-SIB Procedure: Managing Standards Compliance

Since it is a method function block, CE-SIB also defines its application procedure steps to be assimilated by in-place management practices:

Step 1. Adopt and maintain standards. In this step the set "STD" of standards is formulated (i.e., their attributes relevant for EA standards management are described) and promoted within the organization. Standards must be derived from the organizations strategy – i.e., architecture principles [7, 33] and business goals. This is a collaborative effort between subject matter experts and a cross-organizational architecture board to oversee the quality and the strategic/tactical impact of the standards. Each standard is described based on the presented metamodel. Let S be the set of SBBs stated by a standard – i.e., the set of architecture artifacts assigned to a standard via the relation class "Stated SBB", and U be the set of solution building blocks an ABB uses (i.e. is implemented on). The predicate "uses" is represented in the metamodel through the "Realized by" relation assigning SBBs to ABBs, with a variety of possible implementation-level manifestations in a modelling tool (e.g., visual connectors, hyperlinks). Figure 2 illustrates a Standard (Web Application), an ABB (Financial Application), and the sets S and U of solution building blocks (SBBs) - ensured by the standard and used by the ABB, respectively.

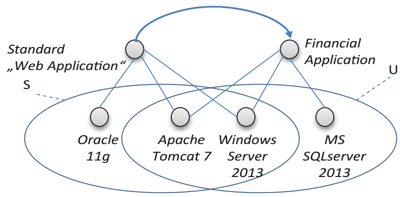


Fig. 2. Example of comparison assessment

The financial application ABB runs on the following SBBs: Apache Tomcat 7.0, Windows Server 2013 and MS SQLserver 2013. The standard "Web Application" assigned to the financial application states the following SBBs: Oracle 11 g, Apache Tomcat 7.0, and Windows Server 2013. The usage of MS SQLserver 2013 is not in conformance with the standard. In order to identify the relevant SBBs (in Fig. 1 the artifacts of modelling class "node") CE-SIB provides mechanisms for evaluating the graph "Application Component > Artifact > Node" and the degree of compliance relative to the compliance types in Table 1.

Step 2. Weigh standards. The defined standards are weighted according to their importance for the organization. Like the weighing of architecture principles (proposed in [23]) each standard is weighted from 1 (minor importance) to 5 (high importance), i.e., $\omega(std) \in \{1, 2, 3, 4, 5\}$.

Step 3. Stipulate SBBs and set lifecycles. Subject matter experts continuously define SBBs best supporting a standard, ensuring that only valid, up-to-date and available SBBs are postulated by the standards. In case of technology standards, non-functional

Table 1. Types of standards compliance

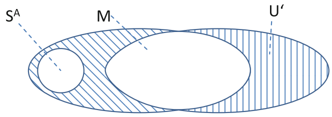
Type	Explanation
Non-conformant	Fulfilment Requirements: $S \cap U = \emptyset$ Description: SBBs which are explicitly not allowed to be used by an ABB Example: No application component shall be implemented on a certain technology component (e.g. technology products which reached end-of-life and vendor-support is not guaranteed anymore)
Compliant	Fulfilment Requirements: $U \subseteq S$ Description: A number of SBBs are endorsed by the standard – at least one of these SBBs must be used Example: A standard "Web Server Technologies" states "Apache 2.4" and "IIS 10" as technology components an application component can be implemented on
Conformant	Fulfilment Requirements: $U \supseteq S$ Description: All stated SBBs are to be used by an ABB, but the ABB might use additional solution building blocks (not stated by the standard) Example: The standard "Allowed Web Application Technologies" might state SBBs such as "Apache Tomcat" and "Unix". Hence, a web application shall use both of these SBBs, however, the developing team has the freedom to use any further technology components
Fully compliant	Fulfilment Requirements: $S = U$ Description: Full conformance between stated SBBs and used SBBs is required Example: An application component needs to use exactly the stated SBBs. Usage of a subset of these SBBs, as well as usage of additional SBBs is not allowed

requirements such as costs, functionality, usability, reliability, supportability [12] need to be considered. A more sophisticated approach is discussed in [8], where categories of services are standardized.

Typically, the owner of the standard will assign appropriate SBBs, by scanning the market for new appropriate (versions of) SBBs. Valuable information sources for technical infrastructures are the mentioned SIBs (SAGA, EIF, NISP, also see [4] for an overview), official vendor support policies like the "Oracle Lifetime Support Policy" (see [28]) as well as existing service level agreements concluded with suppliers. Criteria such as information on internal skills for the support and maintenance of the SBBs are also considered. Based on this information one of the status "Trial Standard", "Active Standard", "Deprecated Standard", or "Obsolete Standard" is assigned: $status(sbb, std) \in \{trial, active, deprecated, obsolete\}$.

Step 4. Define atomic scoring values. Each type of lifecycle state (trial, active, deprecated, obsolete) is assigned to ratings on a scale as recommended in [23]: *r(active)* $:=$ *1, r(trial)* $:=$ *2, r(deprecated)* $:=$ *3 and r(obsolete)* $:=$ *5*. An ABB receives a (desired) low standardization degree (SD), if it primarily uses active SBBs. The scoring can be adapted as required by the EA board when deploying the method.

Step 5. Assess compliance. Compliance levels can be calculated from the perspective of the standards (i.e., the compliance level of a concrete standard along the entire EA) as well as from the perspective of the ABBs (i.e., the compliance level of an ABB along the set of standards it must adhere to). In order to calculate the standardization degree $SD(a, std)$ of an ABB *abb* in the context of a standard *std* (and vice versa) we decompose the set of SBB in the subsets indicated by Fig. 3:



$S^A = \{sbb \in S \setminus U | status(sbb, std) = active\}$, $M = \{sbb \in S \cap U\}$, $U' = \{sbb \in U \setminus S\}$

Fig. 3. Decomposing the SBB set

S^A is the set of SBBs scored "active" but not used by the *abb*. *M* is the set of SBBs stated by the standard and at the same time used by the *abb* independent of their scoring in context of the standard. *U'* is the set of SBBs used by the *abb* but not stated by the standard. Based on these subsets the standardization degree (SD) of an ABB can be calculated for each type of standard (conformant, compliant etc.). Take the running example of the financial application and the standard "Web Application" (see Fig. 2):

$$S^A = \{Oracle\, 11g\},\ M = \{Appache\, 2.4,\ Windows\, Server\, 2013\},\ U' = \{SQLserver\, 2013\}$$

Depending on the type of standard, the weighing of the applied standards, the status of the used SBBs, and the applied atomic scoring values, the standardization degree per standard and per ABB can be calculated. Depending on the type of standard, usage of Oracle 11 g instead of MS SQLserver 2013 might lead to a bad rating of the standardization degree. The standardization degree calculations are performed by model-based assessment mechanisms to be detailed in Sect. 3.4.

Step 6. Address exceptions. Goodhue et al. [18] consider standards without governance to be useless. Peterson [29] discusses the necessity of the institutionalization of monitoring processes in terms of diagnosing IT governance effectiveness and value contribution. CE-SIB recognizes these requirements and proposes evaluating the standardization degree on an ongoing basis. In cases of non-compliance, change requests to improve the architecture need to be raised. However, Gartner [16] rates over-standardization, as one of the worst practices in EAM. Hence, in cases of identified non-compliance, mechanisms for interim conformance are provided. These are exceptions that must be corrected within a granted lifespan of the exception. The CE-SIB method allows exceptions for the tuple of an ABB and its used SBBs, in case of non-compliant SBBs (e.g., SBBs in state "obsolete"). CE-SIB reflects exceptions by neutralizing bad ratings via exceptions. Thus, the value *r(obsolete)* $:= 5$ for non-compliant SBBs is mitigated by subtraction of the value 4 (see next section).

Step 7. Create viewpoints for decision support. As the CE-SIB method is meant to be framework-agnostic, no concrete viewpoints are stipulated - graph-based diagrams (with nodes and edges), matrices etc. can be used. Current deployments have been coupled with the CE-HM (Compliance Evaluation Featuring Heat Maps) mechanisms introduced in [23], which enables color-coding mechanisms ("heatmaps") in arbitrary modelling notations, while also propagating such visual cues to superordinated levels of the EA (e.g., superordinated business processes, business capabilities).

3.4 The CE-SIB Mechanisms: Computing Standardization Degrees

Based on the existing model base (set of ABBs, assigned SBBs, and defined standards) the following metrics compute standardization degrees, relative to Fig. 4. The numbers depicted within the subsets present the ratings of SBBs in context of the particular type of standard (fully conformant, conformant, compliant etc.).

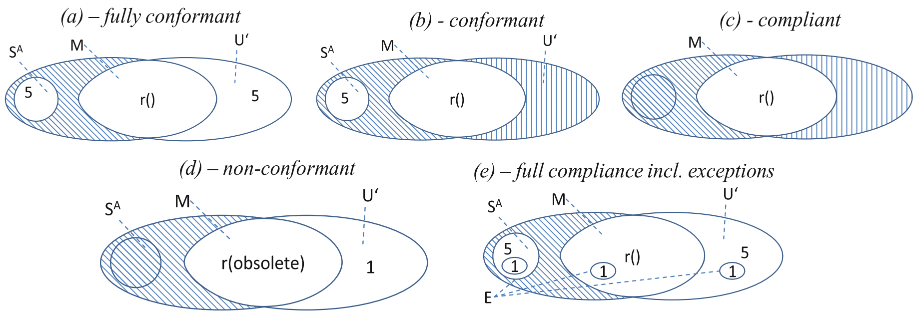


Fig. 4. Sets and scores for different compliance types

The formulas for degrees are marked with the corresponding letters from the figures:

$$SD_{fully}(abb, std) := \frac{\sum_{sbb \in \mathrm{M}} r(status(sbb, std)) + 5 \times |\mathrm{S}^A| + 5 \times |\mathrm{U}'|}{|\mathrm{M}| + |\mathrm{S}^A| + |\mathrm{U}'|} \quad \text{(a)}$$

$$SD_{conformant}(abb, std) := \frac{\sum_{sbb \in M} r(status(sbb, std)) + 5 \times |\mathrm{S}^A|}{|M| + |\mathrm{S}^A|} \quad \text{(b)}$$

$$SD_{compliant}(abb, std) := \begin{cases} \frac{\sum_{sbb \in M} r(status(sbb, std))}{|M|}, & if\ |M| > 0 \\ 5, & else \end{cases} \quad \text{(c)}$$

$$SD_{non\text{-}conf}(abb, std) := \frac{\sum_{sbb \in M} r(status(sbb, std)) + |\mathrm{U}'|}{|M| + |U'|} \quad \text{(d)}$$

Note, that in case of required non-conformance all stated SBBs will be scored r(obsolete) as usage of any of the stated SBBs should be avoided. All other used SBBs are scored with a value of 1.

From the viewpoint of an ABB *abb*, the standardization degree *SD(abb)* of the ABB is the weighted average of the standardization degrees of the tuples of the ABB and its assigned standard, i.e.:

$$SD(abb) := \frac{\sum_{i \in STD} SD(abb, std_i) \times \omega(std_i)}{\sum_{i \in STD} \omega(std_i)}$$

where *STD* is the set of standards assigned to an ABB.

From the viewpoint of a standard *std*, the standardization degree *SD(std)* is the average of the standardization degrees of the tuples of the standard and the ABBs it applies to:

$$SD(std) := \frac{\sum_{j \in \mathrm{ABB}} SD(abb_j, std)}{|\mathrm{ABB}|}$$

Situation (e) in Fig. 4 illustrates how exceptions are considered for a standard requiring full-compliance (E is the set of obsolete SBBs with granted exceptions):

$$SD_{fully,ex}(abb, std) := SD_{fully}(abb, std) - \frac{|\mathrm{E}| \times 4}{|\mathrm{M}| + |\mathrm{S}^A| + |\mathrm{U}'|} \qquad \text{(e)}$$

4 Implementation and Evaluation

The proof-of-concept depicted in Fig. 5 was implemented on the metamodelling platform ADOxx - made available as part of the Agile Modelling Method Engineering framework [22] by the Open Models Initiative Laboratory [24]. It integrates a hybrid of

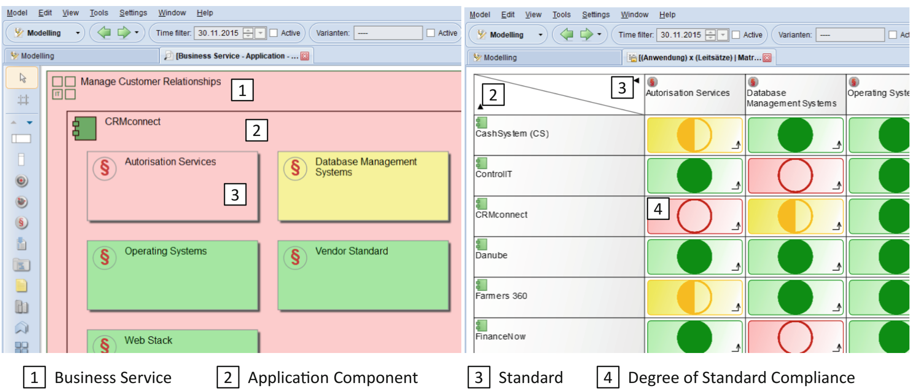


Fig. 5. Exemplary viewpoints: (a) compliance clustermap, (b) compliance matrix

Archimate viewpoints and a subset of UML (namely class/object diagrams, component diagrams, and deployment diagrams) extended with the CE-SIB plug-in, following Shneiderman's visualization "mantra" overview first, zoom and filter, and details on demand [32]. Figure 5a and b exemplarily depict two viewpoints. The clustermap (Fig. 5a) is an EA viewpoint recommended by [14] for communicating standards conformity and exceptions. In Fig. 5a the worst score is propagated to the "higher" levels (from application components to superordinate business processes). Within the application components the relevant standards are depicted. This viewpoint is intended to give an "overview first" on weak spots within the EA.

The matrix view in Fig. 5b is recommended by TOGAF [35] to communicate relationships between architecture artifacts, giving insight to scorings of ABBs in context of the assigned standards. On the x-axis standards such as "Authorization Services" and "Database Management Systems" are depicted. Based on the discussed thresholds and metrics matrix cells are color-coded.

As the CE-SIB method block was introduced here as a Design Science artifact, it can be subjected to the wide tableaux of evaluation criteria surveyed by [31]. The current implementation was driven by requirements from industrial cases (a banking institution and an organization from the public administration sector) therefore certain criteria gained priority:

Generality: CE-SIB is reusable for any semantic "docking points" (as defined in Sect. 3.2) identified between arbitrary metamodels. This relies on the Agile Modelling Method Engineering framework [22] - the key enabler for agilely plugging the CE-SIB block to existing modelling method implementations (where relevant concepts can fulfill the ABB and SBB roles).

Consistency with organization (fit with organization requirements): The implementation was tailored for the mentioned industrial cases to replace legacy Excel-based methods and to support already in place model-based management practices. The generality factor mentioned above ensures that similar requirements from organizations of different domain-specificity may be agilely satisfied. By building on the existing model base, efforts for maintaining the SIB could be dramatically reduced by approximately 70% (based on stakeholder feedback). Additionally, standards compliance reports are delivered up-to-the-minute, as opposed to the annual basis reporting of the legacy data acquisition project.

Consistency with people (usability): ADOxx was employed as the underlying implementation platform to benefit from its built-in usability and understandability facilitators: the basic task of creating a new version of a standard was reduced to 5 clicks only; the change history is written automatically and all owners of affected ABBs are informed automatically; reports such as in Fig. 5a and b are updated automatically without any additional manual modelling efforts.

5 Concluding SWOT Analysis

A SWOT evaluation summarizing the key learned lessons was derived from hands-on experience with the implementation and interviews with key stakeholders in their respective organizations.

Strengths: CE-SIB allows nonambiguous definition and communication of standards and can be integrated in commonly used EA frameworks. Adherence to these standards becomes measureable. In the course of the evaluation it was shown that usage of the method and communication of standards compliance degrees (based on modelling viewpoints) lead to comprehensible results. Understandability and acceptance were assessed through qualitative interviews with major EA stakeholders, where the proposal was deployed in modelling tools agilely extended through the Agile Modelling Method Engineering methodology.

Weaknesses: One major restriction of CE-SIB is that it requires a model-based system engineering (MBSE, see [15]) approach to EA management (in contrast to a traditional document-based approach). Thus the EA documentation must be available as a diagrammatic model base, with models depicting different EA facets under an overarching metamodel (for which CE-SIB acts as integrator).

Opportunities: Standards are defined from the point of view of different domains and organizational units and different standards may be conflicting in their statements regarding SBBs. Providing means to uncover these inconsistencies will add another valuable feature to CE-SIB. The strict focus on the MBSE can be relaxed by applying data integration and clearance mechanisms from the fields of business analytics as discussed in [19]. For this, future work will focus on a Data Integration and Cleansing Environment (DICE) (see [20]) implemented on the same metamodelling platform.

Threats: The stakeholder involvement has shown that the definition of standards throughout an organization requires strong negotiating skills, persuasiveness, and political savviness. The main touch points, contrasted to TOGAFs ADM the phases B-D (where the architectures are designed), phase E (where the best solution is chosen), and H (evaluation) have to be clearly defined. Currently CE-SIB does not address these touch-points in detail. Evaluation results clearly show that more detailed guidance on implementing the CE-SIB procedures in the organization is required.

Acknowledgment. The work of Dr. Robert Buchmann is supported by the Romanian National Research Authority through UEFISCDI, under grant agreement PN-III-P2-2.1-PED-2016-1140.

References

1. Accenture: Architecture frameworks for client/server and netcentric computing. In: Myerson, J.M. (eds.) Enterprise Systems Integration, pp. 39–78 (2002)
2. Anaby-Tavor, A., et al.: An empirical study of enterprise conceptual modeling. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 55–69. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04840-1_7
3. Armstrong, C., et al.: Using the ArchiMate Language with UML (2013). http://cdn2.hubspot.net/hub/183807/file-1805596253-pdf/site/media/downloads/W134.pdf?t=1418385713847
4. Beauftragte der Bundesregierung für Informationstechnik (BfIT): SAGA-Modul Grundlagen Version de.bund 5.1.0 (2011)

5. Bernard, S.A.: An Introduction to Enterprise Architecture. AuthorHouse, Bloomington (2012)
6. Bjeković, M., Proper, H.A., Sottet, J.-S.: Embracing pragmatics. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) ER 2014. LNCS, vol. 8824, pp. 431–444. Springer, Cham (2014). doi:10.1007/978-3-319-12206-9_37
7. Boh, W.F., Yellin, D.: Using enterprise architecture standards in managing information technology. J. Manag. Inf. Syst. **23**(3), 163–207 (2006)
8. Brückmann, M., et al.: Evaluating enterprise architecture management initiatives-how to measure and control the degree of standardization of an IT landscape. In: Mendling, J., Rinderle-Ma, S., Esswein, W. (eds.) EMISA, pp. 155–168 (2009)
9. Buckl, S., et al.: A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftinformatik Proceedings 2007, p. 65 (2007)
10. Colombo, A., et al.: The use of a meta-model to support multi-project process measurement. In: 15th Asia-Pacific Software Engineering Conference, APSEC 2008, pp. 503–510. IEEE (2008)
11. Correia, A., e Abreu, F.B.: Integrating IT service management within the enterprise architecture. In: Fourth International Conference on Software Engineering Advances, ICSEA 2009, pp. 553–558. IEEE (2009)
12. CIO Council: A Practical Guide to Federal Enterprise Architecture, Version 1.0 (2001)
13. CIO Council: Federal Enterprise Architecture Framework, Version 1.0 (1999)
14. Ernst, A.M.: Enterprise architecture management patterns. In: Proceedings of the 15th Conference on Pattern Languages of Programs, pp. 7:1–7:20. ACM, New York (2008)
15. Fisher, A., et al.: 3.1. 1 model lifecycle management for MBSE. In: INCOSE International Symposium, pp. 207–229. Wiley Online Library (2014)
16. Gartner: Thirteen worst enterprise architecture practices, Report No. G00164424 (2009)
17. Goldkuhl, G., Lind, M., Seigerroth, U.: Method integration: the need for a learning perspective. IEE Proc.-Softw. **145**(4), 113–118 (1998)
18. Goodhue, D.L., et al.: The impact of data integration on the costs and benefits of information systems. MIS Q. **16**(3), 293–311 (1992)
19. Grossmann, W.: A conceptual approach for data integration in business analytics. Int. J Softw. Inform. **4**, 53–67 (2010)
20. Grossmann, W., Moser, C.: Big data—integration and cleansing environment for business analytics with DICE. In: Karagiannis, D., Mayr, H., Mylopoulos, J. (eds.) Domain-Specific Conceptual Modeling, pp. 103–123. Springer, Cham (2016). doi:10.1007/978-3-319-39417-6_5
21. Hanschke, I.: Strategic IT Management: A Toolkit for Enterprise Architecture Management. Springer Science & Business Media, Heidelberg (2009). doi:10.1007/978-3-642-05034-3
22. Karagiannis, D.: Agile modeling method engineering. In: Proceedings of the 19th Panhellenic Conference on Informatics, pp. 5–10. ACM, New York (2015)
23. Karagiannis, D., Moser, C., Mostashari, A.: Compliance evaluation featuring heat maps (CE-HM): a meta-modeling-based approach. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 414–428. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31095-9_27
24. Karagiannis, D., Buchmann, R.A., Burzynski, P., Reimer, U., Walch, M.: Fundamental conceptual modeling languages in OMiLAB. In: Karagiannis, D., Mayr, H., Mylopoulos, J. (eds.) Domain-Specific Conceptual Modeling, pp. 3–30. Springer, Cham (2016). doi:10.1007/978-3-319-39417-6_1
25. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: EC-Web, p. 182 (2002)

26. Lara, P., Sánchez, M., Villalobos, J.: Bridging the IT and OT worlds using an extensible modeling language. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 122–129. Springer, Cham (2016). doi:10.1007/978-3-319-46397-1_10
27. Lichka, C., et al.: IT-based balanced scorecard. In: WISU Wirtschaftinformatik, p. 31 (2002)
28. Oracle: Oracle Lifetime Support Policy: Oracle and Sun System Software (2012). http://www.oracle.com/us/support/library/lifetime-support-hardware.pdf
29. Peterson, R.: Crafting information technology governance. Inf. Syst. Manag. **21**(4), 7–22 (2004)
30. Pragmatic EA Ltd.: Pragmatic EA Framework Version 2.0 (2010)
31. Prat, N., et al.: Artifact evaluation in information systems design-science research-a holistic view. In: PACIS 2014 Proceedings - Pacific Asia Conference on Information Systems, p. 23 (2014)
32. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of IEEE Symposium on Visual Languages, pp. 336–343. IEEE (1996)
33. Stelzer, D.: Enterprise architecture principles: literature review and research directions. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave-2009. LNCS, vol. 6275, pp. 12–21. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16132-2_2
34. The Open Group: ArchiMate 3.0 Specification (2016)
35. The Open Group: TOGAF Version 9.1 (2011)
36. Varnus, J., Panaich, N.: TOGAF 9 enterprise architecture survey results. In: 23rd Enterprise Architecture Practitioners Conference (2009)
37. Zivkovic, S., et al.: Facilitate modelling using method integration: an approach using mappings and integration rules. In: ECIS 2007 Proceedings, pp. 2038–2049 (2007)