Interrelating Concerns in EA Documentation – Towards a Conceptual Framework of Relationships

Sabine Buckl, Florian Matthes, Christian M. Schweda
{buckls,matthes,schweda}@in.tum.de

Abstract: Over the last years a multitude of approaches and frameworks making prescriptions on how to document an enterprise architecture (EA) have been developed. These approaches target different purposes and correspondingly different concerns (areas of interest) in the architecture. In this way, a company seeking to develop or evolve an organization-specific EA documentation technique based on different approaches most likely runs into difficulties to understand the interdependencies between the different frameworks. In this research-in-progress paper, we revisit well-known documentation approaches to devise different types of concern relationships that may be helpful during development and evolution of an organization-specific EA documentation technique.

1 Introduction and motivation

The increasing frequency of change, modern enterprises face in today’s globalized and competitive environments, leads to a rising internal complexity of the socio-technical system enterprise. A promising and commonly accepted instrument to deal with this complexity and to foster business-IT-alignment is enterprise architecture (EA) management [HV93, Lan05, RWR06]. Originating from the field of information systems architecture (cf. [Zac92]), EA management takes a holistic perspective targeting all areas of an enterprise from business and organizational aspects via application and information to infrastructure and data aspects. EA is thereby in the sense of the ISO Standard 42020 understood as the ”fundamental organization of a system [enterprise] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution” [Int07]. A multitude of methods to EA management has been developed by researchers, practitioners, and standardization bodies (cf. [Fra02, KW07, Lan05, The09, vdRvV08, RWR06]). Although differing in respect to the scope, reach, and focus, the proposed methods usually distinguish the following activities of the EA management function: a) document and maintain the current state of the EA, b) develop and describe planned and target states of the EA, c) enact and communicate planned EAs and architectural principles, and d) analyze and evaluate architectures.

A central challenge arising during the aforementioned typical activities of EA management is stakeholder involvement [The09, Sch08]. To address this challenge the architectural documentations referring to the current, planned, or future states of the EA, need to represent the corresponding concerns of the stakeholders. Thereby, concerns are defined in accordance with the ISO Standard 42010 as ”those [areas of] interests which pertain to
the system’s [enterprise’s] development, its operation or any other aspect that are critical or otherwise important to one or more stakeholders” [Int07]. Put in other words, a concern can be understood as the area of the enterprise that the respective stakeholder is interested in. Whereas, a product manager for instance is interested in the performance of the business process and the services responsible for creating the product, an application manager is concerned with the standard conformity of the business applications.

In the holistic perspective of EA management the two aforementioned concerns are clearly interrelated. Crosspoints are the terms service and business application. The thereby denoted concepts are most likely related or may even be identical. The differences in terminology can be explained by the different language communities the two stakeholders belong to. To develop a comprehensive EA description, covering the different concerns of the stakeholders, the relations between the concerns should be made explicit. This directly yields the research questions that our paper seeks to address:

*What types of relationships between EA concerns exist? How can these relationships be utilized in developing and evolving EA models?*

In the remainder of the article above research questions are approached in a practice-driven manner. More precisely, we explore exemplary concern descriptions from the *Enterprise Architecture Management Pattern Catalog* [Cha10], a collection of best-practice solutions for recurring problems in the context of EA management (cf. Section 2). Thereby, special emphasis is put on the conceptual models that are used in the pattern catalog to present the conceptualizations underlying the concerns. From this, we identify different types of concern relationships and discuss how these relationships can be defined. Complementing the forerusting findings, we revisit the state-of-the-art of concern relationships as contained in prominent EA management approaches from literature (see Section 3). In Section 4, the paper concludes with a critical reflection of the achieved results and an outlook on future areas of research.

## 2 Concern relationships

The *Enterprise Architecture Management Pattern Catalog* [Cha10] presents a collection of best-practice solutions to recurring problems in the context of EA management. Centrally, the catalog establishes a ‘trifecta’ of successful solutions consisting of *management methods, architectural viewpoints* and *information models*\(^1\). Especially the information models reflect areas-of-interest in the overall EA, i.e. correspond to architectural concerns in the sense of the ISO 42010 [Int07]. Put in other words, it is sensible to understand an information model as representative of an underlying domain ontology targeting a specific part of the EA. Against the background of this understanding, we revisit three selected information models, whose selection is inspired by a practice case. In this case (see [Die08] for a more comprehensive description of the case) an international financial service provider

\(^1\)In the pattern catalog, the conceptual meta-models describing the EA or parts thereof are called “information models”.

244
was using the pattern catalog to implement compliance management in the EA management context. The financial service provider started compliance management by establishing a ‘book of standards’ for business applications, i.e. a collection of technologies that comply to architectural presecriptions. Based on this set-up, a more mature EA management function was built on architectural solutions\(^2\), which define a set of matching technologies on different tiers together with additional information on how to integrate these technologies into a solution to realize e.g. a four tiered architecture. In this sense the initial concern of compliance management in the example case was ‘broadened’. Further, the the company experienced the pressure to standardize its application landscape. This gave rise to a different concern, in which it was determined, if a business application was custom or standard software, respectively.

From the above discussions, we can derive a set of relationship types relating architectural concerns. The most basic type represents the fact that concerns share concepts or do not. In the light of the discussion on terms and their meanings, two different ways to understand ‘share’ exist: syntactic and semantic. While sharing in the latter sense means that two concerns have a concept with similar meaning but possibly different names in common, syntactic sharing describes that two concerns are related via a concept with common name. To qualify these two different understandings of ‘share’, we denote ‘syntactic sharing’ by the name intersection. Based on the intersection, we consistently establish the notion of compatibility, indicating that all intersecting concepts of two concerns are also semantically shared. If in contrast one intersecting concept is assigned distinct meanings in both concerns, the concerns are called incompatible. Admittedly, the foregoing definition of ‘semantic sharing’ is fairly informal and one might argue, that equivalent meanings in a strictly ontological sense may need a more precise definition. In response, we rely on a rather weak form of semantic sharing here, assuming that the concepts’ meanings are given as textual definitions. Based on these equivalence of meanings is interpersonally defined via stakeholder consensus in accordance to Kamlah and Lorenzen [KL96], i.e. two concepts share a common meaning, if “any informed member of the language community would [say so]”.

The following two information models from the EA management pattern catalog [Cha10], namely STANDARD VS. CUSTOM SOFTWARE (I-41) and USAGE OF ARCHITECTURAL SOLUTIONS (I-6), illustrate the compatibility relationship. To facilitate considerations on the relationships between concerns, we present the central parts of the information models, but abstain from adhering to the pattern structure as utilized throughout the pattern catalog.

**Standard vs. Custom Software (I-41)**

**Concern:** You want to know, whether a business application is custom or standard (off-the-shelf) software or not.

Important concepts in the information model of Figure 1 are:

**BusinessApplication** A software system, which is part of an information system used to support an organization’s business processes. The type attribute, if set, indicates whether the application is custom or off-the-shelf software.

\(^2\)Sometimes architectural solutions are also referred to as technology stacks or platforms.
BusinessApplicationType  Enumeration of values stating that a corresponding business application is either custom or standard software.

Usage of Architectural Solutions (I-6)
Concern: You want to know which business application conforms to which architectural solution, if any.

Important concepts in the information model of Figure 2 are the BusinessApplication concept defined above and:

ArchitecturalSolution  An approved set of compatible technologies that can be used to build business applications. The architectural solution further determines how its constituting technologies are used in a thereby realized business application.

conformsTo  Relationship expressing that one or more business applications are built upon the technologies described in the corresponding architectural solution, if any.

The two concerns Standard vs. Custom Software and Usage of Architectural Solutions represented by their corresponding information models I-6 and I-41 can be considered intersecting via the shared concept of the BusinessApplication. This concept’s semantics is textually given in both models with the same description. Further, the intersection quality of the concept can be illustrated along the fact that a valid instance of the class BusinessApplication in I-6 is also a valid instance of the class with same name in I-47. In this light, the two concerns are not only intersecting but are also compatible, i.e. can be integrated with each other without causing contradictions.

Grounding on the basic notion of compatibility, subconcern and superconcern relationships can be established. One concern A is subconcern of another concern B, if the concerns are compatible and all concepts of A are intersecting concepts. Put in other words,
concern B completely covers the area of interest represented by concern A. This relationship type mirrors the notion of more abstract or more concrete concerns as outlined above. In a mathematical sense, the relationship type subconcern and its counterpart superconcern further give rise to a notion of concern equivalence. Two concerns A and B are equivalent, if A is both sub- and superconcern of B. To exemplify the understanding of sub- and superconcern, respectively, we introduce the information model Architectural Solution Conformance (I-67) below.

**Architectural Solution Conformance (I-67)**

*Concern:* You want to know to which extent your custom-built business applications comply with the architectural solutions that they are allowed to be built on.

![Figure 3: Information model pattern I-67 (adapted from [Cha10])](image)

Complementing the concepts from the information model in Figure 3 the following derive expression using the Object Constraint Language (OCL) [OMG06] applies:

```ocl
context BusinessApplication::standardConform
derive: type == STANDARD | allowed.isEmpty() | allowedFor->contains(conformsTo)
```

Important concepts in the information model (cf. Figure 3) are the ones used in I-6 and I-41, respectively. In addition, the following concepts are employed:

- **allowedFor** Relationship expressing that a business application is allowed to build upon the technology set as described by zero or more architectural solutions. If no solution is specified, the business application is allowed to use any technologies.

- **standardConform** Attribute indicating whether a business application conforms to the standards, i.e. is off-the-shelf software or uses an allowed architectural solution, or not.

Relating the concerns **Standard vs. Custom Software** and **Architectural Solution Conformance**, i.e. the information models I-41 and I-67, we can exemplify the sub-concern and super-concern relationship, respectively. Described in more detail,
we see that I-67 shares all modeling concepts of I-41 and further also the textually described semantics for the corresponding classes. In consequence, any valid instantiation of classes from the latter information model also forms a valid instantiation of concepts from I-67, which nevertheless adds some more concepts, i.e. classes, attributes, and associations. The aforementioned arguments also apply on the relationship between the concerns USAGE OF ARCHITECTURAL SOLUTIONS and ARCHITECTURAL SOLUTION CONFORMANCE, in which the former can be considered a sub-concern of the latter.

Refraining the practice case introduced at the beginning of the section, a key benefit of understanding the relationship becomes obvious. Having started with architectural descriptions satisfying the concerns I-6 and I-41, our financial service provider can derive further concerns that not only relate to the already addressed concerns, but are also compatible, i.e. employ a similar understanding of the relevant concepts. Additionally, the sub-super-relationship between concerns facilitates consideration on possible next stages of refinement, when it comes to architectural descriptions. In the practice case, the financial service provider did not find the concerns as covered by I-6 and I-41 to be sufficient for a comprehensive compliance management. Knowing, that I-67 not only subsumes I-6 and I-41, but also provides additional concepts, the company was able to adopt the information model for architectural descriptions as devised by I-67.

In the latter sense, the relationships between the concerns can provide valuable insights into the different ways to conceptualize an EA, but also to evolve the corresponding description technique. To facilitate considerations on the relationships between concerns, we further propose to use a graphical visualization thereof, called concern map. An exemplary map is shown as part of this paper’s example section in Figure 4.

![Concern map for the three exemplary concerns](image)

Figure 4: Concern map for the three exemplary concerns

3 Concerns in the state-of-the-art in EA descriptions

EA management is a research topic with an increasing number of publications [LW04], in which especially the fields of EA modeling and EA analysis are heavily researched. Central to both activities is the notion of the architectural description reflecting the corresponding architectural concerns. Subsequently, we explore the state-of-the-art in EA modeling and analysis with a special emphasis on the EA descriptions used thereby as well as their underlying concerns.
The approach of multi-perspective enterprise modeling (MEMO) was initially presented by Frank in 2002 [Fra02]. Therein, Frank outlines a modeling framework, which is based on an extendable set of special purpose modeling languages. The special purpose modeling languages correspond to the different language communities, which typically exist in an enterprise, e.g. salespersons or project managers. By further providing a common meta-language, the **MEMO meta modelling language (MML)** [Fra09], which the special purpose languages rely on, the integration of the different languages can be facilitated. Examples for special purpose languages are e.g. the **strategy modelling language (MEMO-SML)** [Fra99] and the **organization modelling language (MEMO-OrgML)** [FJ01]. Concepts of two different languages can be associated, i.e. are common to both languages. Therefore, the different concerns addressed by the MEMO special purpose languages can be regarded as being intersected and compatible.

The communication challenge already mentioned in the approach presented by Frank, is further discussed by Buckl et al. in [BEL+09], Schelp and Winter in [SW09], and Schönherr in [Sch08]. While the first publication refers to the communication challenge within an enterprise, the latter two publications discuss the different language communities mirroring the academic groups conducting research in the area of EA management. Due to the absence of a standardized terminology and a commonly-accepted description language for EA management, the different approaches and communities have developed their own terms, leading to incompatibility issues, when different approaches are combined.

A systemic perspective on EA modeling is presented by Wegmann et al. in [WBL+05]. They provide a method and a tool to formalize the alignment of the multiple levels that constitute an EA. In particular, they propose to organize the different concepts that constitute the EA in organizational and functional levels. Thereby, the functional levels represent behavioral and the organizational levels the constructional hierarchy. Within each organizational level, two different viewpoints are available, the **information viewpoint** – a black box view on the respective concern – and the **computational viewpoint** – a white box specification of the concern. These viewpoints can be refined, which results in a hierarchy of viewpoints and the underlying information, respectively. Based on this understanding, a super- and sub-relationship between different viewpoints and concerns can be identified. Furthermore, Wegman et al. point out that a "vocabulary mapping" [WBL+05] between related concepts on different organizational levels has to be performed, which can be ascribed to the already mentioned compatibility question.

In [JE07] Johnson and Ekstedt discuss an approach to EA decision making based on EA models and analyses. Based on the current documentation of the EA, future EA scenarios are derived and assessed in respect to selected quality attributes, as e.g. performance, interoperability, availability, security, or usability. For each of these quality attributes, an influence diagram is presented by the authors, which details on causal dependencies between architectural properties. If for instance availability is considered, the reliability or recoverability of the system under consideration influences the overall availability. Thus, the concerns included in an influence diagram have a subconcern/superconcern relationship to each other, while concerns of different influence diagrams may be intersected.

Central ideas for a language for describing EAs are outlined by Jonkers et al. in [JvBA+03] and further elaborated in [JGBvB05]. They discuss two requirements that relate to the no-
tion of the concern, namely meta model flexibility and integration of heterogeneous models. Meta model flexibility is thereby meant to demand that general EA description concepts can be refined to organization-specific concepts and standards. Conversely, a language for describing EAs must facilitate the integration of heterogeneous models, as special purpose languages for specific parts of the EA exist, whose concepts can be ’translated’ into or ’associated’ with concepts for EA description. Jonkers et al. further discuss in [JvBA+03] the distinct levels of detail, on which the corresponding descriptions may act. In this sense, one or more concerns from special purpose modeling languages might be ’aggregated’ into a single concern at the EA description level. In [JGBvB05] Jonkers et al. provide a core meta-model for architectural descriptions, more precisely behavioral aspects thereof, based on a dichotomy of service- and implementation-aspects. The corresponding basic concepts service and behavior element are refined to specific business, application and infrastructure-layer concepts, respectively. Nevertheless, the basic modeling language allows to omit concepts on some of these layers, if they are not of interest for the specific modeling purpose. Put in the terminology of our concern-discussion, the language implicitly allows for the derivation of subconcerns.

4 Conclusion and Outlook

In this paper we discussed the concept concern in the context of EA modeling and EA management, respectively. Concerns thereby represent distinct information demands in respect to the described architecture and can hence be reflected by information models that provide a ’schema’ for the information to be modeled and collected during the corresponding EA management process. Based on this model-centric understanding of concern, we revisited the state-of-the-art in EA modeling and analysis to derive a set of relationship types that may exist between concerns. Taking also into account the semantic aspects of the information models, we further refined the understanding of the relationship types while abstaining from mathematical formalization thereof. Complementing the forestanding considerations, we exemplified the relationship types with three selected concerns from the EA Management Pattern Catalog of [Cha10].

This paper presents itself as a research-in-progress paper, that outlines the idea of concern relationships. From this point different directions for further development and research are open. At first, the informal presentation of the relationship types in this paper should be complemented with a more formal understanding thereof. In this respect, the semantics of the different relationship types could be formally defined based on a formal understanding of architectural descriptions. This would then allow for more intricate considerations on the relationships between information models that reflect the corresponding concerns. Secondly, concern relationships might not only be an issue of theoretical interest, but may prove to be interesting for supporting maturity considerations for EA management functions. In this respect, the relationships could also be beneficial for evolving an already existing EA management function towards a ’broader’ and ’deeper’ coverage of the overall EA. This subject has yet not been investigated in practice, although the notion of the ”related” concerns as outlined by Ernst in [Ern10] and in [Cha10] have shown to be bene-
ficial during the process of establishing and evolving an EA management function.

References


