Ontology for Characterising Architecture Frameworks

Oddrun Pauline Ohren

SINTEF ICT, PO Box 124, Blindern, N-0314 Oslo, Norway
oddrun.ohren@sintef.no

Abstract. This paper outlines an ontology for characterising architecture frameworks. The ontology is based on the metamodel of MAF, and is currently being tried out on a set of well-known existing frameworks. Selected results from this trial are presented.

1 Background and motivation

During the last couple of decades, quite a few architecture frameworks have been presented. While they differ considerably in a number of respects, their application domains tend to overlap, hence potential users need to be able to compare frameworks with each other. Although some work is performed concerning architecture framework issues [1, 2], we have not been able to find a generally applicable, simple framework for assessment and comparison of architecture frameworks. As a first step towards such a mechanism, this paper outlines an ontology for characterising architecture frameworks. An architecture framework may be viewed as *a set of rules, guidelines and patterns* for describing the *architecture* of *systems*. According to the IEEE 1471 standard [3] 'architecture' is the 'fundamental organization of a *system* embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution'. The term 'system' is defined as 'a collection of components organized to accomplish a specific function or set of functions'. MAF [4] is a *model-based* architecture framework, implying a strong focus on models as the main formalism for describing architectures. The model-based approach also implies that the MAF metamodel be expressed as a conceptual model for architecture frameworks. This paper illustrates how the MAF metamodel forms the basis for defining dimensions along which architecture frameworks may be characterised.

2 Characteristics of architecture frameworks

Architecture frameworks can be characterised by several distinguishing features, the types of which are shown in **Fig. 1** as a UML specialisation class hierarchy.
The types of characteristics identified are:

- *Application domain characteristics*, specifying to which kind of systems the framework may be applied. The 'kind of systems' is defined by *system type* and *system scope*;
- *Conceptualisations of architecture or application domain*, indicating whether the framework provides ontologies covering relevant aspects of the application domain, the domain of architectures and architecture descriptions, and possibly other relevant areas;
- *Prescriptions for architecture descriptions*, indicating *how* and to what *degree of specificity* the framework prescribes *content*, *organisation* and *representation* of the products forming the architecture description;
- *AD methodological characteristics*, expressing whether or not the framework provides a methodology for developing ADs, and if so, whether it specifies an *AD development process*, *supports architecture evolution*, provides *consistency and conformance principles* for the ADs, whether it is *supported by software tools*, etc.;
- *Relations to other frameworks*, indicating whether (and how) the framework is related to other frameworks in any way worth documenting, be it factual or purely conceptual;

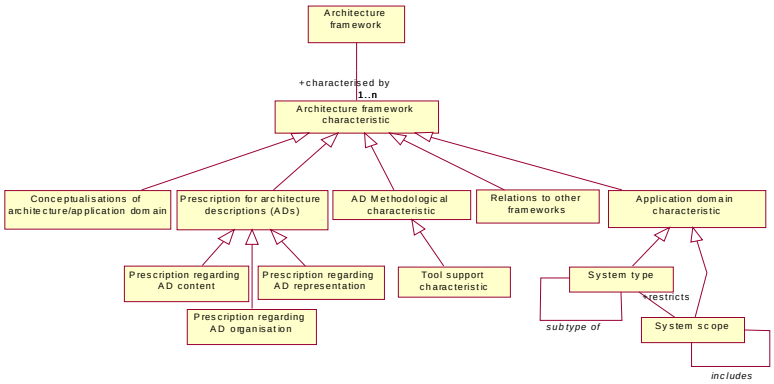

Fig. 1. Architecture framework characteristics

Within each of the characteristic types above, one or more value sets are specified, providing predefined, possibly interrelated values to be used as descriptors of the framework in question regarding that particular characteristics. For example, the value set of 'System type' as an application domain characteristic contains the system types 'System', 'Enterprise' and 'Software system', with a 'subtype of' relation between 'System' and the other two. Hence, an enterprise architecture framework will typically be assigned 'Enterprise' as its application domain. The elements of the value sets are instances in the characterisation ontology and represent a formalisation of the corresponding characteristic types. The degree to which a characteristic type is suitable to formalisation varies considerably. Some characteristic types are best described informally, and therefore, using the characterisation ontology never tells the whole story about an architecture framework. However, it provides a template for

recording information about any architecture framework. This is beneficial whenever there is a need to assess or compare architecture frameworks, for instance in cases where several frameworks are candidates for an application domain, or we need to relate architecture descriptions originating from different frameworks.

3 Comparing some existing enterprise architecture frameworks

We have studied a set of existing frameworks, described them in a semi-structured way in [5], and are now in the process of characterising them using the ontology above. The frameworks studied are Federal Enterprise Architecture Framework (FEAF) [6], Department of Defense Architecture Framework (DoD AF) [7], Department of Treasury Architecture Framework (DoT AF) [8], Zachman Framework [9], The Open Group Architectural Framework (TOGAF) [10] and Generalized Enterprise Architecture Architecture and Methodology.(GERAM) [11].

Below we focus on Application domain characteristics, Prescriptions regarding AD organisation,and Relations to other frameworks, and outlines the corresponding values for each of the frameworks.

Framework	**Application domain char.**	**AD organisation**	**Rel. to other frameworks**
FEAF	*System type*: Enterprise *System scope:* Ent. in US Federal Gvt.	Organised according to a combination of 2 dimensions: *Perspective* and *Focus*, both explicitly enumerated	AD organising principle based on Zachman: Perspectives the same, but foci defined according NISTs' layered model
DoDAF	*System type:* Enterprise *System scope*: within DoD	Organised according to 3 explicitly specified views	Operational view corresponds roughly to the Planner+Owner perspectives in DoT AF. System view corrsep. roughly to DoT AF's Designer+Builder persp.
DoT AF	*System type:* Enterprise *System scope:* within DoT	Organised according to a combination of 2 dimensions: *Perspective* and *View*, both explicitly enumerated	AD organising principle based on Zachman: *Perspectives* the same, but *foci* are

Framework	Application domain char.	AD organisation	Rel. to other frameworks
			termed *view*, and specified specifically for TEAF. Specification of products taken from DoD AF.
Zachman	*System type:* Enterprise System scope: any enterprise	Organised according to a combination of 2 dimensions: *Perspective* and *Focus*, both explicitly enumerated	Used as a product organising principle in FEAF and TEAF
TOGAF	*System type:* Enterprise *System scope:* Ent. in US Federal Gvt.	No explicitly specified organisation of products. Prescribes development of an architecture repository to support the organisation of re-usable architectures. Provides a classification scheme for re-usable arch.	
GERAM	*System type:* Enterprise *System scope:* any enterprise	Organised according to 3 dimensions*: Life-cycle*, *genericity* and *view*. All dimensions explicitly enumerated	

4 Future work issues

We are shortly completing an initial version of the architecture framework ontology. The following challenges have to be faced:

- Ontological discrepancies between frameworks: When talking about architectures and descriptions, the various frameworks often use different definitions of common terms like view and perspective. In other cases different terms are used for the same architectural phenomenon. This makes it difficult to compare the various frameworks, and must be handled by the characteristic ontology
- Extending the use of the characteristics ontology:
  - While it is useful to be able to compare architecture frameworks in a systematic way, there is also a need to perform *assessment* of frameworks, e.g. evaluate the suitability of a particular framework to the problem at hand.
  - To support interoperability within and between enterprises, we need to be able to *interrelate architecture descriptions* created by different frameworks. In a model-based world this means mapping between metamodels, which is no trivial task.

5 References

1. Mili, H., et al., *Enterprise frameworks: issues and research directions.* Software-Practice & Experience, 2002. **32**(8): p. 801-831.
2. Martin, R. and E. Robertson, *A comparison of frameworks for enterprise architecture modeling*, in *Conceptual Modeling - Er 2003, Proceedings*. 2003, SPRINGER-VERLAG BERLIN: Berlin. p. 562-564.
3. IEEE, *IEEE Std 1471-2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. 2000, IEEE,.
4. Aagedal, J.Ø., et al., *Model-Based Architecture Framework, v 3.0*. 2003, SINTEF Telecom and Informatics. p. 68.
5. Ohren, O.P., *Rammeverk for beskrivelse av virksomhetsarkitektur [in Norwegian]*. 2003, SINTEF: Oslo. p. 61.
6. Chief Information Officers (CIO) Council, *Federal Enterprise Architecture Framework. V. 1.1*. 1999, Chief Information Officers Council (CIO Council): [Washington DC, USA].
7. Department of Defense Architecture Framework Working Group, *DoD Architecture Framework. Vol. I-II; Deskbook*. 2003.
8. Department of the Treasury CIO Council, *Treasury Enterprise Architecture Framework. V.1*. 2000, Department of the Treasury: [Washington DC, USA].
9. ZIFA, *The Zachman Institute for Framework Advancement.*
10. The Open Group, *The Open Group arcitectural framework (TOGAF), version 8. "Enterprise edition"*. 2002, The Open Group: Reading, UK and San Francisco, USA. p. Available on the web at http://www.opengroup.org/architecture/togaf8/.
11. IFIP-IFAC, *GERAM: Generalized Enterprise Architecture Architecture and Methodology. V. 1.6.3.* 1999, IFIP-IFAC Task Force on Architectures for Enterprise Integration.