Rich Hilliard
r.hilliard@computer.org
20 February 2011

Metamodels in 42010

Executive summary: The purpose of this note is to investigate the use of metamodels in IEEE 1471|ISO/IEC 42010. In the present draft, metamodels serve two roles: (1) to describe the conceptual framework or language of the standard [in 4.2], and (2) as a point for architecture frameworks to claim conformance to the standard [in 6.1].

Current definitions of "metamodel" are presented from the literature. Current architecture frameworks are examined for their use of metamodels. These insights are applied to use of "metamodel" in present 42010 draft.

It is concluded (1) these frameworks are not exemplary such that 42010 should follow their uses of the term; (2) "metamodel" is a suitable term to use in 4.2; and (3) the framework metamodel reflection requirement [6.1] is (i) not an onerous one on architecture frameworks aspiring to conformance with the standard; and (ii) is consistent with stated goals for current framework metamodels—although the actual practice in those frameworks lags behind their own stated goals.

What is a "metamodel"?

This section presents a few recent definitions.

Widely cited (*origin, unknown*):

> a **metamodel** is a model that defines the language for expressing a model.

OMG MOF [OMG, Meta Object Facility (MOF) Core Specification, version 2.0, formal/06-01-01]:

> A **metamodel** is a model used to model modeling itself. … A **metamodel** is also used to model arbitrary metadata (for example software configuration or requirements metadata).
>
> Metamodels provide a platform independent mechanism to specify the following:
>
> - The shared structure, syntax, and semantics of technology and tool frameworks as metamodels.
> - A shared programming model for any resultant metadata (using Java, IDL, etc.).
> - A shared interchange format (using XML).

From the ISO & IEEE software and systems engineering vocabulary <http://www.computer.org/sevocab/>:

> **metamodel. (1)** a logical information model that specifies the modeling elements used within another (or the same) modeling notation . . . **(4)** specification of the concepts, relationships and rules that are used to define a methodology

Does 42010 contain a metamodel?

ISO/IEC 42010 specifies requirements on architecture descriptions and other artifacts (Clauses 5, 6 and 7). Those requirements are expressed in terms of a conceptual model of architecture description (4.2).

Using the definitions cited above, that conceptual model of architecture description is depicted using a *metamodel* and its accompanying text:

- the 42010 metamodel is "a model that defines the language for expressing a model" [i.e. the *language* for expressing architecture descriptions which are *models* of architectures]
- the 42010 metamodel is "a logical information model that specifies the modeling elements used within another (or the same) modeling notation" [i.e., it specifies *modeling elements*: stakeholders, concerns, views, viewpoints, models, model kinds, correspondences, AD elements, etc., used in architecture descriptions and related artifacts (viewpoints, frameworks, ADLs)]
- the 42010 metamodel is a "specification of the concepts, relationships and rules that are used to define a methodology" [i.e., the *concepts* (enumerated under the previous bullet represented as UML classes), their *relationships* (shown using UML relationships (Relationship is the superclass of Association and Generalization) and *rules* (as specified in the class diagrams and elaborated in subsequent requirements)]

"Metamodel" as used in current architecture frameworks

This section is a brief look at use of **metamodel** in some recent frameworks. Representative quotes from each framework are followed by my observations at the end.

DODAF Meta Model (DM2):

> The data in a described architecture is defined according to the DoDAF Meta-model (DM2) concepts, associations, and attributes" [DoDAF 2.0, Volume 1, pdf page 11]

The purposes of the DM2 [DoDAF_Metamodel_Walkthru_2008-07-22.ppt]:

> The vocabulary for description and discourse about DoDAF models (formerly "products") and core process usage
>
> The basis for generation of the "physical" exchange specification for exchange of data between architecture tools and databases.

Another role for the DoDAF metamodel is to support discovery of architecture descriptions (ADs):

> Discovery metadata (i.e., the metadata that identifies a specific Architectural Description, its data, views, and usage) should be registered in DARS as soon as it is available to support discovery and enable federation." [DoDAF 2.0, Volume 1, 70]

MoDAF Metamodel (M3)

> The MoD Architecture Framework (MoDAF) Meta Model (M3) is the information model for MoDAF, defining the structure of the underlying architectural information that is presented in the views. The goal is that MODAF tools are "model-driven" - ie the views that are presented to the user are snapshots of underlying architectural data which is stored in the tool or in a repository. <www.mod.uk>

MoDAF uses "metamodel" and "reference model" as synonyms:

> The MoDAF Meta Model (M3) is the reference model that underpins MoDAF. It, defines the structure of the underlying architectural information that is presented in the MoDAF views. The goal is that MoDAF tools are 'model-driven'; ie, the views that are presented to the user are snapshots of underlying architectural data which is stored in the tool or in a repository. [MoDAF Meta Model, V1_0U]

NAF Metamodel (NMM)

NAF states a metamodel is "a model which describes a model" [NAF, 7.2.1]. According to NAF, the NMM "extends the UK MoDAF Meta Model" [NAF v3, 1.7.3]. Furthermore:

> The foundation of the NAF is the NATO Architecture Framework Metamodel. The metamodel defines the relationships between the different components of the framework. The NAF Metamodel uses the IEEE Std 1471 as a starting point. [NAF v3, 1.7.3]
>
> The metamodel is ***not*** … a conceptual data model or ontology – the intent is to capture the architectural meta elements and the relationships between them. The semantic classification of the architectural meta elements is captured in NAV-2. [NAF v3, 5.1.3]
>
> Architectures should be reliable, comparable, and integratable across NATO. Architectures must be built in compliance with the architecture metamodel. [NAF v3, 1.12]

TOGAF Metamodel

TOGAF describes a content metamodel:

> The TOGAF Architecture Development Method (ADM) provides a process lifecycle to create and manage architectures within an enterprise. At each phase within the ADM, a discussion of inputs, outputs, and steps describes a number of architectural work products or artifacts, such as process and application. The content metamodel provided here defines a formal structure for these terms to ensure consistency within the ADM and also to provide guidance for organizations that wish to implement their architecture within an architecture tool. <http://www.opengroup.org/architecture/togaf9-doc/arch/chap34.html>

TRAK Metamodel

> The TRAK Metamodel "Defines the stereotypes their attributes and the relationships between them. This provides the set of "things" from which a TRAK architecture description is constructed and how they are connected." [TRAK Metamodel, 1]

Observations

1. None of the above frameworks provide a definition of "metamodel". The quotes I have selected are my best attempts to get at what they might have in mind.
2. Using any of the definitions above, these metamodels appear to be incomplete, even when looked at from the perspectives of their intended purposes. (*See next few observations*)
3. The DoDAF metamodel (DM2) is meant to establish the "vocabulary for description and discourse about DoDAF models", and to provide "discovery metadata … that identifies a specific Architectural Description, its data, views, and usage", *yet DM2 provides no vocabulary for ADs, views and models*. Therefore, there are no rules for models or view, and no basis for interoperability, except at the lowest level of data commonality: "The data in a described architecture is defined according to the DoDAF Meta-model (DM2) concepts, associations, and attributes" [volume 1, 11]
4. DM2's stated purpose includes "vocabulary for description and discourse about … core *process* usage", *yet the metamodel has no vocabulary pertaining to the architecting process* discussed informally throughout the DoDAF text (which depends on numerous constructs including: architecture principles, analytics, composite views, dashboard views, fusion views, graphics views, reference models, viewpoints, configuration management, etc.
5. The MoDAF Metamodel (M3) seems similar in spirit to the DoDAF metamodel, but is clearer in articulating its intended limits: "defines the structure of the underlying architectural information that is *presented in the MODAF views*". This seems to imply that it does not pretend to be a full metamodel for "description and discourse about [MoDAF] models", to paraphrase the DoDAF quote above.
6. NAF's description of the NMM is a study in contradiction. It claims to "extend IEEE Std 1471" – presumably intending to say it extends the metamodel in that standard, yet the "architectural meta elements" of that standard are not reflected in NMM. Although NAF states that "Architecture views and subviews are standard ways to present aspects of an architecture, [and] users can define their own subviews to suit their own purposes." [NAF v3, 4.2.3], NMM provides no constructs for views, subviews, either as "standard ways" of expression or as "containers" for user-defined views "to suit their own purposes".
7. NAF uses its metamodel as a point of conformance: a "NAF-compliant architecture must also be fully NMM-compliant." [NAF v3, 4.3.3.1]
8. NMM seems to allow metamodel extension, but discourages it: "extensions to the metamodel impede exchangeability and understandability of NAF-based architectures" [NAF v3, 4.3.3.2]. Thus, trading off exchangeability for expressivity.
9. DM2, M3 and NMM "metamodels" are focused on domain data with names like: Performer, Resource Flow, Project, Organization, Location, Capability, Asset, Materiel, Platform, Facility, System, Node, Needline, etc. While these are certainly part of what we would expect in a framework metamodel, typically associated with the ontologies of one or more viewpoints, they are insufficient by

themselves. These might be more appropriately called "domain metadata models" rather than architecture framework metamodels.

10. In addition to views and viewpoints, TOGAF introduces a number of other architectural constructs: building blocks, catalogs, matrices, and diagrams. It is unclear whether these are "meta architectural elements" in the TOGAF metamodel.
11. TOGAF provides predefined extensions for specialized areas (governance, services, process modeling, etc.) First-class extension not defined.
12. All frameworks informally talk about principles, constraints, assumptions—nowhere reflected in their respective metamodels.
13. Aside from NMM, the frameworks do not offer provisions for metamodel extension, thereby making the (implicit) claim that their current ontologies are *fully adequate for all architecting within their domains*. This is surprising since none of them deal with the wide range of concerns that one would expect defense systems to confront frequently, including safety, security, reliability, etc. (DoDAF version 2.0 has added some security characteristics manually and provides guidance on using those characteristics in DoDAF models.)
14. The TRAK metamodel "defines the set of 'things' from which a TRAK architecture description is *constructed and how they are connected*." Insofar as TRAK ADs might be constructed with views, models, etc., using the connections required by 42010 of those "things", I would expect views, viewpoints, etc. reflected in the TRAK metamodel. As with the above frameworks one does not "construct" an AD from domain metadata elements: Actors, Organizations, Systems, etc.—the AD is a container for models of these and is "constructed" based on a structure of its own. If that structure aspires to conform with the standard, I would expect the reflection requirement to hold.

What is a "*framework metamodel*"?

The requirements on architecture frameworks (6.1) contain this text:

> When an architecture framework includes a framework metamodel, that metamodel shall reflect the provisions of the metamodel in 4.2.
>
> A metamodel *M1* reflects a metamodel *M0* if and only if all of the classes in *M0* occur in *M1*, and all class associations in *M0* occur in *M1* with the same roles and multiplicities.

The idea here is that to claim conformance to the standard, an architecture framework must build upon the concepts of the 42010 conceptual foundations, as embodied in the 42010 metamodel.

The phrase **framework metamodel** should be interpreted as a *metamodel of an architecture framework*. Applying the definitions above, that means:

- a framework metamodel is "a model that defines the language for expressing a model" [i.e., the language for expressing ADs constructed within the framework]

- a framework metmodel is "a logical information model that specifies the modeling elements used.." [i.e., specifying the constructs made available by the framework]
- a framework metamodel "specification of the concepts, relationships and rules that are used to define a methodology" [*exercise left to the reader*]

Are the "metamodels" above "framework metamodels" or something else?

As my observations above suggest, the "metamodels" associated with the frameworks I have examined are a mixed bag and only partially fulfill even their own stated purposes for the role of a metamodel as part of an architecture framework.

The frameworks seem to suffer several problems with respect to metamodels:

First, it is unclear from where their constructs derive their meanings. [This is a philosophical matter, more than a technical issue, and I won't discuss further here. Ask me over a beer.]

Second, the metamodels (hence their architectural ontologies) are limited and closed. The range of concerns expressible in these frameworks is limited, certainly inadequate for what is usually considered "architecting". Extension to allow new models to address the range of known system concerns is traded off for exchangeability. Perhaps that is OK because the only real role of these frameworks is bookkeeping a fixed set of interoperability characteristics between systems. How this bookkeeping ever came to be called "architecture" is a separate matter, out of scope of this note. IEEE 1471 made an explicit assumption that one could never enumerate all the system concerns that may be relevant to architecting, therefore viewpoints are extensible: each viewpoint can introduce new constructs to frame specific concerns.

Third, the metamodels presented are incomplete even when evaluated against their own stated purposes. DoDAF's DM2 aspires to provide a "vocabulary for description and discourse about DoDAF models", and I suspect several of the other frameworks would find sympathy in that phrase. However, none of the metamodel cover the range of constructs used in their own narratives about architecture, architecture description or architecting process: where are principles, constraints, assumptions, drivers, objectives, by-laws, viewpoints, building blocks and the many other concepts? They are not visible in the metamodels. As noted above, one does not "construct" an AD from domain metadata elements: Actors, Organizations, Systems, etc.—the AD is a container for models of these and is "constructed" based on a structure of its own. If that structure aspires to conform with the standard, the reflection requirement should hold.