Leading the Practice in Layered Enterprise Architecture

Simon Polovina[1], Mark von Rosing[2] and Georg Etzel[3]

[1] Conceptual Structures Research Group, Sheffield Hallam University, Sheffield, UK
[2] Global University Alliance, Château Du Grand Perray, La Bruère Sur Loir, France
[3] LEADing Practice ApS, Augsburg, Germany

S.Polovina@shu.ac.uk, mvr@globaluniversityalliance.org, ge@leadingpractice.com

Abstract. While Enterprise Architecture (EA) causes organisations to think, work and model in domains, there are inadequacies in such a waterfall approach. By restating domains as layers, i.e. LEAD (Layered Enterprise Architecture Design/Development) based on the LEAD Enterprise Ontology, EA performs better in enterprise layers and levels of abstraction. Through LEAD, the domain relationships are also better captured, hence leading the advancement of agile EA.

1 Introduction

There are multiple Enterprise Architecture (EA) framework and methods, each with their metamodels and specific approaches. Besides missing relevant objects within their meta models and thereby their concepts, more damaging is that most of them have a problematic way of thinking and working. This paper will illustrate that working ether in architectural domains, perhaps with a linear waterfall approach is counterproductive to the effort. We provide evidence that working with and across layers enables concurrent work within and across multiple domains, thus promoting an agile way of thinking and working around EA. Organisations can draw upon common best practices and leading practices to gain insight into how best to fulfil their value and purpose. Formal architectural views for business, information and technology perspectives assist these practices. The formal models are portrayed as enterprise ontology components to reduce misinterpretation, i.e. building blocks and metamodel views. Organisations can thus specialise their organisation knowledge according to their specific needs. Computer science and informatics contribute to the expressibility in these metamodels through its advances in ontology and semantics; together they capture the objects and relations that describe the interplay and effects of business in a formal, computable model [1, 2]. These objects and relations deliver generic EA patterns that any organisation can reuse in the fulfilment of that organisation's overall purpose. The organisation thus avoids "reinventing the wheel" which causes it to make mistakes or waste resources on rediscovering what is already known in modelling and architecting the enterprise.

2 Understanding the Architectural Layers in Organisations

The Enterprise Ontology, as well as the EA research that has been ongoing for over ten years, has identified that independent of size or industry all organisations have a common underlying structure [3]. Furthermore, from identifying the structures and the context in organisations, the following enterprise layers emerge [4]:

- **Business Layer**: Such as the value, competencies, processes, and services aspects;
- **Information Layer**: Such as the application systems, as well as the data components;
- **Technology Layer**: Such as the platform and infrastructure components

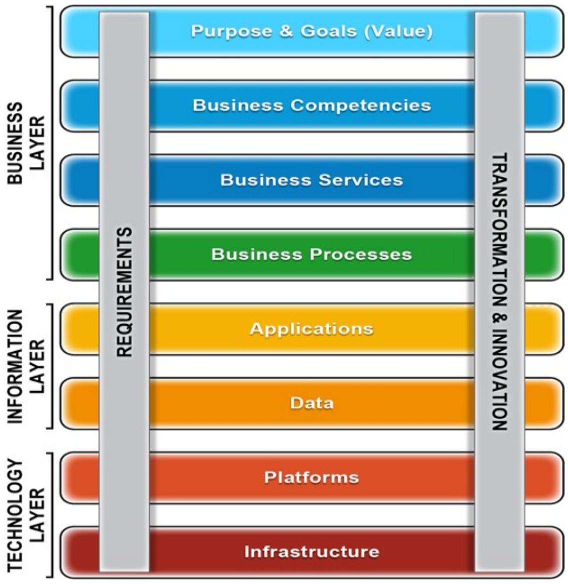

Fig. 1. Overview of the common enterprise layers

The organisation thus has to align its way of thinking with its way of working within and across all these perspectives. The Global University Alliance (GUA)[1], which is a non-profit body consisting of over 450 academics and researchers have developed and integrated each perspective into one holistic view, outlined in Figure 1, which outlines Layered Enterprise Architecture Development (LEAD) [2]. The layers were also used as the basis to develop the LEAD Enterprise Ontology [3, 5]. The enterprise standards body, LEADing Practice[2] has applied this layered enterprise structure as well as the LEAD Enterprise Ontology to develop standards as well as detail the most common enterprise concepts, i.e. meta-objects found across the business, information and technology layers [4].

3 The Meta-Object as relevant to the Enterprise Layers

The enterprise layers and sublayers are an abstraction that represents and considers the enterprise as a whole. For example, a disruptive force, strategy, plan, policy, measure

---

[1] www.globaluniversityalliance.org

[2] www.leadingpractice.com

or quality aspect is a part of the business layer, while the application systems and data aspects are a part of the Information layer. Using the LEAD Enterprise Ontology and applying it to the newest research from GUA[3], the most relevant objects class types, have been identified according to their context a very precise affiliation to a specific layer. With it, all the underlying instance types that are relevant for enterprise architecture, according to their context, can relate to a specific layer. Figure 2 gives an overview of the most common meta-objects found within the enterprise architectural layers.

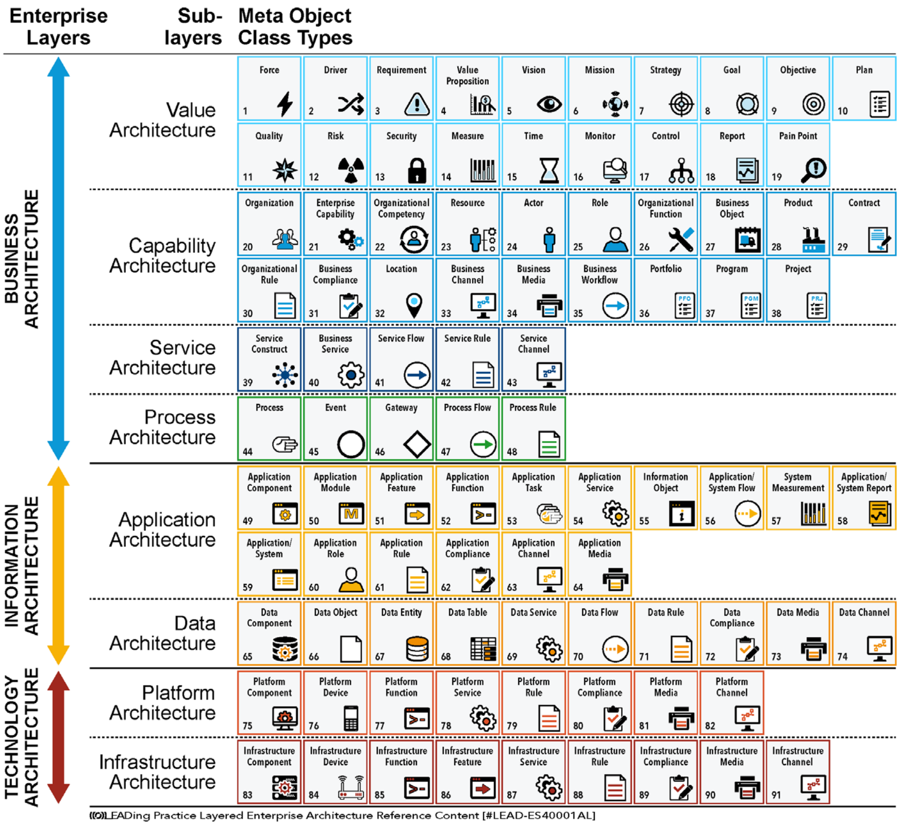

Fig. 2. Overview of the Enterprise Layered Meta-objects.

The objects purpose, goal, aim, target, objective and context arise from their affiliation to a specific layer and sub-layer. While each meta-object has multiple semantic relations across the layers, based on their context, they have an explicit affiliation to a specific layer. This affiliation is a set association which does not change with time or the semantic relationships the object has with other objects, within or across the layers. Therefore one of the findings of GUA's ongoing research and evidenced by LEAD was

[3] www.globaluniversityalliance.org/research/enterprise-architecture/, www.globaluniversityalliance.org/research/enterprise-ontology/

the identification of the objects and their semantic relationships, which lead to the development of Enterprise Ontology metamodels [3]. The categorisation of the class type objects according to their relevant layers and subjects enables practitioners to use them directly, underpinned by EA principles appropriate for handling the different tasks, correlations, relationships and connections [3–5]. The meta-objects not only have one relationship but multiple interaction points within one layer and across the layers. Figure 2 could be the LEAD Periodic Table analogous to the periodic table in chemistry, as it gives an overview of the enterprise layers and sub-layers, along with their affiliated enterprise meta-objects (as its 'elements') with the object's specific notation symbols. The semantic relations interrelate these elements into allowable combinations, analogous to compounds in chemistry and the metamodel in LEAD.

4 The Agile Enterprise Architecture Way of Working

LEAD's principal principle that makes it distinct from other EA ways of working is that it extends the EA domains (i.e. Business, Information, and Technology) through layers. By thinking in layers (and sublayers) as the frame of reference and the semantic relations as the pathways, any metaobject can be a starting point navigated to or from other metaobjects within and between various layers. This agility extends into levels as an instance, i.e. object of a metaobject specialises into stereotypes, types, subtypes, decompositions or compositions. Thus the layers, sublayers, levels and sublayers enable organisations, framework developers and standard bodies alike to use these researched and validated objects and definitions to develop models where the ontology part is not 'self- or home-made' but on a common, reusable ontology that the LEAD Enterprise Ontology depicts, highlighted by figure 2. One of the major challenges facing the framework, method and approaches in the market today, is overcoming a presently fragmented way of thinking, working and modelling around the myriad EA concepts that currently exist. Business frameworks, methods, approaches and concepts like TOGAF, ITIL, BPMN, CMMN and others all have their own-defined concepts, objects, relations and vocabulary. The resulting conflicts extend to within the standards of a standards organisation, i.e. ArchiMate and TOGAF from The Open Group or VDML, BPMN and DMN from OMG [6].

Figure 3 illustrates how an Enterprise Architect (EAt) would work in an agile way across the layers. As another analogy, a meta-object acts as a building block that can be picked up and worked with as needed. Thinking and working with these elements enables full agility using those elements that are 'fit for purpose', i.e. relevant to the purpose at hand. How and where the EAt could use the 'building block' is defined by its layer, and the other building blocks the EAt wants to relate the metaobject's object instance including it sub-layers through the metaobject's semantic relations. The requirements, the capability, resources, tasks and information of the specific 'building block' would matter in relating it to other objects.

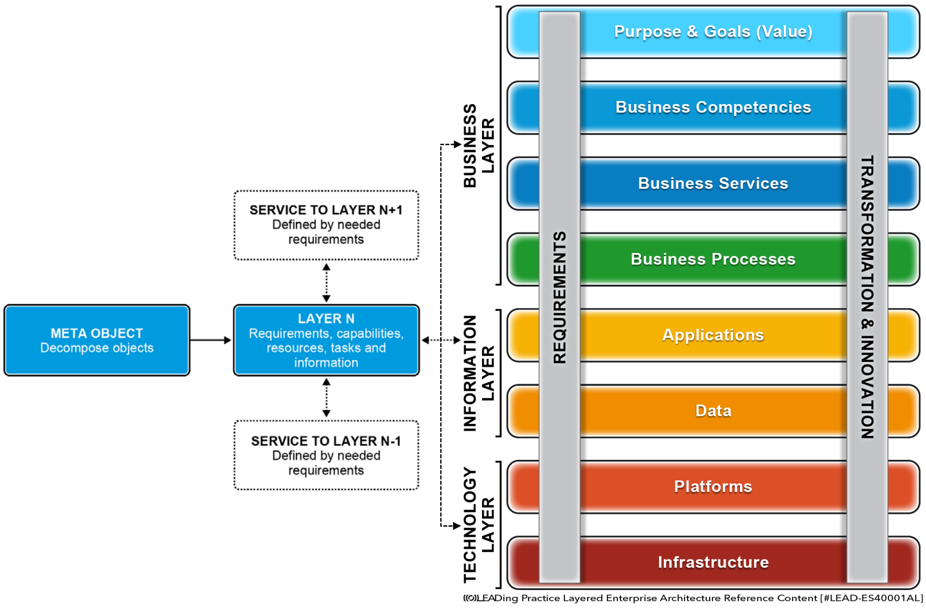

Fig. 3. The Agile EAt's Way of Working

This figure highlights that a layer provides a set of functions and tasks and thereby services to its upper or lower layer. The EAt thus relates and structures the objects across layers, touching upon services between them. In turn, the upper layer draws on the lower layer's services to achieve its service. The *n'th* layer (±1) therefore acts as a service requester or provider since its ether gives input or uses the services provided by its lower layer. The semantic relations are these services, shown by the following examples that run within and across the enterprise layers. The (#) symbol is a reference to the meta-object numbers found in figure 2:

- Strategy (#7), Goal (#8) and Objective (#9) define the direction of the Organisation (#20), thereby the specific Organisational Function (#26)
- An Organisational Function creates and works with a Resource (#23) and a business Role (#25) to execute the defined strategies, objectives and goals
- A Value Proposition (#4) influences a Plan (#10) around an Organisational Function (#26)
- An Organisational Function (#26) creates a Business Service (#36)
- An Organisational Function (#26) is executed as a task within a business Process (#44)
- An Organisational Function (#26) can partly or fully be automated as an Application Function (#52) within an Application Components (#49) and Application Module (#50)
- A Business Service (#40) can partly or fully be automated as an Application Service (#50)

- A business Process (#44) can partly or fully be automated as an Application Task (#49)
- An Organisational Function (#26) and Process (#44) have a Business Role (#24)
- When automating an Organisational Function (#26), Process (#44) or Business Service (#36) within an Application System (#55), there will be an Application Role (#60).
- A business Role (#24) as well as an Application Role (#60) work with both Business Objects (#27) as well as an Information Object (#55)
- The Control (#17) of an Organisational Function (#26), Business Role (#24), Process (#44) and or Business Service (#36) can be ensured through an Organisational Rule (#30), i.e. policies, acts, procedures and standards
- An Organisational Rule (#30) is also set in place to ensure Quality (#11), lower Risk (#12) and ensure Security (#13)
- An Organisation (#20) relates an Organisational Rule (#30) throughout the enterprise, for example, when applying it to a business Process (#44) and Business Service (#36), these would become a specific Process Rule (#48) and Service Rule (#42)
- All the rules can also be related to the Information Object (#55) and Data Object (#66) and can be automated into an Application Rule (#61), Platform Rules (#79) or Infrastructure Rule (#88).
- A Platform Device (#76), e.g. smartphone, tablet, or scanner are used by a business Role (#24) partially or fully automated by Application Role (#60) to support the functions, processes and services

These example semantic relations only illustrate some of the possible requirements, relations and services between the objects and thereby the layers. Nonetheless, an EAt could choose any object and work in an agile way across the layers and levels to integrate effortlessly across the Enterprise Layers, which also includes all the available semantic relations.

5 The Agile Enterprise Architecture Way of Modelling

LEAD also provides artefacts. An artefact is a user-friendly view that encapsulates the richness of the relevant objects and semantic relations in the layers[4]. An EAt uses three types of artefacts populated to present information succinctly to decision-makers, e.g. the organisation's stakeholders, management or leaders:

1. *Artefacts Map*: A Map may represent subtype, decomposed or composed instances of the relevant meta-objects. Examples of a Map are a process map, a data map or an application map populated by these instance levels. It is often in the form of a list (or lists within lists) that can be in a simple set of rows, or as a catalogue, and has the purpose of building an inventory or index list of the instances within the different layers, e.g. Business Layer, Information Layer or Technology Layer.

[4] Modellers often refer to artefacts as models and Engineers refer to them as templates

2. *Artefact Matrix*: A Matrix may likewise represent subtype, decomposed or composed instances of the relevant meta-objects. Examples include a process/rule matrix that interlinks processes and rules or a platform service/data matrix. A matrix typically consists of objects in rows and objects in columns and instance levels as the cross product between the rows and columns. The Matrix allows the EAt to relate in an agile way the unfamiliar to the familiar objects in the different layers usually through the form of a table or chart, e.g. rows and columns in a matrix, thereby outlining their direct connection between instances of row and column objects, underpinned by the common pattern of the relevant meta-objects and semantic relations.
3. *Artefact Model*: A Model is a representation that shows the relationship and the interconnection between instances graphically. It may show the semantic relations directly or, more often, encapsulate them in terms that are more digestible to an EAt or familiar to a decision-maker. Examples include a BPMN process view or a UML user sequence diagram. The model is a graphical representation, view or illustration of levels from the layers to represent an aspect of an enterprise (e.g. business, information or technology), using a specific set of rules that the view or model has defined, e.g. to accord with the OMG's BPMN or UML standard, or the Open Group's Archimate standard. A Model may draw on a Map or a Matrix (or both), to enable complex information to be communicated more easily to decision-makers through an expressive graphical depiction.

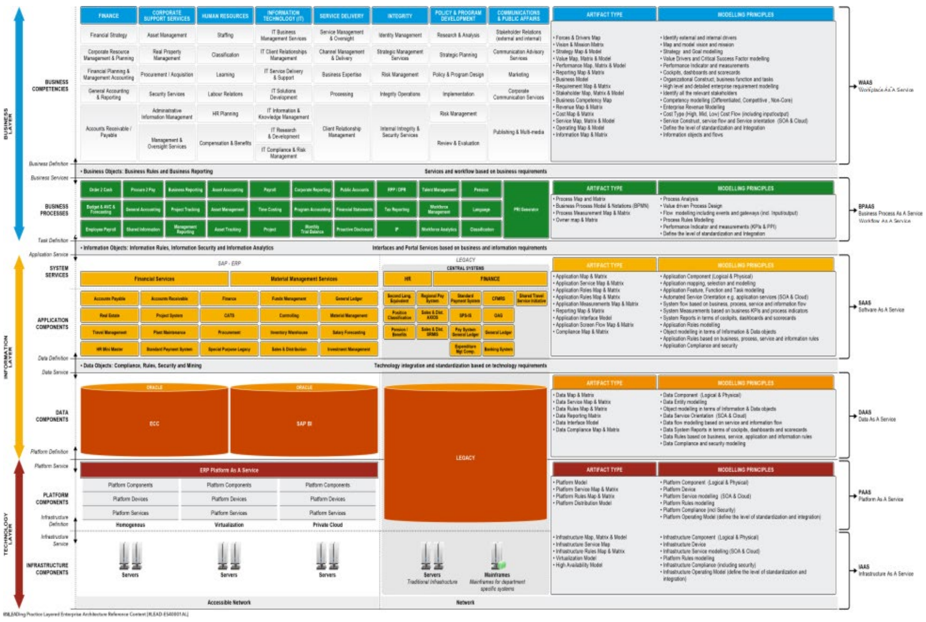

Fig. 4. The Agile EA Way of Modelling: Integrated Artefact for Cloud-Based Architecture

Figure 4 highlights an example of a commonly used artefact populated for Cloud-Based Architecture. Elements of Maps, Matrices and Models appear in this integrated artefact.

6 Concluding Remarks

By thinking, working and modelling with layers and levels, LEAD provides an agile way to capture an EA most expressively. More than 70% of all IT projects suffer and fail in being on-time, quality and budget [7]. Ineffective ontology, semantics modelling and architecture principles have contributed to this issue [3–6]. IT blueprint, development, implementation and maintenance groups and departments would perform better if they referred to a common way of working, thinking and modelling that LEAD epitomises. LEAD represents domains as layers, sub-layers, levels and sub-levels with user-friendly artefacts that capture this conceptual structure. Through modelling in layers–i.e. LEAD–the relationships within and between each domain are better captured, hence leading the advancement of Enterprise Modelling and Enterprise Ontologies in EA.

References

1. Laurier W, Kiehn J, Polovina S (2018) REA2: A unified formalisation of the resource-event-agent ontology. Appl Ontol 13:201–224. https://doi.org/10.3233/AO-180198
2. Andrews S, Polovina S (2018) Exploring, Reasoning with and Validating Directed Graphs by Applying Formal Concept Analysis to Conceptual Graphs. In: Croitoru M, Marquis P, Rudolph S, Stapleton G (eds) Graph Structures for Knowledge Representation and Reasoning. Springer International Publishing, Cham, pp 3–28
3. von Rosing M, Laurier W (2016) An Introduction to the Business Ontology. Int J Concept Struct Smart Appl 3:20–41. https://doi.org/10.4018/ijcssa.2015010102
4. von Rosing M, von Scheel H (2016) Using the Business Ontology to Develop Enterprise Standards. Int J Concept Struct Smart Appl 4:48–70. https://doi.org/10.4018/ijcssa.2016010103
5. Polovina S, von Rosing M, Laurier W, Etzel G (2019) Enhancing layered enterprise architecture development through conceptual structures. In: Endres D, Alam M, Sotropa D (eds) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer International Publishing, Cham, pp 146–159
6. Laurier W, von Rosing M (2017) Special Issue on Leading Enterprise Standards, Theories, and Practices Journal Article | IGI Global. Int J Concept Struct Smart Appl 5:preface
7. How to beat the transformation odds | McKinsey. https://www.mckinsey.com/business-functions/organization/our-insights/how-to-beat-the-transformation-odds. Accessed 24 Feb 2020