The four domains

TOGAF defines four high-level domains, covering the business, data, application and technology architectures. These four domains help structure much of the documentation – for example, five chapters describe the development of these domains in the ADM (Phase B, Phase C, and Phase D – note that there are three chapters covering Phase C, an overview, then Data and Application).

The Architecture Content Framework, which provides a pictorial overview of the EA metamodel, is based on the Phases of the ADM. A large proportion of the time and effort spent by the EA team is taken up developing the architectures, and this is shown graphically with the four domains dominating the central section of the framework, as shown in Figure 34-5: Detailed Representation of the Content Metamodel:

The same structure of the content framework is continued throughout the documentation of the metamodel (although, rather confusingly, the background colour for each domain changes for later diagrams). These more detailed diagrams are:

- Figure 34-7: Content Metamodel with Extension: this shows the core and extension entities, grouped into the four domains
- Figure 34-8: Relationships between Entities in the Full Metamodel: this shows the core and extension entities, plus their relationships, and again it is grouped by the four domains.
Logical and Physical Components

The definitions of items in the metamodel are generally not that difficult to grasp, but there are six components prefixed by Logical or Physical that sometimes cause problems, so we will explain these components here.

The core metamodel includes Application Component, which is a very general term and I think it was therefore intended to literally cover all types of application component. Application Component doesn’t make a distinction between a type of class of application, an actual application, a self-contained module within an application, or any other unit of functional software.

It’s helpful to bear in mind the TOGAF definition of Application – “a deployed and operational IT system that supports business functions and services; for example, a payroll service, or other deployable component of an enterprise.” Applications are referred to as Application Software, and the TRM recognises two categories of Application Software – Business Applications and Infrastructure Applications. [See Section 43.3.3 of Application – “a deployed and operational IT system that supports business functions and services; for example, a payroll service, or other deployable component of an enterprise.” Applications are referred to as Application Software, and the TRM recognises two categories of Application Software – Business Applications and Infrastructure Applications.]

Now in Section 34.4.5 Infrastructure Consolidation Extensions, “Application Components are extended to include Logical Application Components (a class of application) and Physical Application Components (an actual application)”. Here are the definitions:

- Logical Application Component: “An encapsulation of application functionality that is independent of a particular implementation. For example, the classification of all purchase request processing applications implemented in an enterprise.”
- Physical Application Component: “An application, application module, application service, or other deployable component of functionality. For example, a configured and deployed instance of a Commercial Off The Shelf (COTS) Enterprise Resource Planning (ERP) supply chain management application.”

You might consider renaming these: Logical Application might be better named “Type of Application” or “Application Category”; while Physical Application might be simply Application. In practice, you might also want to explicitly have sub-types for application, application module, and application service.

Now just to complicate things a bit more, the colour-coding in Figure 34.15: Infrastructure Consolidation Extensions, Changes to Metamodel suggests that the core metamodel Application Component and Logical Application Component are identical.

The Infrastructure Consolidation Extensions also extend Technology Components to include Logical Technology Components (a class of technology product) and Physical Technology Components (an actual technology product).

It’s also worth mentioning that the Technical Reference Model (TRM) also has similar components but are distinct from the technology components that are sometimes referred to as Components for requirement and modules for solution. Physical Technology Components are analogous to Logical Application Components and are analogous to logical data components, databases, registries, repositories, schemas, and other techniques of segmenting data.

So, Application Component in the metamodel often only hints at what you will need. In practice, you might also want to explicitly have sub-types for application, application module, and application service.

There are many times when it is useful to distinguish between similar components – for example, an application, module or service of a process, procedure and transaction. In these situations, you will need to create a metamodel that covers all your needs; the TOGAF metamodel often only hints at what you will need.
Core Architecture Artifacts

TOGAF also lists some core artifacts (in section 34.3.2 Core Architecture Artifacts). The architecture artifacts included in TOGAF are discussed in a lot more detail in Chapter 35, and you should go here for a detailed description of each artifact and the information that it contains.

The key point here is that each artifact is created from information that is based on the EA metamodel. For example, an Interface Catalog includes the Logical Application Component and Physical Application Component, and the relationship Application communicates with Application.

A lot of work has been put into the TOGAF documentation to map the EA metamodel to the listed artifacts. This emphasizes the fact that EA teams need to produce a wide range of different artifacts to meet the needs of all stakeholders. You will often need to present the same information in different ways. If you like, the metamodel structures information about the enterprise architecture, its components and their relationships so that you can easily produce artifacts that present and communicate information effectively to stakeholders.

Once again, the set of artifacts listed in TOGAF will not necessarily be the ones that you want to create! Customization is key. Think of the artifacts in TOGAF as examples, rather than prescribed outputs that you must produce.

Also bear in mind that there are many EA artifacts that are not listed or described in the TOGAF documentation. And TOGAF only covers three types of artifacts (catalogs, matrices, and diagrams).

However, the link between the EA metamodel and presenting and communicating information through artifacts is important. It is a useful exercise to create your own version of the following diagram (Figure 35-3: Artifacts Associated with the Core Content Metamodel and Extensions), to provide a quick and simple overview of the artifacts that you are planning to use, and when you are likely to use them. There are more sophisticated frameworks to manage these links, but as with much of the TOGAF material, the general concept is useful.

Key points:
- The four domains—business, data, application and technology—permeate the TOGAF metamodel, and you may want to divide your own metamodel using additional domains or sub-domains.
- TOGAF refers to some components as “logical” or “physical”; these terms are confusing. Adapt your metamodel so that it uses clear labels of the components you are describing.
- The metamodel structures and organizes the information that you capture as you analyze and develop architectures. Make sure that you define artifacts that present and communicate this information effectively to stakeholders.