

Topic: Frameworks

Date: April 2012

Author: Lawrence Wilkes

Establishing a Reference Framework. An Overview.

In any given domain, a Reference Framework (RF) provides a common backplane for consistency, collaboration, sharing, and reuse. Without a consistent framework, the domain in question will remain an interesting concept but deliver suboptimal business value. With an appropriate RF the work of individual projects, programs, divisions and partners will be coordinated with just enough formality to ensure that the many moving parts can fit together when and were needed.

A RF is needed to:

- Identify those things that need to be common
- Create consistency where needed
- Indicate where individual projects can diverge from the RF where appropriate
- Provide a structured approach to managing standards, policies, patterns in order to deliver on objectives

This research note provides an overview of the process for the creation and evolution of a RF.

Reference ‘Things’

A Reference Architecture (RA) “should” provide a blueprint or template architecture that can be reused by others wishing to adopt a similar solution. A Reference Model (RM) should explain the concepts and relationships that underlie the RA. At Everware-CBDI we then use the term Reference Framework (RF) as a container for both as shown in figure 1. Reference architectures, models and frameworks help for example to provide a consistent understanding of topics such as [Cloud Computing](#) or [Enterprise Mobility](#).

Unfortunately, such formality is often absent from the various reference architectures, models and frameworks that have been published in many domains; these frequently mix elements of architecture, model, process and organization and then apply one of the terms seemingly at random.

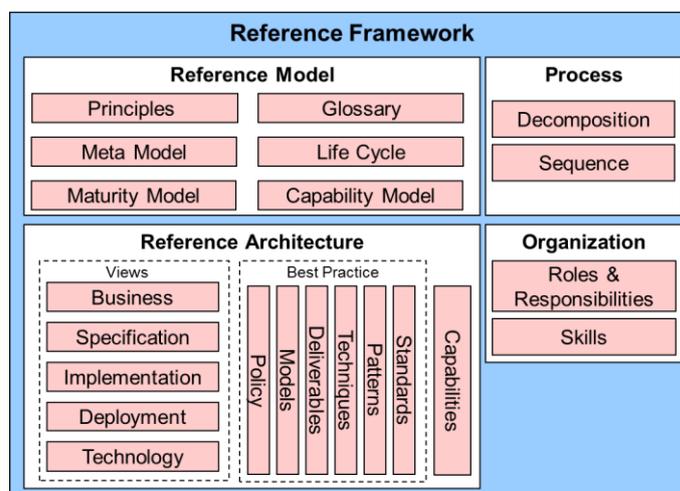


Figure 1 – Generalized Reference Framework

In developing the [CBDI-Service Architecture and Engineering Reference Framework](#) (SAE) in support of SOA (Service Oriented Architecture) Everware-CBDI separated out these various parts along the lines shown in figure 1. We developed a detailed [RA for SOA](#) and a RM for SOA, with particular emphasis on a rich and detailed [Meta Model for SOA](#) and a [Maturity Model for SOA](#). We also developed a detailed process and task decomposition for SOA activities.

But the RF is easily generalized, where the various elements could be applied to any domain. The benefit of this approach is that elements of the framework can then be mapped to each other in different ways to support alternative perspectives such as different usage or adoption scenarios, or the viewpoint of an individual participant or organization.

Elements of the Reference Framework

An overview of the elements in the RF is provided in table 1. They are discussed in more detail later in the note.

Model	Content	Relationships
Reference Model	<p>Defines the concepts and principles</p> <p>Defines the Life Cycle for the key deliverables or assets.</p> <p>Defines a maturity model against which an organization can benchmark itself</p> <p>Defines the capabilities an organization should possess in order to be successful in the domain</p>	<p>Provides the common vocabulary that underlies the Reference Framework.</p>
Reference Architecture	<p>Defines the architectural views at different levels of abstraction.</p> <p>Maps best practice to each of the views.</p> <p>Identifies the deliverables for each view, and the policies, patterns and standards that apply to each view.</p>	<p>Architectural Views can be expressed in terms of the meta model which defines the key concepts, deliverables, and assets.</p> <p>e.g. the subject of a policy should be a concept in the meta model</p>
Reference Process	<p>Defines the decomposition and sequencing of activities that take place relative to the domain</p>	<p>Process model should reflect how the life cycle state changes of the key deliverables or assets are accomplished, and how the deliverables or assets will be transformed from one architecture view to another</p>
Organization	<p>Defines the roles and responsibilities of the participants in the domain, and the skills they require</p>	<p>Organization defines the roles and responsibilities which regards to the key deliverables and assets, as well as the participation in the process</p>

Table 1 – Elements of the Reference Framework

The extent to which each of these elements is detailed will depend on the requirements and objectives. See the later section on “Framework Assessment and Requirements”.

These elements are not isolated but related to each other as highlighted. The real value is often in the mapping that can be made across these elements. Hence focusing on only one element may be less productive in the long term if the other elements are not addressed at some point.

Reference Framework Process

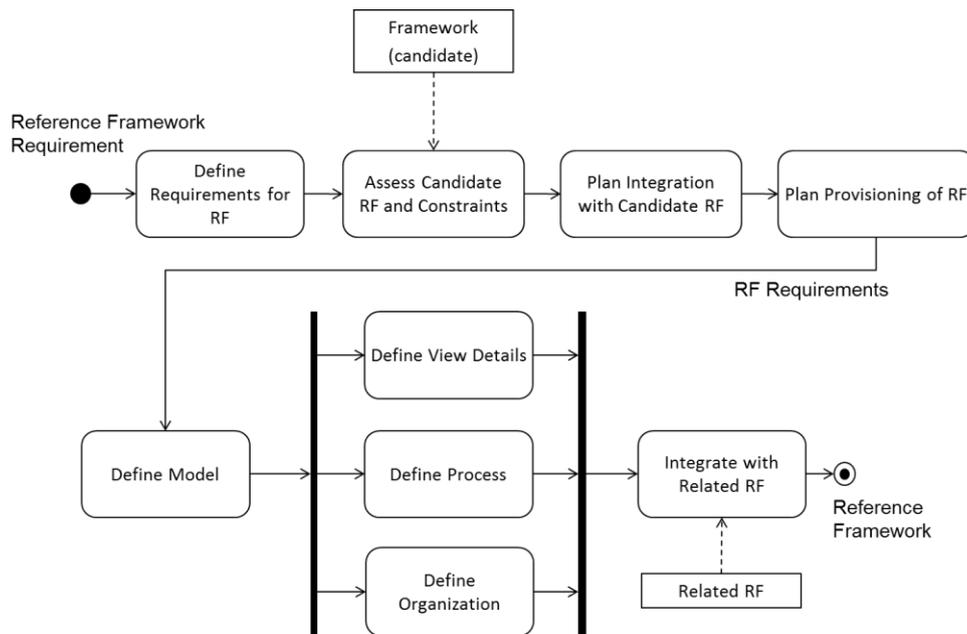


Figure 2 – Establish Reference Framework Process

Figure 2 illustrates a high level activity diagram of the process to establish a RF. Activities include,

- The activity commences by establishing the requirements for the framework, because a) it is likely most enterprises will already have existing frameworks and b) the creation of the RF is going to be an evolutionary process over several phases of the Adoption Roadmap.
- Once requirements are understood, candidate frameworks can be assessed and provisioned if suitable.
- The framework can be refined, or defined if starting from scratch, through the following activities
 - Define Model is a crucial foundational activity that establishes the vocabulary and things that need to be managed – the key deliverables and assets.
 - The Define View Details task then populates the details of layers, deliverables, policies, patterns etc together with parallel tasks that define the contract types and architectural assets.
 - Define Process establishes the activities that produce the key deliverables and assets
 - Define Organization determines the roles and responsibilities for the various participants in the domain
 - Finally there is a task to integrate the RF with other existing or planned frameworks because SOA is not a standalone concept.

The process is likely to be iterative. For example, you are unlikely to define the complete meta model before defining the architectural views. Much of the meta model will be discovered or refined whilst defining the views.

Again, the amount of effort spent on each of these activities will depend on the requirements and objectives. See the following section.

Framework Assessment and Requirements

Often it is not necessary for an enterprise to invent a green field RF. There are various framework initiatives that are in varying stages of maturity and have varying characteristics. It is quite likely that enterprises will

wish to integrate with pre-existing frameworks, and also possibly coordinate with other RFs. These may also include fragments of de facto frameworks for the given domain.

The place to start evolving a framework is to articulate the business value that may be derived from the RF in order to support decisions relating to collaboration and sharing. This is essential in order to justify the investment.

An understanding of the objectives for establishing an RF will help to drive the RF activity and perhaps focus the work on specific elements. Ask some straightforward questions to establish the purpose of the RF, for example,

1. What things is the organization trying to govern/control/manage using the RF? For example,
 - a. Deliverables and Assets? The life cycle for deliverables and assets isn't clear, and prevents adequate governance. There is no common schema for deliverables or assets.
 - b. Process? There is no repeatable process. It cannot be measured or governed in a consistent way.
 - c. Roles and responsibilities? People are unclear as to what their responsibilities are.
2. What problems are you trying to overcome? For example
 - a. Lack of understanding? It is a new domain that most are not familiar with and needs explaining
 - b. Lack of consistency? People understand the domain, but everyone does their own thing.
 - c. Lack of productivity? People don't have access to best practice, templates or other assets they can reuse. People reinvent the wheel.
 - d. Lack of governance? Policies are not clear, or difficult to apply

Existing Frameworks

An organization may already use one or more frameworks and they need to be reviewed to assess:

- Quality and completeness of existing models and tooling
- Compliance with regulatory requirements

Existing architectural frameworks may already contain extensive details of existing assets and provide valuable input to the RF. Some elements of existing frameworks may be immediately useful, particularly data architectures and logical models, application asset registers and integration schema definitions. However there will usually be differences and gaps in the underlying meta model that will render the information less than complete.

Enterprise Architecture (EA) frameworks such as TOGAF, DoDAF, MoDAF, FEA, etc., may provide a good starting point. These will often contain a high level cross section of many of the elements of the RF, but not at sufficient detail for a specific domain. Also, by their nature they may be more abstract than is required for a specific domain, where the vocabulary needs to be domain specific.

Define Reference Model

Conventional enterprise architecture describes an information system in terms of structural properties of the system. The architecture identifies components, building blocks, standards, policies and products which form the basis for planning and guiding systems delivery.

Not surprisingly new domains can introduce change to the structural properties. There are new and different building blocks, standards etc. These don't necessarily replace the existing properties, mostly they complement and extend. However there are also areas where fundamental differences apply, for example in

areas such as scoping and applicability, security models, reuse policies and so on, and we need a new model that combines the old and new in a cohesive whole.

Tasks include

- **Define Principles and Architectural Style.** Principles are not inherent to any technology, product, platform or tool. They require enterprise specific architectural and engineering thinking to utilize them. Most projects will conform with a subset of the core principles. For example, in SOA most are attempting to address some level of loose coupling because this is the obvious place that agility may be achieved.
- **Define Meta Model.** A meta model defines the rules for building a model of a business or software -- or anything else for that matter. The meta model for a domain is vital because it assists architects to define the domain concepts, establish consistent terminology and facilitate elimination of redundant concepts
- **Define Life Cycle.** An understanding of the life cycle helps an organization to better organize the provisioning and deployment of key assets in the domain. Key assets should be traceable, manageable and governed from the time they are planned through to retirement.
- **Define Maturity Model.** An organization may already have adopted more general maturity model frameworks such as [CMMI](#). However it is also useful to map the RF elements of the specific domain in terms of maturity levels so that an organization can assess where it is today in terms of capabilities to support the domain, and to set roadmap targets in terms of where it want to be in future.
- **Define Capability Model.** A [capability model](#) details the capabilities required by an organization in order to be successful in the given domain. These may be capabilities possessed by, or provided by people, computer systems, or 3rd parties. The capabilities should be implementation independent. That is the model should express what capabilities are required and their dependencies, who or what might possess them, but not how they are currently implemented.

Define Assets

Assets are the key deliverables that an enterprise wishes to track and manage. They will reflect concepts in the meta model. In addition the relationships (dependencies) between assets are also important to manage. In modern agile environments, there are typically many moving parts, which is what can potentially deliver greater agility. Hence identifying them and showing how they will be governed as they move through the life cycle is essential. For example to help resolve issues with versioning and impact analysis in complex component- and service-based systems. It is therefore absolutely critical that asset inter-dependency is:

1. subject to architectural decision making to create integrity units or clusters of assets that have a high level of internal cohesion and a low level of external dependency and
2. managed throughout the life cycle so that architectural decisions are not compromised, reducing the planned agility characteristics.

To manage the domain therefore we require a framework that defines and manages assets throughout the entire life cycle. Tasks here include,

- **Define Asset Types and Schemas.** For the key assets that are going to be tracked and managed.
- **Define Publishing Policies.** It is important that a consistent approach is adopted to publishing the availability of assets, so that providing projects know what should be published and when, and consuming projects know what is available.
- **Define Version and Release Management Policies.** Many initiatives have foundered on inadequate version and release management.

Define View Details

One of the defining characteristics of any methodology is the structure used to capture the relevant aspects or perspectives of a system, whatever system that may be – business, information system, hardware, or what have you. The RF includes five views – Business, Specification, Implementation, Deployment, and Technology. These views comprise a consistent level of abstraction for deliverable artifacts that relate to distinct set of stakeholders. This provides an effective mechanism for grouping practices, deliverables, policies, etc., based on a particular stakeholder view or perspective.

The framework is organized by view, and within each view there are key architectural elements that together provide a coherent approach.

To establish the framework each of these architectural elements must be defined and guidance provided sufficient for the purpose of the specific enterprise, to guide or govern appropriate levels of consistency.

Tasks include

- **Define Views.** There is a fairly high level of consensus in the industry about architectural views. Most architectural frameworks share a similar approach to creating stakeholder perspectives, however unsurprisingly there is wide divergence on naming. In defining the views considerations will be made relating to naming and consistency with existing frameworks and asset stores, in order to sensibly integrate with other initiatives.
- **Detail Architecture Layers.** The use of layers is a common architectural construct to organize assets by type. Everware-CBDI suggest that layers should specialize behaviors that maximize opportunity for reuse of layer contents as well as provide commonality of patterns, policies etc. A key purpose of a common approach to layering is to establish maximum opportunity for cross organizational or industry-wide sharing and reuse. If the layering is common, then firstly framework elements can be shared, but equally delivered instances of assets can be shared too. Equally if different approaches to layering are adopted then many opportunities for sharing and consistency will probably be lost.
- **Detail Reference Architecture Policies.** The purpose of the RF is to provide guidance on how key aspects of architecture should be implemented in a consistent manner in order to ensure the objectives of the domain are met. In practice certain aspects of the architecture will be mandatory within an enterprise and implemented in policy. Others will be recommendations for best practice and will be attributed as guidelines.
- **Detail Models.** Create detailed guidance on models that may be used to establish architecture and deliverables.
- **Detail Patterns.** Patterns are the primary way the principles will be implemented. If principles are the theoretical objective and style, patterns are the documentation of practical knowledge that is relevant to the problem. In practice many patterns will be available in the general industry domain, but enterprises will customize and extend them to make them relevant to the enterprise task.
- **Define Contract Types.** Business and IT agility requires the loose coupling, abstraction, and virtualization of assets. Therefore, assembling a reliable solution from these assets is highly dependent upon the existence of a contract between provider and consumer. We therefore need different contracts that create the necessary level of formality needed to establish the layer separation that delivers inherent business agility, and ensures the outcomes desired by various participants. TNot surprisingly we have three closely related subtasks as shown in table 7.

Define Reference Process

The reference process should provide a consistent view of the activities required to plan, architect, enable, deliver and manage assets, services and solutions relevant to the domain. The reference process shows how the necessary state changes to the key deliverables and assets will be achieved.

The reference process may be decomposed or organized along the following lines,

- **Discipline:** A Discipline is defined as a significant competency which has the ability to perform one or more process units.
- **Process Unit:** Each Discipline is comprised of one or more Process Units - a set of (one or more) tasks that can be performed by a Discipline. Each Process Unit forms a discipline “service” with a clear objective and set of possible inputs and outputs. A Process Unit provides a way of chunking a Discipline into reusable pieces. The outcome of a Process Unit is normally a key deliverable or asset.
- **Task:** A Process Unit consists of one or more Tasks - The finest-grained work unit that appears in a project schedule, to which resources are assigned.
- **Technique:** Each Task can be performed by following a Technique - a special procedure for performing a task, or group of tasks
- **Work Package:** One or more Process Units or Tasks may be grouped into a work package - a subset of a project that can be assigned to a specific party for execution.

Define Organization

A discipline is performed by roles which have the ability to execute the process units and tasks defined for that discipline or a work package. While some companies might establish a team that corresponds to a Discipline, others organizations might design more fluid organization structures.

To perform these tasks, roles must possess the appropriate skills.

Responsibility for key deliverables and assets can also be assigned to roles. A RAEW matrix may be appropriate for this – setting out which roles have Responsibility, Authority, possess the Expertise, or do the Work.

Related Processes and Tasks

The RF should be more than a pretty picture or report that is read once by people and then filed away.

Ideally it should be a living document that provides a growing body of knowledge on the domain that interested parties will return to for guidance, to access useful templates and tools, and to collaborate in its development.

At a minimum the RF should be published as some form of wiki that can be gradually extended, rather than a fixed report.

It is beyond the scope of this report to look at the following in detail, but these are all related activities that should be considered

- Implement Knowledgebase
- Provide Communications and Collaboration Systems that allow the RF to be advertised, and promoted collaboration in its rollout and development
- Provide a repository for shared deliverables and assets that users can discover
- Establish a template and patterns library
- Create associated education materials and rollout
- Implement corresponding Management Information System on the usage and status of the RF

Without enabling the RF through these kinds of capabilities the RF runs the risk of politely ignored!

Summary

The Reference Framework or Architecture will not emerge by accident from pilot or pathfinder projects. Larger enterprises need to take a structured approach to delivering the levels of architectural commonality and diversity that are appropriate for the organization at evolving states of maturity.

A more complete and detailed version of this research note, together with task definitions and accompanying assessment and planning tool in Microsoft Excel format is available to our [CBDI-SAE Knowledgebase](#) subscribers.

About Everware-CBDI

Everware-CBDI is an innovator in architectures and practices for Cloud, Service and Component based concepts, technologies and techniques. The company has guided many F2000 companies and government departments in establishing architecture and best practice. We make our documented best practices available to our customers through continuous skills development products, as well as providing collaborative consultancy that is designed to facilitate enterprises to demonstrate practice improvement and internalize the experience as repeatable processes.

Contact Everware-CBDI at www.everware-cbdi.com