UNCLASSIFIED

Representing Information Exchange Requirements using the MOD Architecture Framework (MODAF)

Version 1.1

12 November 2007

Ian Bailey

This document outlines the preferred approach to developing operational architectures in MODAF. It stresses a logical approach (which has been greatly clarified in v1.1 of MODAF) and identifies a simple iterative process for establishing information exchange requirements (IERs)

It is aimed at anyone wishing to use MODAF when developing IERs, or anyone who may need to read or review a MODAF architecture. The document was produced to clarify the preferred approach, which will help to improve consistency across MOD's architectural products.

Bringing service to life

serco

This document was produced by Model Futures Limited under sub-contract to Serco Consulting for the UK MOD.

Introduction

There has always been some confusion about how the OV-2 and OV-5 views in MODAF should be used. These views form part of the Operational Viewpoint in MODAF. OV-2 represents the logical structure of the enterprise[i], and OV-5 presents a very simple activity-based representation of the behaviour of the enterprise. Together, these views provide a useful tool for specifying and analysing information exchanges.

MODAF version 1.1 has sought to tighten-up their usage, and their relationship to other views, however, consistent Enterprise Architecture is not guaranteed by the existence of a framework alone. There is also a need for a consistent approach in the use of the framework. The need for a methodology for MODAF has been debated in the past, and is still being debated. What is being suggested here is not a methodology as such, but just a clarification about how some of the key MODAF views are related.

Background

There were some significant changes to MODAF with the release of version 1.1. At first sight they seem minor, but they were intended to clean up some of the inconsistencies in the framework and to ensure that architects were clear about how certain views were to be used.

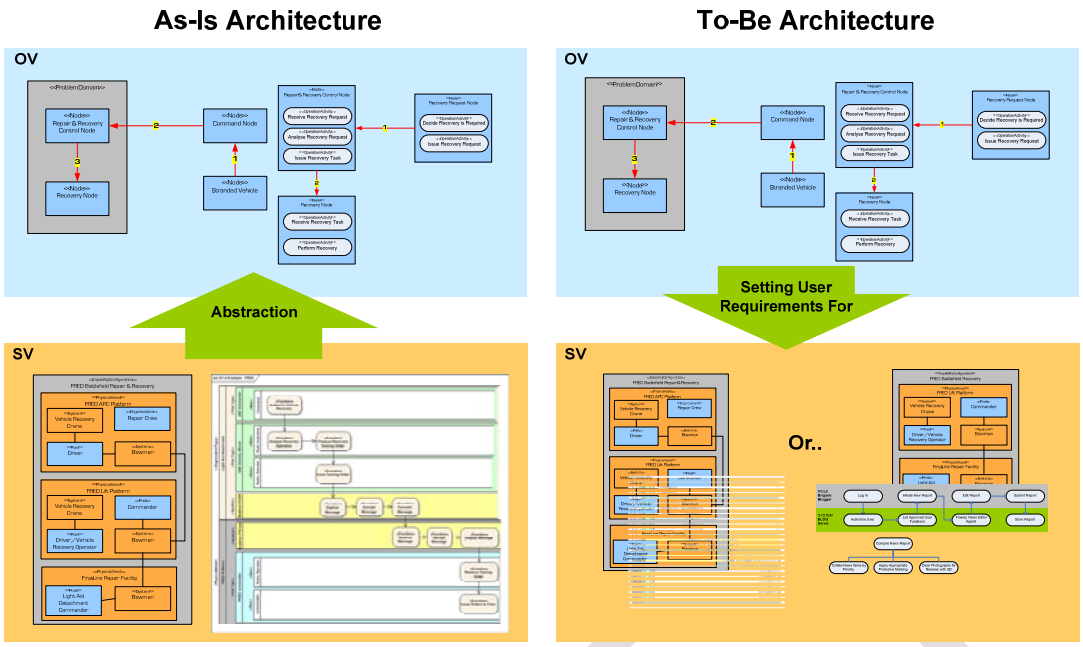
One of the reasons they seem minor is that the views retain the same codes (OV-2, OV-5, SV-1, SV-4, etc.). There was probably a strong case for renaming the Operational Views (OVs) and the System Views (SVs). The OVs aren't necessarily operational in the military sense of the word, but are logical representations of required or existing capabilities in context of each other. Similarly, the SVs describe systems in the broadest sense – i.e. they include humans. The original names were kept in order to maintain consistency with MODAF v1.0 and DoDAF, but it is quite useful to think of the OVs as "Logical" and the SVs as "Physical".[ii]

Previous to these changes in MODAF v1.1 some architects were interpreting the operational views to be business models and the systems views to be technical models. This was never the intent, nor was it the original intent of DoDAF. This tighter adherence to logical and physical distinctions means that architects need to think clearly before deciding if their process model is an OV-5 or an SV-4 (even if it is a business process model – remember the SVs now model human factors). Similarly, it is not acceptable to just treat the OV-2 nodes as platforms or organisations.

Logical Architecture

It is important to stress that the intent of a MODAF operational architecture is to present an abstracted, logical view of the enterprise. OV-2 describes *nodes* and the relationships between them (needlines). *Nodes* are simple agents that perform *operational activities* which, in turn, are the subject of OV-5. The OV-2 & OV-5 serve two purposes depending on the type of architecture[iii]:

- In an as-is architecture, they are used to abstract away physical complexity to show the key logical features of an enterprise. The actual implementation of this functionality is described elsewhere (in MODAF System View products). An as-is logical architecture allows a consistent view of enterprise functionality to be maintained, so that various what-if SV designs can be trialled in change programmes – i.e. it provides a logical baseline against which the impact of change can be measured.
- In a to-be architecture, they provide a logical description of what the enterprise shall do, and they key actors needed in order to do it. For this particular logical requirement, there may be many possible solutions, and these may be traded-off against each other (and may be described using MODAF System View products).



This then begs the question; what is a logical architecture? A broad rule of thumb is to consider the *what* rather than the *how*. A logical architecture specifies what the enterprise is, does, or is required to do, and what the logical actors in that process are. The actors (Nodes) may correspond directly to individual organisations, people or systems, but this does not have to be the case. For example, in a to-be ISTAR[iv] architecture, we may specify only one Operational Node that conducts the observation activities, but in reality this is implemented by a variety of different sensors and human observers.

How far the architecture is logically abstracted is really a matter of fitness for purpose, and MODAF allows for multiple levels of decomposition in OV-2 and OV-5 (i.e. the architect can produce different, but related OV products for different stakeholders). However, in general the overall level of detail should be kept as simple as possible – so as to abstract away complexity in an as-is architecture, and to allow the necessary freedom of design expression in a to-be architecture.

Information Exchange Analysis

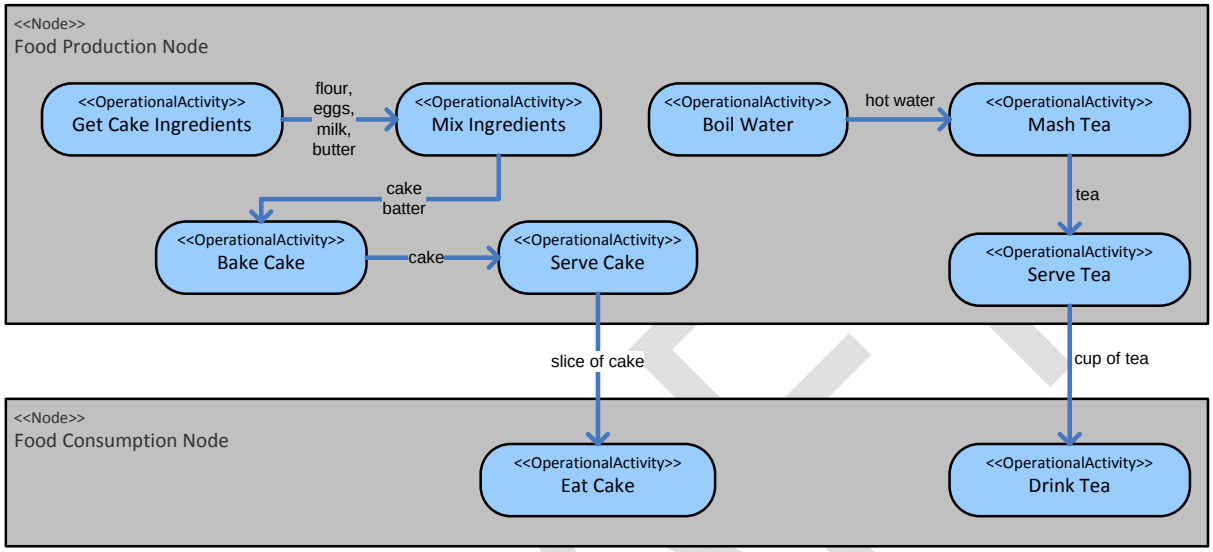
Note: *The term "Information Exchange" is used here rather than Information Exchange Requirement (IER). An IER tends to suggest a requirement for some future capability rather than something that exists. MODAF architectures are either to-be or as-is. Therefore an information Exchange in a to-be architecture will indicate a requirement, whereas in an as-is it illustrates and existing capability.*

A logical architecture is commonly used to identify key information exchanges in an enterprise – these may be actual exchanges (in an as-is architecture) to requirements for exchange (to-be). The logical approach is particularly useful in information exchange analysis, as it allows the architect and other stakeholders to clearly see the information exchanges without the complexity of implementation. A real implementation of an information exchange may be very convoluted, requiring relays, junctions, redundant connections, etc. and although this is useful information to a technical architect, it doesn't always help in getting the message over to managers and users.

In developing a logical architecture in MODAF, there is always something of a chicken-and-egg aspect to OV-2 and OV-5 – do we specify our processes first, or our nodes[v]? Most experienced DoDAF and MODAF architects will say that actually the OV-2 & OV-5 are both developed together, and there are usually a few iterations as new Nodes are introduced, processes established, etc. Any information exchange analysis, therefore, will also be an iterative one. The basic approach in MODAF (and in DoDAF) is to model the flows of information between Operational Activities. This can be quite a

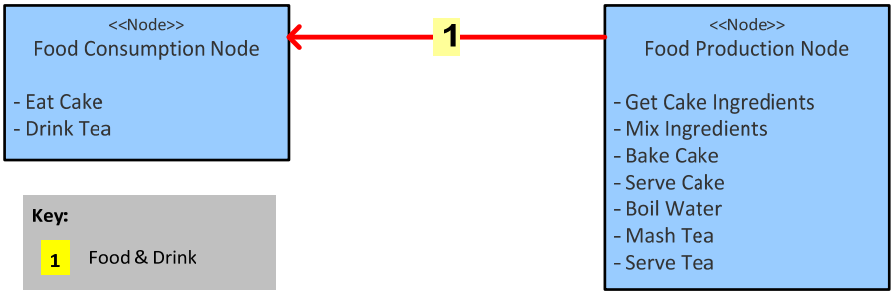
difficult task, but the best way to think about it is to consider what information an actor (Operational Node) would require in order to carry out a task (Operational Activity). It should then be reasonably clear which other tasks produce that information and so a flow can be established. Again, this is an iterative process – the architect may realise that there is no activity which produces the information required, and so this must be modelled.

Once the information flows between activities are in place, and it is clear which nodes conduct the activities, the flows across nodal boundaries become apparent – in the example below, the flows are not information (they're *food* and *drink*) but the principle is the same:



We've used tea and cake here rather than information for two main reasons 1) it is universally understood; 2) the cake baking example is usually used in process modelling classes to teach the principles. In the example, there are clearly two flows which cross the nodal boundaries, hence there are two information (or in this case food) exchange requirements. Note that the model says nothing about how any of the activities take place (e.g. do we use a kettle or a pan to boil the water ?), which keeps everything suitably abstract for analysis purpose. The description of how the water is to be boiled will be specified in an SV-4 functional model, with the accompanying SV-1 resource interaction model. By hiding the complexity of how it is actually done (or could be done), the key information exchanges are exposed in sharp relief, and the need to highlight the exchanges should be a guiding principle in developing OV-2 and OV-5 products.

An information flow between activities (official MODAF name: Operational Activity Flow) has greater significance when it crosses nodal boundaries. When this occurs, it signifies some need to exchange information between people, systems, sites, etc. An OV-2 highlights these key exchanges by simplifying the OV-5 view down to nodes and "needlines":



Clearly, the OV-2 provides similar information to the OV-5, but hides the activity flows. In this case, the cup of tea and slice of cake are bundled into one "Food & Drink" needline. How many information flows are bundled into a needline is again a design decision for the architect – OV-2 is all about presenting a complex architecture in a simple way. The usual practice, however is to bundle those flows that carry information that is likely to be related. *Note that it is optional to overlay nodes on an OV-5, and also optional to show activities on an OV-2.*

In most cases, information flows / exchanges will be finer grain than needlines – i.e. needlines will tend to be collections of information exchanges. In some cases though, a needline may carry only one information exchange. MODAF's meta-model relates needlines and information exchanges by introducing the idea of an information element. These elements are carried by information flows between activities. Selected elements can then be identified as part of the information exchange, and needlines can be used to group the exchanges. The OV-3 itself is a tabular view, showing the needlines, information exchanges and information elements. The tabular approach also allows the architect to define properties of the information exchanges such as information assurance attributes, bandwidth requirements, etc.

Needline N^{o}	IE N^{o}	From Act	To Act	Bandwidth Needed	Security Classification	etc.	etc.
1	1	Serve Cake	Eat Cake		Unclassified		
1	2	Serve Tea	Drink Tea		Top Secret		

Conclusion

The key to developing OV-2 and OV-5 products is to keep the modelling as abstract as possible. In so doing, this creates a structure against which information exchanges can be modelled without the need to show the complexity of the underlying physical architecture (or, in the case of a to-be architecture, to prevent "solutioneering" in the user requirement).

The combination of nodes and behaviour (operational activities) shouldn't be any great surprise to a systems engineer, though it may be unfamiliar to requirements managers who tend to work with textual specifications for user requirements. Not all systems engineers will be accustomed to working at this level of abstraction though, and it requires a certain level of discipline to prevent the operational architecture becoming physical. It is very easy to represent systems, people, organisations, etc. as nodes, but this doesn't really add anything above and beyond the physical architecture (which is handled by the Systems Views in MODAF).

Information flows are specified between operational activities. Those activities are performed by nodes. When a flow exists between activities conducted by different nodes, there will be an information exchange. In a to-be architecture, this represents a potential[vi] information exchange requirement. In an as-is architecture, the information exchanges and needlines are used to simplify the presentation of the key exchanges that take place in the enterprise.

Following these simple guidelines will help result in consistent MODAF architectures, where the operational views do more than simply re-present the physical architecture.

[i] For the purposes of this document (and for Enterprise Architecture in general), an enterprise may be of any size from a small project to a large multinational company or Govt department. It is a common mistake to assume enterprise architecture is about modelling the whole business. EA is about bringing together the different business and technical strands to provide an "enterprise view" – i.e. not one that is purely business or IT.

[ii] It is useful to put the MODAF OVs in context of the Strategic Views (StV) and SVs. The MODAF Capability Taxonomy (StV-2) describes capabilities in general. It does not specify the implementation of the capabilities, nor how they are deployed. A capability in MODAF is simply a statement of some ability to deliver an effect. The architect may specify metrics for the level to which the effects are achieved (e.g. a maximum rate of advance for a ground manoeuvre capability). StV-6 specifies standard (i.e. doctrinal) processes, and as with StV-2, these are specified independently of implementation or deployment (so think JETL more than METL). To be of any use, the capabilities and processes must be shown logically in context of a particular scenario, and this is the job of the OV. The nodes in an OV-2 may be instances of one or more capability, and the activities in an OV-5 may well be defined originally in an StV-6 . Finally, the logical nodes specified in the OV-2 are detailed as resources (systems, people, platforms, etc.) in the SV-1, and the logical activities presented in OV-5 are put in context of the resources that deliver them in an SV-4 functional model.

[iii] Enterprise architectures usually either describe the existing state of an enterprise (as-is) or some future, required state of the enterprise (to-be)

[iv] Intelligence, Surveillance, Target Acquisition, and Reconnaissance

[v] Nodes in MODAF are logical agents (or actors) that conduct operational activities – they may be realised by people or systems, or combinations of the two.

[vi] Potential, because in developing the physical architecture, it may be more economic to co-locate the information provider and consumer than implement the necessary communications interface.