

Using Soft Systems with MODAF

Version 1.0
06 November 2008

Ian Bailey

(ian@modelfutures.com)

This white paper reports on research into how the Soft Systems Methodology (SSM) can be best integrated with a MODAF enterprise architecture approach. The paper concentrates on mapping the products of an SSM analysis into the appropriate MODAF views, and how to frame the products in terms of MODAF enterprise phases.

The similarities between MODAF OV-2 & 5 products and the SSM Consensus Primary Task Model are highlighted, and a recommendation is made for how SSM models should be positioned relative to As-Is and To-Be architectures.

This white paper was sponsored by the Thales CIPHER Team and authorised for release to the wider MODAF community as a contribution to guidance on best practice in the use of MODAF.



Acknowledgements

Many thanks to **Brian Wilson** for conducting two very detailed reviews of this document. It goes without saying that Brian is a world authority on SSM, and without his support this document could not have been written.

Thanks also to **Mike Duffy** and **Fintan Hanbury** for driving this work and suggesting the white paper topic. We have used SSM and MODAF on three different live projects now (IADP, NHS CfH and CIPHER), and this experience has been invaluable in forming the conclusions presented in this document.

Introduction

The Soft Systems Methodology (SSM)¹ sets out to build a model of *what* a Human Activity System must do if it is to achieve the purposes defined. The purposes are intended to be relevant to a specific part of “The Real World”, which might range in size from a multi-national enterprise, through an individual organisation to elements within an organisation. These statements of purpose are captured in structured definitions called “Root Definitions”. Typically, using these definitions, an SSM practitioner will derive the logically interconnected activities that an enterprise *should* conduct in order to fulfil its purpose as defined in the root definition. The resulting SSM Consensus Primary Task Model (CPTM) is usually used as a reference model against which the current enterprise can be compared to see how efficiently and efficaciously it fulfils its purpose. A CPTM is only as good as the root definitions from which it is derived – provided consensus is reached on the root definitions, the CPTM is considered logically defensible.

An SSM model is also often used to analyse information flow requirements. An SSM model can be decomposed into sub-systems (i.e. a set of defined areas of responsibility in the model that conduct and control the enclosed activities). The activities will produce and consume information. An SSM practitioner can develop a set of information flow requirements by deriving the information categories that are required as a support for each activity. These information flow requirements can then act as a guide or precursor to information systems development.

MODAF is the MOD Architecture Framework. It is an MOD standard for representing and exchanging enterprise architectures. It defines a set of views – standard types of diagrams, tables and documents – that are to be used by architects working on MOD projects. The views cover a number of domains and levels of abstraction. The *Strategic Views* deal with management and specification of military and business capability. The *Operational Views* provide high-level logical specifications of the enterprise and the *Systems Views* describe the implementation of the logical specifications from the Operational Views. In v1.2 of MODAF, *Service Oriented Views* were added, which augment the logical Operational Views in providing a specification of what the enterprise shall do. There are also standards views and programmatic views, which although they are not strictly architectural views, do support the enterprise architecture process.

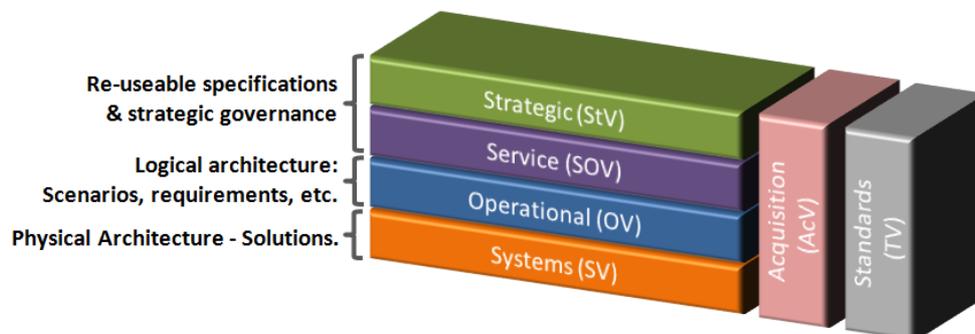


Figure 1 - The MODAF Viewpoints

As well as the view specifications, MODAF also specifies the types of elements that architects can use in their models, and also the key relationships that can exist between those elements. This is known as the MODAF Meta-Model, and it is a formal specification which MODAF tool vendors are expected to implement. The most significant element in a MODAF architecture is the enterprise (or enterprise phase). This element represents the domain of interest for the architecture. A given enterprise may have several enterprise phases, and each of these is defined in terms of its goals, capabilities and enduring tasks. This allows an architect to model how the enterprise is expected to evolve over time, and the enterprise phases act as the entry points for the logical and physical architectures (as depicted in Figure 2).

¹ For the purposes of this white paper, we have only considered the “Wilson” flavour of soft systems. However, it is unlikely that the mappings presented here would be any less true for a “Checkland” approach. See http://en.wikipedia.org/wiki/Soft_Systems_Methodology

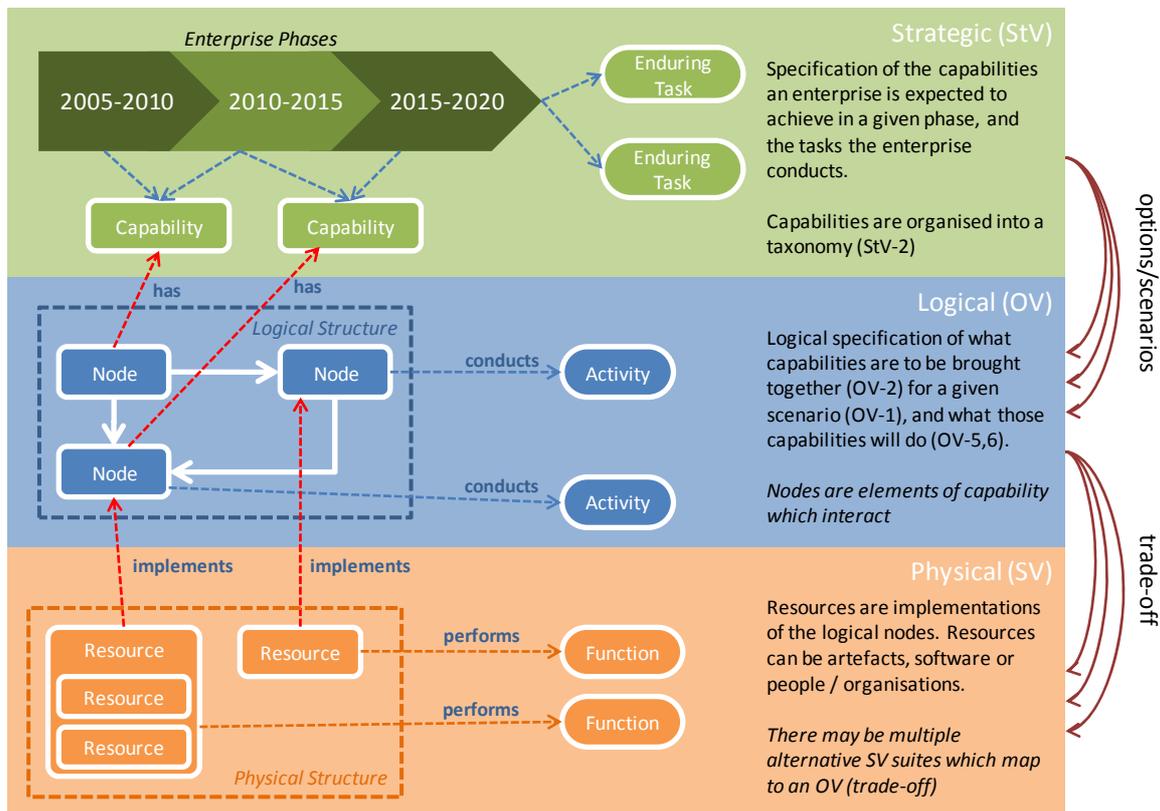


Figure 2 - Key MODAF Elements and their Relationships

The MODAF concept of enterprise phases, and the logical architectures that represent them are crucial to how SSM is to be mapped into MODAF.

Initial Mapping of Soft Systems (SSM) to MODAF

The purpose of this white paper is to ascertain how MODAF and SSM can best be used together – i.e. how the MODAF framework should be used to represent the products of an SSM analysis. MODAF is not a methodology, but in order to ensure tight integration and coherence in enterprise architectures, MODAF does force a certain order on the architectural process. For example, the Strategic and Service views are independent of any project or scenario – the capabilities and services they define are intended for re-use across multiple architectures. There is also clear distinction between logical and physical architectures. The Operational Views are intended to specify *what* an enterprise does; the elements of the enterprise that are logically required to deliver its capabilities. The Systems Views then define *how* the logical elements and activities specified in the Operational Views are actually implemented as human resources, systems, platforms, etc. So, although there is no formal method, a MODAF architect has to work with certain built-in dependencies in the framework – it is especially important to ensure that OV products are logical (i.e. implementation-independent).

Soft Systems defines a method for arriving at a conceptual model relevant to an enterprise. SSM is not just a method though, it also specifies set of views. These are the root definitions (text), the consensus primary task models (CPTMs - activity models), the systems and sub-systems, and the “Maltese Cross” tables for information flows. SSM also has no formally published meta-model, but the meta-elements are implicit – systems, sub-systems, tasks, information flows, etc.

At a superficial level, it is clear that both MODAF and SSM deal with enterprise structure and behaviour. It is also clear that there are some syntactic similarities in their products – e.g. OV-5 in MODAF and the CPTM in SSM. However, to integrate SSM and MODAF properly requires a more in-depth consideration of each standard. In particular, SSM is often used as an early stage analysis tool in change programmes and information systems delivery projects, and there is benefit in ensuring that the outputs of the SSM can be re-used in a MODAF EA environment.

The most benefit would seem to be in using SSM products as a reference² or goal architecture – i.e. a logically defensible model of what the enterprise *should* be doing (provided the root definitions accurately represent the purposes of the enterprise). This (logical) reference architecture can then be compared with as-is and to-be architectures to see how efficiently and efficaciously they meet the purposes of the enterprise. In addition, there is potential for using SSM as a method for arriving at capabilities to populate the Strategic Views in MODAF:

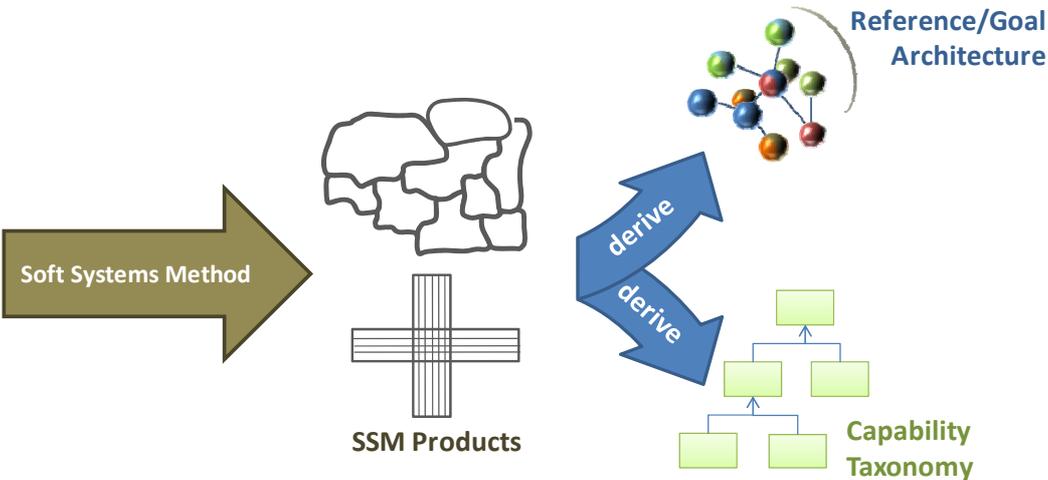


Figure 3 - Deriving EA Products from SSM

The challenge therefore is to define what outputs from the SSM can be used in MODAF, and what further analysis is required to derive a reference architecture and capability taxonomy. In addition, the concept of a reference architecture is not covered by MODAF, and a formal way to map this will be required.

Reference Architecture

Before looking into the process of deriving MODAF views from SSM products, it is necessary to position the SSM products in the enterprise architecture. As mentioned in the introduction, MODAF enforces a top-level context to architectures based on the principle of enterprises and enterprise-phases. An enterprise is any aspect of a business endeavour the architect chooses to analyse. The enterprise sets the scope for the architecture – it may be anything from a small project to an entire Govt department. MODAF then splits the enterprise into temporal phases – this allows the architect to have any number of architectures for given phases of the enterprise:

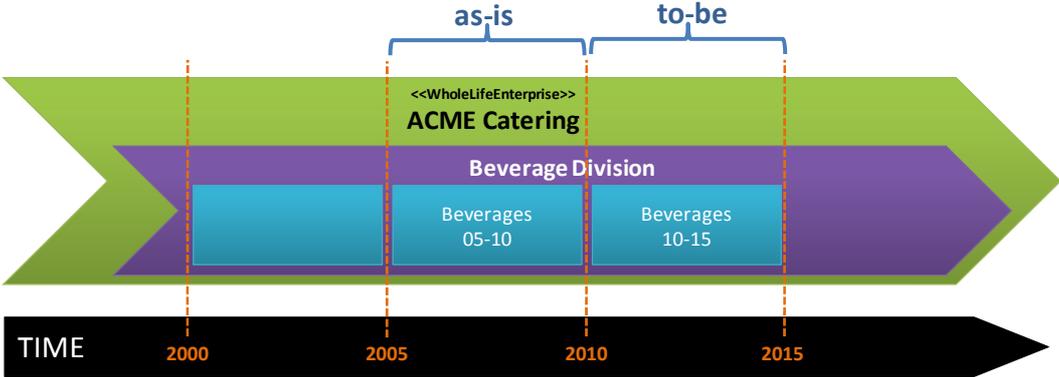


Figure 4 - Example of Enterprise Phases in MODAF

² This should not be confused with the TOGAF concept of a reference model. In this case, we mean a logical architecture depicting what the enterprise should be doing and how it should be structured in order to meet its purposes. Chances are, the enterprise will never actually match this logical model, so should be considered the ideal case against which reality can be compared. Hence the term “reference architecture”. To avoid this confusion altogether, it may be wiser to use the term “goal architecture”.

The products of an SSM analysis are, by their nature, not specific to a given phase of an enterprise. Provided the purposes of the enterprise do not change, the SSM models will specify what the enterprise should be doing, regardless of when the organisation is examined³. This raises the first issue of mapping SSM to MODAF. There is no concept of a reference enterprise in MODAF – though it could be argued strongly that this concept should be added. For this reason, it is necessary to adopt a best-fit policy and use Enterprise Phase. This would result in something like:

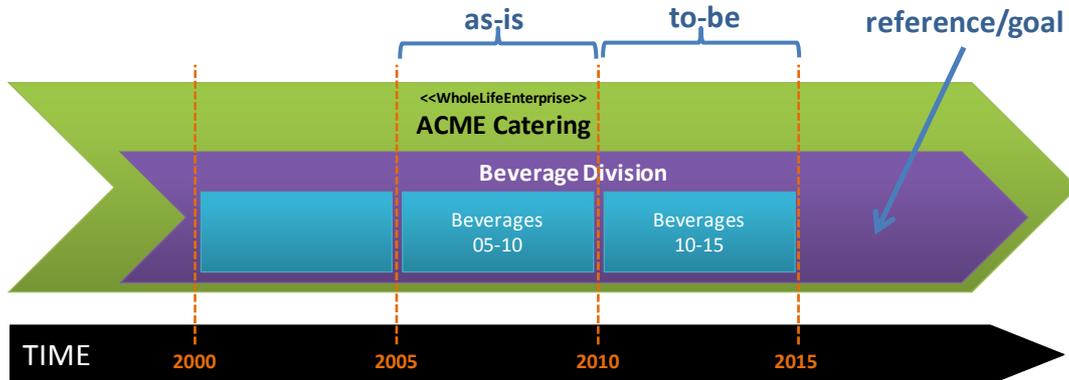


Figure 5 - Representing "Reference" State of Enterprise in MODAF

This approach places the SSM model as a description of the whole-life enterprise – i.e. regardless of whatever change programmes are delivering, this is the model of *what* the enterprise should be doing at any given time. There is a strong case for introducing the concept of reference/goal architectures in a future version of MODAF, however.

With this in place, the SSM products fall quite neatly into MODAF Strategic and Operational (Logical) views – StV-1, OV-2, OV-3 and OV-5. The root definitions map onto vision statements in StV-1. The systems and sub-systems from a CPTM are equivalent to nodes in OV-2. The functions performed by the systems are operational activities in an OV-2. The information flows between functions correspond to operational activity flows (OV-5 & OV-3). Where the flows cross system (node) boundaries, they correspond to IERs & Needlines (OV-2 & OV-3). This is illustrated below:

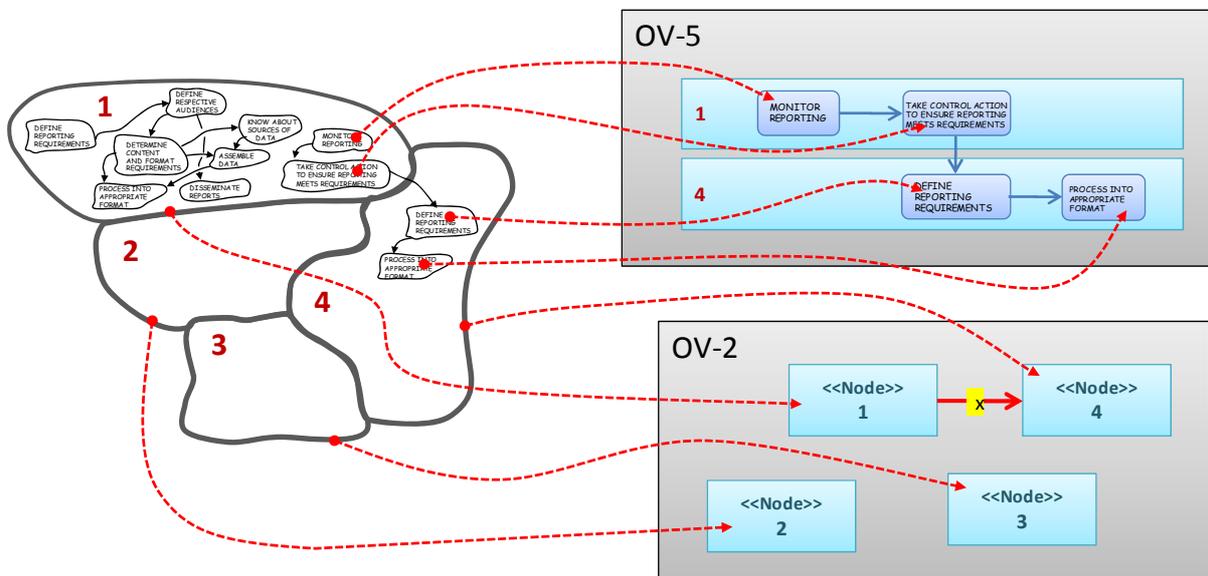


Figure 6 - Mapping SSM Products and Elements onto MODAF

³ There can be exceptions though. An SSM model is defined taking into account external influences – e.g. legislation, markets, etc. In some cases, changes to these external influences may render a model out of date.

The OV-2 and OV-5 constitute the logical architecture for the reference enterprise – i.e. the idealised state of the enterprise as defined by the SMM model:

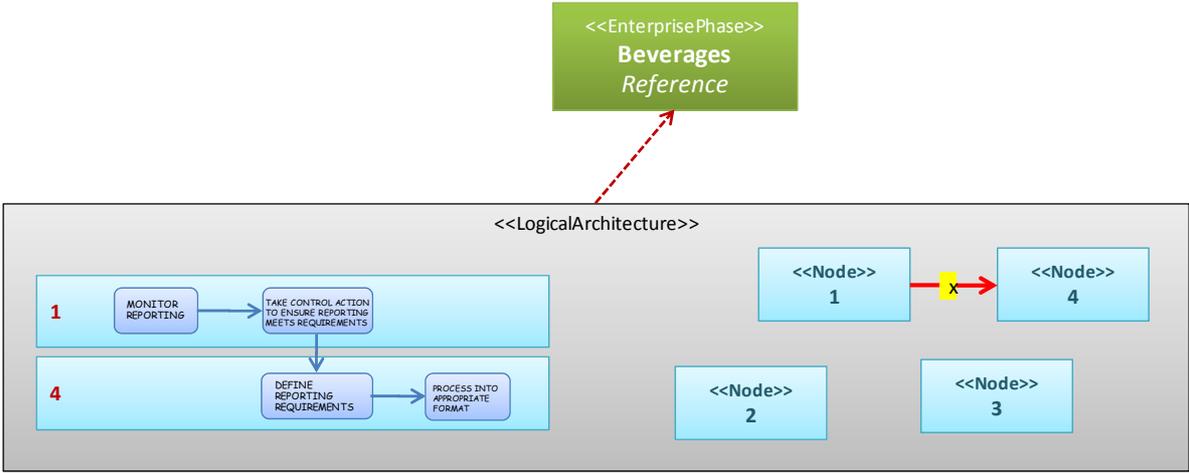


Figure 7 - Logical Architecture for Reference Enterprise

Deriving Capabilities

The other interest in SSM is as a defensible method for arriving at capabilities. In MOD / MODAF terms, capabilities are high level specifications of an enterprise’s ability to do something. Given that SSM defines a set of sub-systems and activities which form the ability of the enterprise, it should be possible to derive required capabilities from the sub-systems and their activities. For example, if the SSM analysis results in “a system to provide liquid refreshment”, then we can easily derive a capability “liquid refreshment provision”. The derivations may not always be this straightforward. In addition, there are likely to be capabilities of the current organisation that are not derivable from the SSM products – either because the current organisation has unnecessary functionality, or because the capability is at a finer-grain level than those derivable from the SSM analysis. Finally, in architecting future changes to an enterprise, there may be a set of customer-driven requirements (URDs) to which the architect must comply. These requirements may also point at capabilities not deemed to be required by the SSM, or those not identified by the SSM:

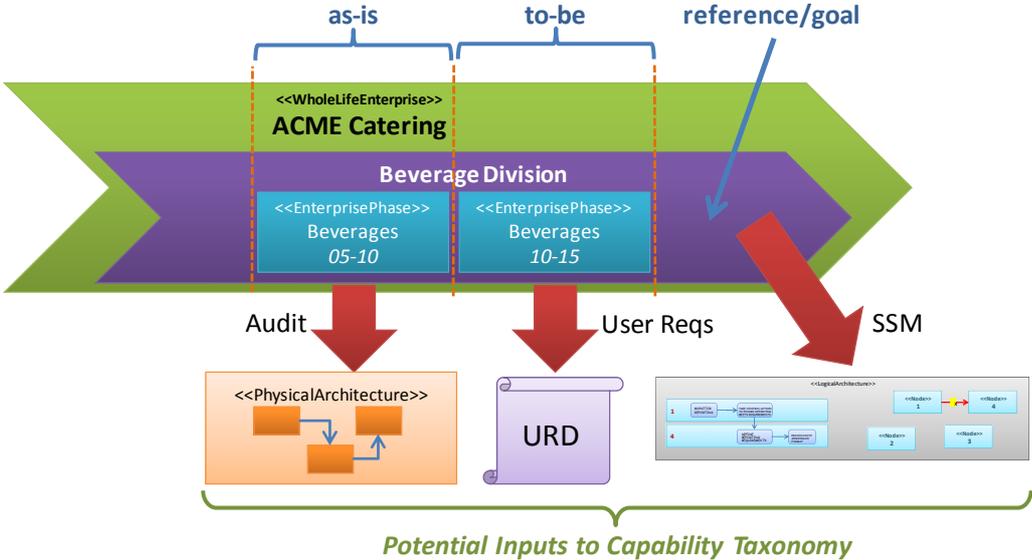


Figure 8 - Source Material for Capability Derivation

A scenario such as this is not uncommon, and is often faced by solution providers. Their customer has requirements, they have a pretty good idea of current capability, and it’s fairly common practice (at least in UK Govt) to have a soft systems model of the domain. It is far less common for the customer to provide the supplier with a MODAF capability taxonomy (though as organisations become more mature in their approach to EA, such an approach is likely to take hold).

The challenge in deriving capability taxonomies from these types of input is in finding a rigorous and defensible method for comparing the data from the raw input data; documents, SSM models, physical system models, etc. In most cases, this will rely on technical and domain (e.g. military) judgement – it will require an ability to assess when two activities represent the same type of capability.

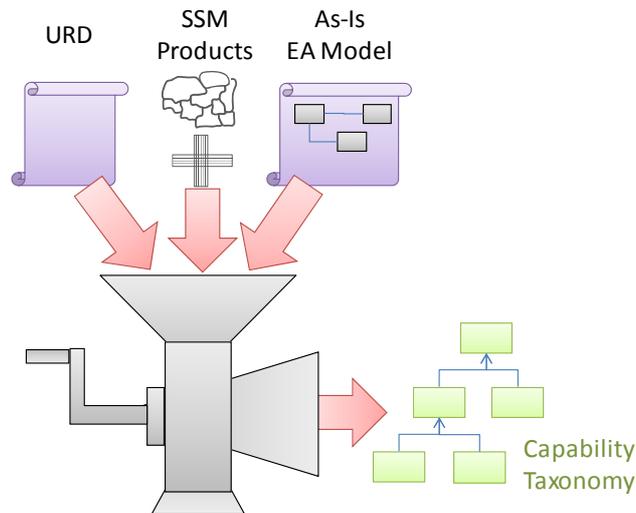


Figure 9 - Processing Raw Input Data to Derive a Capability Taxonomy

One possible solution to this may be the BORO⁴ Method. This approach has been used to great success in the IDEAS Group⁵ to compare the enterprise architecture frameworks in use in the AUSCANUKUS community. Although BORO was not designed for deriving capability, its rigour means that any capabilities derived will have a fully defensible audit trail back to their source documents or models. The process is repeatable at a semantic level – i.e. the output will always be the same if the input refers to the same thing in the real world (even if very different terminology and approaches have been used in generating the input material). The main drawback to this method is that it is time consuming - each concept has to be analysed according to the strict rules of the BORO method. However, for any given project, this need only be done once, and the utility of a formal, fully-defensible set of capabilities should not be underestimated.

Conclusions

Soft Systems models do map well into MODAF, but require the creation of a “reference” enterprise, for which they form the logical architecture specification. There is no concept of a reference enterprise in MODAF, so the best alternative is to use a whole-life enterprise as the basis for the SSM products. The resulting MODAF products are:

- StV-1 – the vision statement can be taken directly from the SSM Root Definition
- OV-2 – the nodes in OV-2 represent the sub-systems in an SSM model
- OV-5 – the operational activities in OV-5 represent the activities in an SSM CPTM
- OV-3 – the information flows and elements from the SSM Maltese Cross and CPTM

Producing a capability taxonomy requires a further derivation step. If done in an ad-hoc fashion, the resulting capabilities will be hard to defend. However, the BORO™ method may offer a defensible way to create a set of capabilities that can be precisely and defensibly traced back to the originating material (e.g. SSM models, URDs, existing systems and organisations, etc.).

⁴ Business Object Reference Ontology™ – see <http://www.boroprogram.org/>

⁵ See <http://www.ideasgroup.org>

Annex – A Quick Overview of the BORO Methodology®

In setting out to develop an ontology for data sharing and integration, it is important to ensure that all parties use the same approach when mapping legacy information onto the ontology. Without this assurance, there cannot be any integration or sharing at a semantic level. Without formal methods of analysis, even two experts are unlikely to come up with the same ontology for the same domain. This level of unreliability and lock-in of expertise isn't acceptable to IDEAS, so the modelling team cast around for a suitably formal method. The IDEAS Group adopted BORO, not because they wanted to develop an ontology, but because they needed a defensible, solid method that would deliver repeatable results in the hands of any trained analyst.

BORO stands for Business Object Reference Ontology. The purpose of the method is to re-engineer disparate sources of information into a common model. It is particularly good at semantic analysis – establishing whether two concepts are the same, if they overlap, or if they are unrelated. BORO is different to most of the existing data analysis methods. The process itself ignores the names of things. Instead, the analyst is forced to identify individual concepts by their extent. In the case of physical objects, this is their spatial/temporal extent. For types of things, the extent is set of things that are *of that type*. The BORO methodology can be summarised as a flowchart:

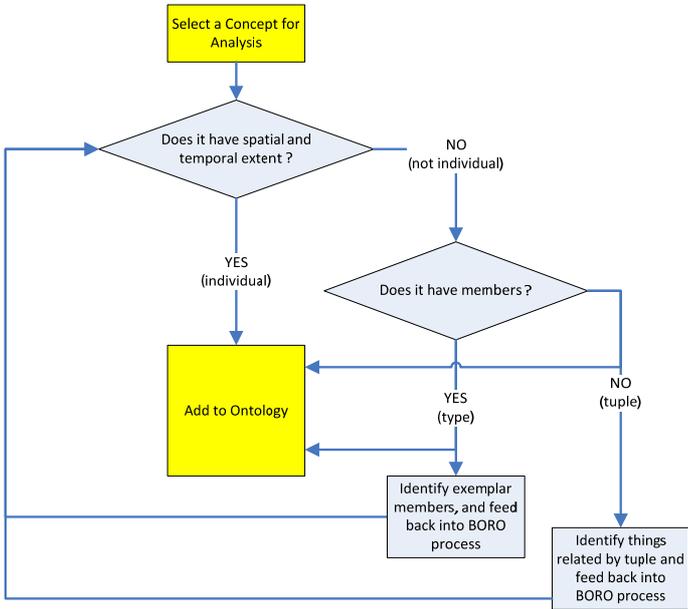


Figure 10 - Flowchart Summarising the BORO Methodology

Because the elements in a BORO-derived ontology are identified by their extent, there can be no question as to whether two things are the same or different (this is the key to semantic interoperability). Once the identity of something has been determined using BORO, names may then be applied to the elements in the ontology.

About the Author

Ian Bailey was part of the original team that developed MODAF. He has since provided the technical expertise for versions 1.1 and 1.2 of the standard. He was the MODAF subject matter expert in the MOD DG-Info Enterprise Architecture Programme, and has worked on EA projects and provided EA training to customers such as the NHS, Cabinet Office, CESG, Serco, BAE Systems, FOI, Thales and Shell. He provides the MODAF expertise to the OMG UPDM team, and is lead modeller in the IDEAS Group. Prior to working on MODAF, he was editor of AP233, an ISO/INCOSE standard for systems engineering data, and also acted as AP233 liaison to SysML. He co-authored the white paper on DoDAF, SysML and AP233 that was the trigger to the UPDM project. He has a first degree in engineering & management from Brunel University and Henley Management College, and a PhD in software integration.