INTEGRATION AUTHORITY

MOD Abbey Wood #2308 Bristol UK BS34 8JH

Implementation Strategies for the MODAF Taxonomy

1.0

14 February 2005

IA/02/16-ERMcm01

Prepared by: ………………………..

Name: Ian Bailey
Post Title: IA1dCon5
Phone: +44 (0)117 913 4045
Mobile: +44 (0)7768 892 362
Email: ian.bailey@cornwell.co.uk

Approved by: ………………………..

Name: Mr A North
Post Title: IA1d
Phone: +44 (0)117 913 4237
Fax: +44 (0)117 913 4935
Email: IA1a@dpa.mod.uk

RECORD OF CHANGES

This page will be updated and re-issued with each amendment. It provides an authorisation for the amendment and a checklist to the current amendment number.

Issue No.	Date	Revision Details
Draft 0.1	23 November 2004	First draft for internal review
Draft 0.2	13 December 2004	Second draft for IA review
Draft 0.3	18 January 2005	Modifications in-line with selection of XMI for tool exchange.
Release 1.0	14 February 2005	Cosmetic changes, prior to release.

Copyright Details

Summary

This paper proposes a way forward for developing the MODAF[1] taxonomy that makes use of semantic web standards under development by the World Wide Web Consortium (W3C). The proposed solution is to use OWL (the Web Ontology Language) which was originally intended for defining structures and relationships that lend meaning to information on the web. In its simplest form, however, OWL is rather good for defining classification hierarchies – taxonomies.

The OWL approach offers the following advantages:

- *It is XML, as specified by the MOD for all future data exchange projects*
- *It allows taxonomies to be developed and hosted at different sites, with links such that the set of taxonomies can behave as one integrated taxonomy*
- *There are a wide variety of free authoring and development tools*
- *It is a standard from the same stable as HTML, XML and Web Services*
- *It can be easily converted into formatted text for publication*
- *It is already being trialled in other parts of the MOD for taxonomy and ontology development*

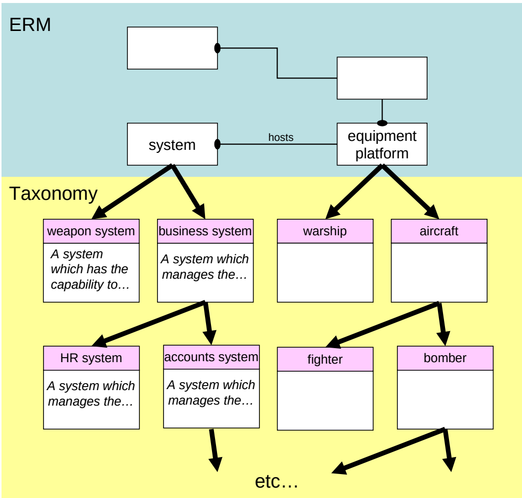
This paper does not address implementation architectures for the taxonomy, but does suggest possible technologies that should be considered – including web services to provide applications with online access to taxonomy.

[1] The MOD Architectural Framework (MODAF) is being developed with the intention of providing a rigorous way to specify systems of systems. The framework will predominantly be used for acquisition purposes, and a key driver for its adoption is the need to improve interoperability between systems. However, MODAF could equally well be used to analyse existing, operational systems and better enable their integration with other systems (both new and existing).

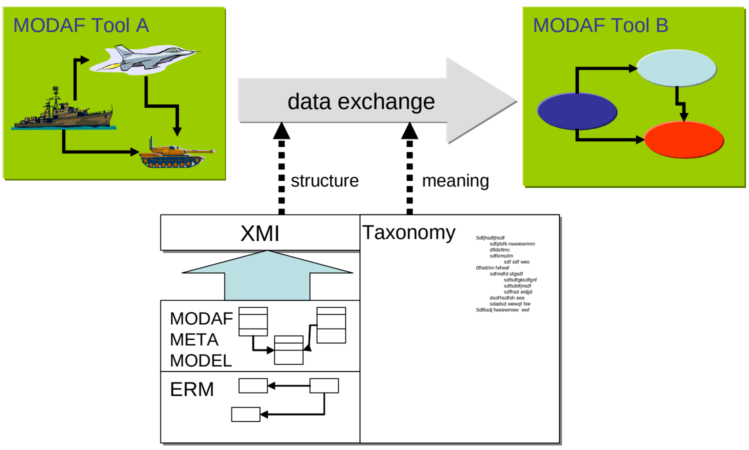
Introduction

> **taxonomy** ***noun*** (***taxonomies***) **1** *biol* the theory and techniques of describing, naming and classifying living and extinct organisms on the basis of the similarity of their anatomical and morphological features and structures, etc. **2** the practice or technique of classification. **taxonomic** ***adj***. **taxonomist** ***noun***.
> **ETYMOLOGY:** 19c: from Greek *taxis* arrangement.
>
> ***Chambers 21st Century dictionary***

The MOD Architectural Framework is underpinned by a conceptual information model. This model, the *Enterprise Reference Model (ERM)*, is intended to capture all the business concepts required for MOD enterprise architecture. The ERM acts as the driver for the MODAF meta-model that specifies the XMI[2] exchange file content between MODAF tools. The ERM is a high-level model, with the intention that a taxonomy will be used to specialise the generic concepts with commonly used terminology, equipment types, etc. The taxonomy provides specific definitions for the (generic) elements defined in the framework – i.e. it provides reference data.



All MODAF data exchange files will use the same data structure, the different views being covered by overlapping subsets of the ERM. Data elements (instances) in the exchange files will refer to appropriate definitions in the taxonomy, adding meaning to the high-level ERM instances.



[2] See IA paper IA/02/16-ERMcm03 – "XMI, UML & MODAF"

There are several business and technical requirements on the taxonomy. From a technical perspective, it is important that the taxonomy be based on open information standards that are easy for vendors to implement (e.g. XML) and for tools to link into (e.g. web services). In other words, the taxonomy should be available to tools in such a way that the user views the taxonomy information seamlessly in the tool. For example, when adding a system to an SV-1 view, the tool would query the taxonomy for all the different system types and give the user a choice from a menu. From a business perspective, the MODAF taxonomy shall:

- enable specialisation of the generic elements defined in the ERM, but also support the data exchange elements defined in the logical data model
- supplement the MODAF documentation to provide element definitions
- be extensible – e.g. an IPT may wish to further specialise some definitions
- define classes – e.g. equipment types, ranks, communications specs, etc.
- define individuals where necessary – e.g. numeric constants, countries, etc.
- be compatible with other reference data / taxonomy approaches in the MOD (e.g. PLCS, CDMA's CRI)

XML and Taxonomies

In order to discuss the development of the taxonomy, one first has to consider the ERM. The taxonomy is inexorably linked with the ERM, and also must take into account the implementation models based on the ERM[3] – the MODAF meta-model and the web services specification for MARS. The implementation format for MODAF data exchange is to be XMI, the OMG's model interchange standard. The XMI content will be constrained by a UML profile which is founded on the ERM. XMI is an XML specification.

The use of XMI for MODAF tool interchange does not necessarily dictate that XML be used for the taxonomy, but there are several reasons to use XML:

- XML permits simple referencing between files using URIs, and implementations are capable of navigating such references. In the case of the W3C OWL XML specification, this enables the taxonomy to be distributed and also allows individual teams to extend the "standard" taxonomy for their own use.
- There is an abundance of reliable XML tools on the market, many are even free
- There is a strong IT skill base in XML – virtually every programmer has worked with it at some time
- XML Schema is an effective data modelling language for XML data, and has most of the capabilities of an entity-relationship model
- XSLt, the XML transformation language, allows XML data to be presented in a number of different formats, which need not be XML. Hence, an XML taxonomy can be presented in a tabular format for inclusion in the MODAF documentation, or as a navigable HTML tree

For a taxonomy, extensibility and connectivity are what really matter though. Any taxonomy developer must accept that other taxonomies exist, even within their own organization. For this reason, it is important that there are references between elements in different taxonomies, to:

- specialise those elements – e.g. you may wish to specialise "system" to "weapon system"
- indicate a synonym – e.g. an element "tank" in one taxonomy may be the same as "armoured fighting vehicle" in another

[3] See IA paper IA/02/16-ERMcm02 – "Implementing the MODAF Enterprise Reference Model"

- add a predicate relationship – e.g. you may wish to link the class “aircraft” in one taxonomy to the class “pilot” in another, using a predicate called “operator”

Given these requirements, the MODAF team has looked to other efforts within the MOD for guidance. The natural home for the taxonomy is DG-Info. The CDMA has long term plans for a corporate reference data repository (to be called CRI – Corporate Reference Information). However, work has only just begun on their DDR system for XML Schemas, and there are no specific plans for CRI as yet. There is another data exchange project within the MOD, product lifecycle support (PLCS), which is also looking to CDMA to manage their reference data. This project has decided to use an XML approach for their taxonomy development, based on the W3C (world wide web consortium) semantic web standards. They have chosen OWL (web ontology language) to specify their various taxonomies, because it provides all the extensibility and referencing capabilities required for reference data development.

Suggested Approach - OWL

The OWL language is a very simple XML structure for defining classes, individuals and the relationships between them. OWL, at its most simple, can be used to define the kinds of hierarchies of classes needed for a taxonomy. The example below shows a simple specialisation of vehicle into car, boat and aeroplane:

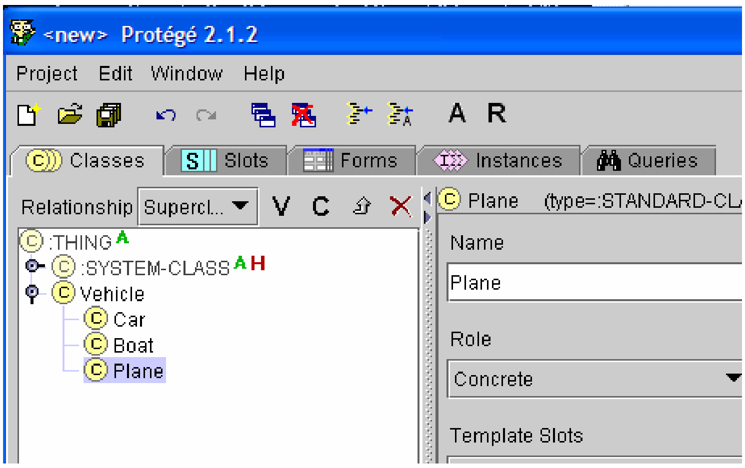
```
<owl:Class rdf:ID=”vehicle”>
</owl:Class>

<owl:Class rdf:ID=”car”>
      <rdfs:subClassOf rdf:resource = “#vehicle”/>
</owl:Class>

<owl:Class rdf:ID=”boat”>
      <rdfs:subClassOf rdf:resource = “#vehicle”/>
</owl:Class>

<owl:Class rdf:ID=”aeroplane”>
      <rdfs:subClassOf rdf:resource = “#vehicle”/>
</owl:Class>
```

The OWL language is designed primarily to be computer interpretable. To develop a taxonomy in OWL would require a suitable editing tool. There are a number of free tools, but Protégé seems to be the most widely used:



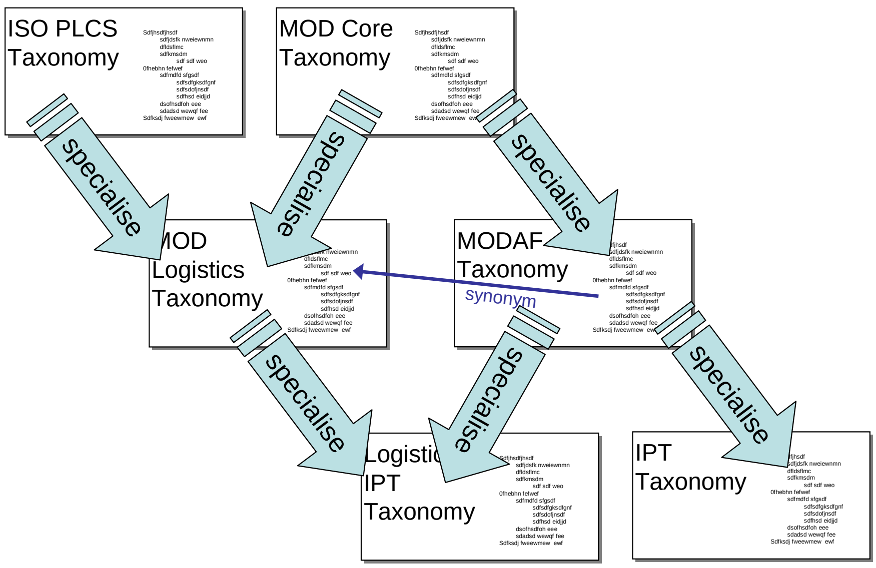
One of the main strengths of OWL is that it was designed to be the key enabler of the "semantic web" efforts underway at the W3C. This means that the language is intended to deal with classes that are distributed across networks. Hence it is ideal for specifying taxonomies that extend and relate to other taxonomies. Also, as each class in an OWL taxonomy is identified by its web address (a URI) and ID, it is a simple matter to refer to the classes from a MODAF ERM exchange file. The example below shows a (mocked-up) MODAF XMI exchange file which contains two systems:

```
<UML:Class xmi.id="sys1" name="TLAM">
  <UML:ModelElement.stereotype>
    <UML:Stereotype xmi.idref="systemstereotype"/>
  </UML:ModelElement.stereotype>
  <UML:ModelElement.taggedValue>
    <UML:TaggedValue xmi.id="taxonomyRef" isSpecification="false">
      <UML:TaggedValue.dataValue>
         http://taxonomy.mod.uk/#weapon_system
      </UML:TaggedValue.dataValue>
    </UML:TaggedValue>
  </UML:ModelElement.taggedValue>
</UML:Class>
<UML:Stereotype xmi.id="systemstereotype" name="system">
  <UML:Stereotype.baseClass>Class</UML:Stereotype.baseClass>
</UML:Stereotype>
```

The class data element represents a system (TLAM), and has a "taxonomyRef" tagged value which refers to the URI of the taxonomy element that describes it. This ability to uniquely reference OWL classes in a taxonomy allows relationships to be established within and between taxonomies. The simplest case is specialisation:

```
<owl:Class rdf:ID="saloon_car">
      <rdfs:subClassOf rdf:resource = "http://taxonomy.mod.uk#car"/>
</owl:Class>
```

In this case, the class "car" that is being specialised may be in another taxonomy, or in the same one. Using this capability, it is easy to build a distributed taxonomy:



This offers a solution to the problem that many large organizations face – competing and overlapping taxonomies. Provided that each taxonomy developer is prepared to use OWL, it is possible to allow each developer to maintain their own taxonomy. In cases where disagreements on terminology exist, OWL also provides a technical solution – allowing equivalent classes to be identified. Human intervention is needed when identifying these cases, however.

OWL, and the RDF standard it's based on are both XML specifications. This means that XML stylesheets (XSLt) can be developed which enable publication of the taxonomy in document and web formats. There are some technical issues with using XSLt with RDF due to there being many ways to structure the RDF definitions. These can be overcome by specifying regularised ways of defining the RDF.

Other Approaches for Taxonomies

The Lightweight Directory Access Protocol (LDAP) was developed to provide a way to look up directory entries for e-mail addresses. LDAP servers can provide hierarchical directory structures which could easily be used to describe a taxonomy. Each element in the directory can have properties associated with it, which allows for additional information (e.g. definitions) to be added. LDAP has the advantage of being robust technology, and it provides excellent performance. However, the protocol is somewhat low-level, and although editing and authoring tools exist, these are mostly aimed at software developers. LDAP itself is a protocol used for querying directory servers (it's also used by the servers in their response), so does not lend itself well to document publication.

Another approach is to use a relational database containing tables of taxonomy entries. Relationships in the database can be used to define specialisations and other connections between elements. Implementing the database would be a simple task, and web-based user interfaces could be developed for browsing and editing the taxonomy. Additionally, web services could be implemented to allow programmatic access to the taxonomy. The use of a relational database offers performance, scalability and the possibility to provide multiple representations of the taxonomy – e.g. OWL, HTML, and through an interactive user interface. It is more of a complimentary solution than an alternative – even an OWL based solution is likely to use a relational database at its core.

An XML schema could be developed to specify the structure of the taxonomy. The advantage of this is that the developer has complete control over the content and structure of the taxonomy. Unfortunately, the use of a bespoke XML taxonomy means that editing tools either have to be developed from scratch or tailored from existing XML editors. The freedom to develop a specialist taxonomy schema for MODAF may cause some problems of compatibility with the rest of the MOD and other organizations – it is usually easier to persuade someone to adopt an open standard than to adopt another's internal standard.

There are other data/meta-data specification standards that offer sufficient features to represent a taxonomy. For example XMI, the OMG interchange standard for UML could be used to exchange class specialisation structures, with relationships between the classes, properties, descriptions, etc. However, XMI brings with it a lot of baggage that is associated with UML and that is not needed to define a taxonomy. XMI is a complex XML structure and is barely readable to all but the most advanced UML expert. Other possibilities include IDL, the Interface Definition Language which allows for class hierarchies to be defined. IDL is strictly speaking for software developers (the syntax is similar to C++) so probably wouldn't be ideal for this work. Other modelling languages exist, such as ISO 10303-11 (EXPRESS), but there is no built-in capability to reference remote taxonomies.

Conclusions and Recommendations

This paper has detailed an approach for implementing the MODAF taxonomy based on the W3C OWL (web ontology language). Specifying the MODAF taxonomy in OWL opens up the possibility to draw upon taxonomies developed by other organizations, provided they are also represented using OWL. It also allows the different IPTs to specialise the core MODAF taxonomy for their own project needs.

Other approaches have also been considered, but none offer the flexibility and simplicity of OWL. It is clear that there is a lot to learn from other implementation approaches, however. If the taxonomy is to be used in an online scenario, providing dictionary support to MODAF applications, there is a requirement to serve the data. This will inevitably involve some form of database and web services.

Some work has already been carried out by MOD (DLO CTS) in implementing an OWL server to provide the online taxonomy for PLCS (Product Lifecycle Support) data exchanges. This approach has the support of a number of defence contractors and consultants working in the area. In addition, IA6 has been investigating possible applications of OWL in other areas of defence. This background of expertise and available technology helps to smooth the introduction of OWL for MODAF taxonomies.

Acknowledgements

Thanks go to the reviewers:

- Andy North, IA
- Fariba Hozhabrafkan, Cornwell Consulting