ONTOLOGIES AND THEIR APPLICATION IN MODAF

This document is intended to provide a brief overview of the concepts surrounding the use of ontologies and then to show how they may support a MODAF based architecting process.

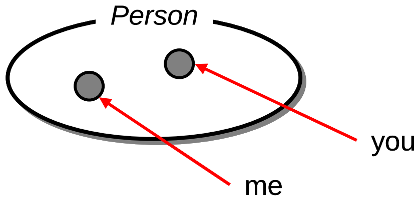
What is an Ontology? [1]

Ontology is the study of what exists. As a discipline, it has been honed over thousands of years by the finest minds in philosophy and mathematics. In recent years, the topic of ontology has come to some prominence in the domain of software engineering. The formal principles of ontology used by mathematicians and logicians have been shown to be useful in enabling software systems to better represent the physical world, and so more accurately support users' requirements.

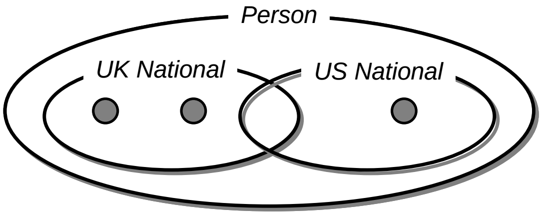
Developing a proper formal ontology is not a task to be taken lightly. New ontologies appear daily on the web, which are little more than traditional data models represented in OWL (the W3C's Web Ontology Language). Where true ontologies do exist, they are the result of years of hard work by academics and software professionals. Good examples are SUMO , ISO15926 and Dolce.

The main benefit of an ontology for an organization like the MOD is that, if properly designed, it can offer great benefits of interoperability. This is because a true, formal ontology aims to describe what exists rather than what is perceived – in other words, it is not slanted towards any particular stakeholder's view of the world. This is sometimes described as a "view from nowhere", and it is this feature that makes an ontology particularly useful in enabling parties with very different views to come to an agreement on meaning. This feature is also the reason that ontologies are difficult to develop properly – each new term in the ontology must be adequately analysed to assess its true meaning and establish how it relates to other parts of the ontology. Great care has to be taken not to "model the entire world", and a practical ontology should remain focussed on the domain it is to support.

Most formal ontology development is based around set theory or similar branches of mathematics and logic such as category theory or type theory. The fundamental components of an ontology are classes and individuals. Individuals are things which have spatial and temporal extent: me, you, the computer you're using, the Eiffel Tower, etc. Classes are categories of things: people, organizations, computers, monuments, etc. This is easily illustrated with a Venn Diagram:



Aside from these fundamental concepts, there are a number of important relationships. The first to consider is the relationship between class and individual (as illustrated in the previous diagram). The second is specialisation; ie one class being a subset of another:

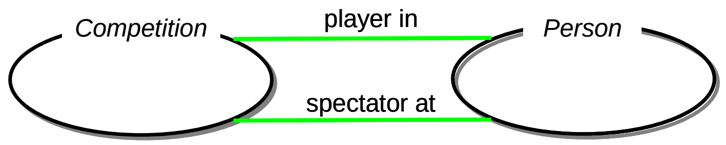


The example above shows that there are two subsets (specialisations) of person which are UK nationals and US nationals. Note that the sets overlap to cover the case of dual nationality.

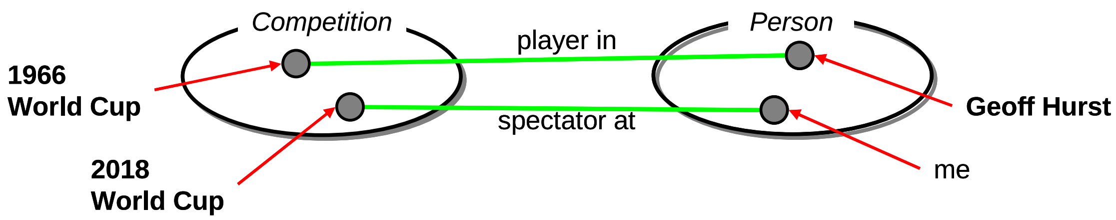
[1] This section was first published in a report on Ontology for the Information Coherence Authority for Defence (ICAD) and the Integration Authority (IA) (now known as the System Engineering Integration Group, the SEIG).

Mistaking class-individual for specialisation is a common mistake; you are not a type of person, you are a person. This degree of semantic precision is essential in developing an ontology and is at the root of what makes an ontology useful and also what makes it difficult to develop.

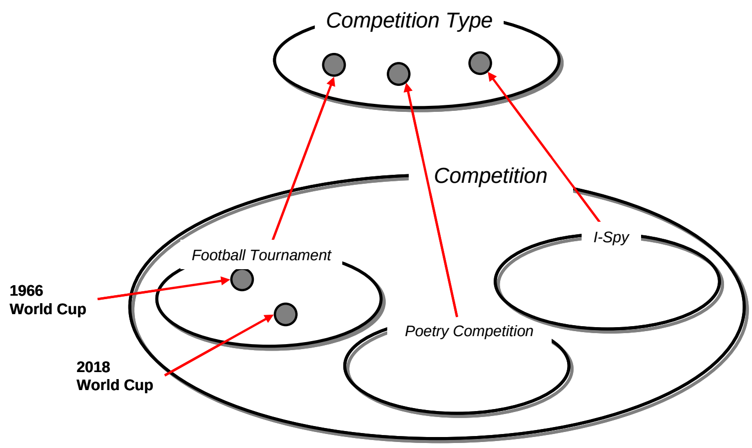
Other than class-individual and specialisation, there is the general case of relationships between classes (sometimes called predicates). These establish common patterns in the ontology, such as:



The example above establishes two relationships that assert people can both play in competitions and be spectators at them. These relationships are classes themselves, that is, there can be actual relationships between individuals:



Another important aspect of formal ontology development is the fact that classes can be classified; it's not just individuals that belong to classes. This is often ignored by ontology developers but it has to be realised that a very large proportion of the information that businesses work with is classification information. To be able to manage this information properly, it has to be classified; ie classification of classes. The example below shows members of the class competition type which are themselves classes:



Any ontology that is to be useful must deal with classes of classes. In formal terms, this is called a higher-order ontology (an ontology which only has classes whose members are individuals is first-order). A practical problem of software implementation exists with higher-order ontologies; a machine reasoner is not guaranteed to resolve an answer from a higher-order ontology in a finite amount of time. This presents something of a dilemma to ontology developers, the real world is higher-order but the reasoners and inference engines can only practically work with first-order ontologies. Two solutions are possible. The first is to develop a higher-order ontology and implement without using reasoners (the commercial benefits of reasoning and inference are still as yet confined to niche applications). The second is to "compress" the higher-order concepts into a first-order framework (this usually means replacing the class-individual relationships with simple predicates).

What is the MODAF Ontology?

Although the MODAF Meta Model describes generic types of architectural information and their relationships, if re-use and integration of architectural products is required, those products must also utilise a common terminology and library of standard elements across architectures. The MODAF Ontology serves this purpose and ensures that each instance of an architectural element (organisation, system, activity, etc) uses a commonly agreed and shared definition for its name. By providing a standard set of terminology and reference data, the MODAF Ontology supports:

- Architectural coherence across the MOD. This is achieved through ensuring all MODAF users employ the same terminology to describe the elements in their Architectures.
- Architectural comparison. Using the same base definitions for standard organisations, systems, activities, etc allows comparison of different aspects of the business.
- Data exchange clarity. Information exchanged between architectural tools can be fully defined using the MODAF Ontology.

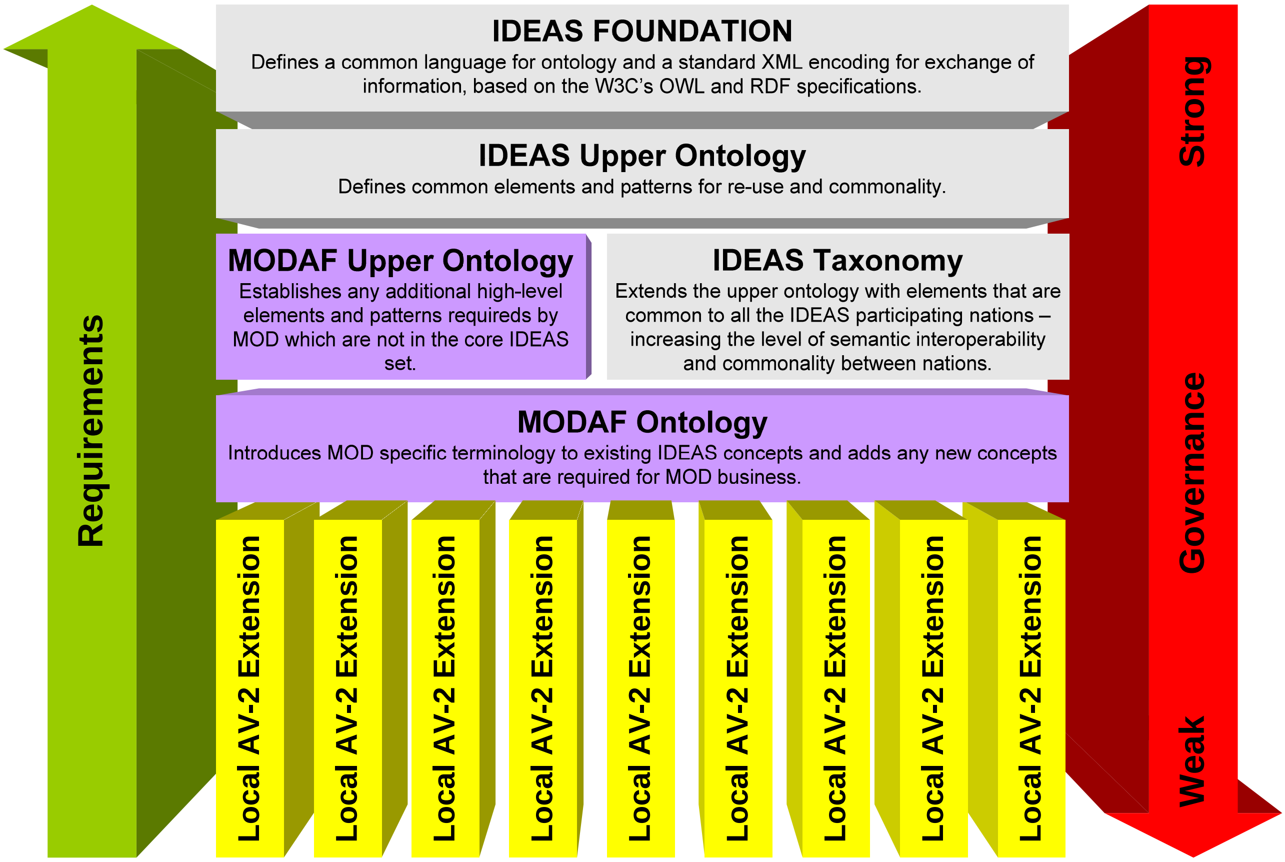
At the time of publishing the MODAF v1.1 document, the MODAF Ontology was at the feasibility stage. ICAD (Information Coherence Authority for Defence – part of DG Info) is responsible for the MODAF Ontology. The approach being investigated is based on the following premises:

- The BORO Methodology[2] is to be used in developing the Ontology
- The MODAF Ontology will extend the IDEAS[3] model – an AUSCANUKUS effort to develop a common ontology for defence enterprise architecture
- The UK Defence Taxonomy and Thesaurus will be the starting point for development, with inputs from other sources of reference data such as the Defence Data Repository, PLCS, BMS, etc.

A governance regime for the Ontology is yet to be formalised, but will be based on a tiered principle, with IDEAS at the top and individual architectures' AV-2s at the bottom:

[2] BORO consulting are currently working on the construction of the Core Enterprise Ontology (CEO) - an 'industrial strength' ontology to be used as a tool by enterprises to significantly improve the semantic aspects of their information systems.

[3] The International Defence Enterprise Architecture Specification for exchange. The purpose of the project is to develop a data exchange format for military Enterprise Architectures to allow seamless sharing of architectures between the partner nations.

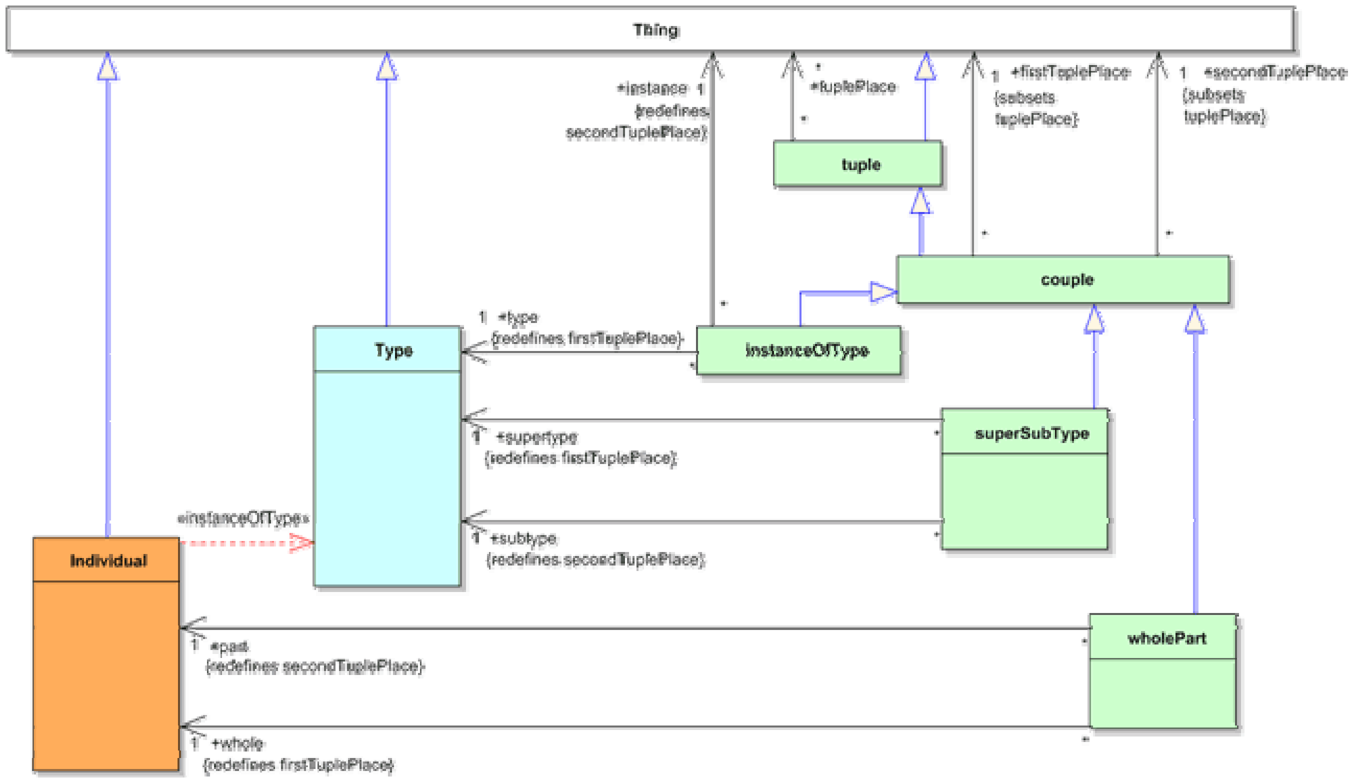


This model allows architects the freedom to make local extensions to the ontology provided the extension is made formally in an AV-2 and linked back to the ontology. The upper layers are, however, more strictly governed. For example, a change to the IDEAS Foundation would require consensus from the four nations and a change to the MODAF ontology would go through the appropriate MOD information management body (eg ICAD).

It is in the nature of architectures that they tend to reference each other and use common elements. If an architect introduces a new element (eg a new system) then it is likely that it may need to be referenced in other architectures. For this reason, there needs to be a way to migrate elements up the tiers as time goes by; this could even happen at the national level, where more than one nation adopts a new technology or way of working.

The approach being taken in IDEAS allows multiple names (with context) to be applied to all the elements. In other words, elements are created once, but may be named several times by different stakeholders. This mechanism allows communities, nations, etc. to work with their favoured terminology but still allows for commonality between the parties.

The IDEAS model is still in development. In the mean-time, some early parts of the IDEAS Foundation have been made public (but these may change before formal publication):



The Key Elements of the IDEAS Foundation

The suite of MODAF documents on this web site also includes a glossary and acronym list. Whilst these will be consistent with the Ontology, their purpose is only to support the readability of the MODAF documents themselves.