Peter F Brown
Independent Consultant

Note on Methodology

Some comments and guidelines for the development of the OASIS Transformational Government Framework

"Far buon vino è semplice ma non facile"

Update: 3 January 2011

Peter F Brown

Purpose

This note has been prepared in order to render explicit some of the modelling and methodology that will be used in the development of work deliverables for the OASIS Transformational Government Framework

Background and Rationale

The Transformational Government Framework Technical Committee has been set up within OASIS with a view to develop a series of deliverables that together form a framework for 'transformational government'.

At the first face-to-face meeting of the committee (Washington, D.C., 10 Dec 2010), an agreed starting point was to establish a common terminology and associated definitions that would be used consistently across all proposed deliverables.

This work is non-trivial. The principles involved in putting together a common terminology are relatively straightforward but actually capturing those terms and agreeing definitions can involve considerable effort – "Far buon vino è semplice ma non facile" ("Making good wine is simple but not easy"). In the vast work and literature on eGovernment, many terms are used, often loosely, often with implicit definitions and, most often, used inconsistently across different problem areas. One of the goals already set out by the committee has been to reduce ambiguity to an absolute minimum. This is particularly important given the international context of the work. Concepts that are clearly and explicitly defined are more easily translated and transposed into different geo-political contexts.

The objective of this document is to underline some of the methods and modelling principles that the TC editors intend to deploy in the committee's work and that can be applied to all aspects of the development of the committee's deliverables.

Some core terminology

So that we are clear from the get-go, we will use certain key words as they are explicitly defined in "*Terminology work* – Vocabulary – Part 1: *Theory and application*" [ISO 1087-1:2000]. A **concept** is a distinct unit of meaning or knowledge. It should not be confused with a **term**, which is a specific (usually language dependent) representation or expression of the concept. In everyday work, we use terms – 'governance', 'service', 'citizen' – as common, often implicitly accepted labels for concepts. The concept is the abstract mental idea (which should be universal[1] and language independent) to which the term gives a material expression in a specific language. Particularly in an

[1] but are frequently culture-dependent

international environment such as global standardization initiatives, the distinction is important as it is common *concepts* that we wish to work with, not common *terms*[2].

This distinction also helps avoid common modelling pitfalls. Terms that may seem similar or the same across two or more languages may actually refer to different concepts; or a single term in one language could be understood to refer to more than one concept which another language expresses with discrete terms:

- the English term 'sensible' does not refer to the same concept as the French word 'sensible';
- The English term 'service' can refer to an organisational unit ('Passport Service') or something that is performed by one for another ('a dry cleaning service'), whereas distinct terms are used in German ('Dienst' or 'Dienstleistung').

In order to avoid ambiguity in understanding a particular term, we associate an explicit **definition** with each concept, as we do in a dictionary. As well as minimising misunderstandings within a particular language, this helps us also identify the most appropriate term for that concept in other languages.

In many subject domains or 'information territories', relevant concepts are organised with regard to their relationships to each other in a **taxonomy**. Taxonomies come is a range of forms, non-hierarchical (such as a controlled vocabulary) or hierarchical (such as classification schemes and thesauri). Hierarchical taxonomies organise concepts from the broadest to the most specific, using relationships such as 'broader than', 'narrower than', 'part of (aggregation)', 'part of (composition)', etc.

More recently, many organisations are looking at a broader understanding of the relationship between concepts, recognising that many concepts are related to each other in multiple ways. This is a starting point for developing an **ontology**. In information science, an ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It can be used to describe the domain (the coverage should be sufficiently comprehensive to include all concepts relevant to the domain) and to reason about the domain.

Our objective is not to build a formal ontology but to be sufficiently clear in our distinctions, definitions and relationships between concepts that the various deliverables created will use consistent terminology thus establishing an internally coherent set. They should also be clear enough that subsequent ontology development is possible if so desired.

Modelling methodology

The TC will employ "common sense" use of terms wherever possible. However, dangers of ambiguity abound and therefore a range of concepts are defined explicitly and the preferred terms indicated.

[2] This is central to all multi-lingual thesauri, where the core item of organisation is the concept, not the term.

We will use an escalation method in order to satisfy our needs, using increasingly complex models as needed, and starting with establishing agreed terminology.

Terminology

We will be looking at terms as employed in a range of reference documents (including all contributions to the TC) and determining whether they adequately distinguish and represent the respective concepts of our domain. The steps followed, implicitly or explicitly include:

- Distinguishing a "candidate concept";
- Describing the concept (in formal terms, in order to identify its "unique combination of characteristics");
- Defining it by providing a descriptive statement which serves to differentiate it from related concepts; and finally
- Naming it – providing the preferred term in English to designate the concept

Concept Maps

The next step available will be to express, very loosely, the relationship between the concepts in the domain, using **concept map**. Although there is no normative convention for creating or interpreting concept maps, they generally consist of a set of concepts with lines indicating that two concepts are related in some way. The concept map cannot provide a formalisation of the relationships between concepts but this is also why it can be a valuable tool at an early stage of modelling, giving stakeholders some leeway while capturing essential relationships.

Topic Maps

An easy to use and expressive way to capture concept types/classes and named relationship types is the ISO 'Topic Maps' standard [ISO 13250:2000]. The standard has the advantage of being easy to use while encouraging good modelling practices. It incorporates a modelling standard, the Topic Map Constraint Language TMCL) that allows us to model concept classes (or 'topic types') and relationship classes (or 'association types') and explicitly constrain which types of relationship are 'allowable' between any two concept types.

UML

The Unified Modeling Language, a popular software engineering modelling language consists of a series of structure and behaviour diagram types that can capture different types of information relevant for any given project.

Of particular interest to us could be:

- Class diagrams, to capture concepts and relationships between them;
- Component diagrams, to capture the components making up a software system and the relationships between them;
- Activity diagrams, to capture the flow of control between components in any system;
- State machine diagrams, to describe the states and state transitions of a system;
- Use case diagrams, to describe system functionality in terms of users and their goals

4 The last two – Topic Maps and the UML – are **tractable**, in the sense that they can be directly used to develop services and systems.