



**ADMS.F/OSS**

**JOINING UP GOVERNMENTS**



Property	Value
Release date	21/02/2012
Status:	Draft
Version:	0.01
Authors:	Phil Archer W3C
Reviewed by:	
Approved by:	

[illegible]

## TABLE OF CONTENTS

1. Introduction.....	3
2. Conformance Statement .....	4
3. Namespaces .....	4
4. Conceptual Model .....	7
4.1 The Software Asset Class .....	7
4.2 ADMS Asset Relationships and Properties inherited by ADMS.F/OSS Software Asset Class7	
4.3 Technical Description of the Software .....	9
4.3.1 Topic .....	10
4.3.2 Intended Audience.....	10
4.3.3 Locale .....	10
4.3.4 User Interface Type .....	10
4.3.5 Programming Language .....	11
4.3.6 Operating System .....	11
4.4 Usage and Assessment.....	11
4.4.1 Used By .....	11
4.4.2 Assessment and Assesses.....	12
4.4.3 Metrics .....	12
4.4.4 The Assessment Class .....	12
4.4.5 The Metrics Class .....	13
4.5 The Project and Community .....	15
4.6 Functional Classification for the Public Sector .....	16
4.7 Languages .....	16
5. Properties Considered and Excluded .....	17
6. The RDF Schema (To Do) .....	17
7. The XML Schema (To Do) .....	17
8. Usage Guidelines .....	17
9. Background and Objectives .....	18
10. Approach & Community .....	19
11. Change Control .....	19
12. References .....	19

## 1. INTRODUCTION

The Asset Description Metadata Schema [ADMS], is a vocabulary for describing Semantic Assets, defined as a collection of reference data items that is used for eGovernment metadata, the sharing of which among administrations contributes to increased interoperability across organisational and geographic boundaries. This broad definition covers specifications, guideline

documents, metadata schemas, code lists, controlled vocabularies, and references to various types of entities in the real world, such as organisations, people and places.

ADMS.F/OSS (ADMS for Free and Open Source Software) is an extension to ADMS to encompass software, typically made available through a catalogue known as a 'software forge.' Like ADMS, ADMS.F/OSS has been created under Action 1.1 of the ISA Programme [A11]. Further background is available in "Towards Open Government Metadata" [TOGM] which offers an overview and context for the work.

The intention is not to create a new vocabulary, but to identify and re-use existing methods for describing software assets. In particular, ADMS.F/OSS draws on the following work:

@@@ Insert details from Related Work page@@@

## 2.CONFORMANCE STATEMENT

A publisher using the vocabularies can choose whether to publish using either RDF or XML as their technology and may use any of the terms defined in this document.

A consumer of data published using the vocabularies must understand all the terms defined below in one of three ways which should be declared when claiming conformance:

RDF conformance, meaning that RDF data published using any term in the vocabularies will be consumed and processed accordingly;

XML conformance, meaning that XML data published using any term in the vocabularies will be consumed and processed accordingly;

XML and RDF conformance, meaning that both RDF and XML data published using any term in the vocabularies will be consumed and processed accordingly.

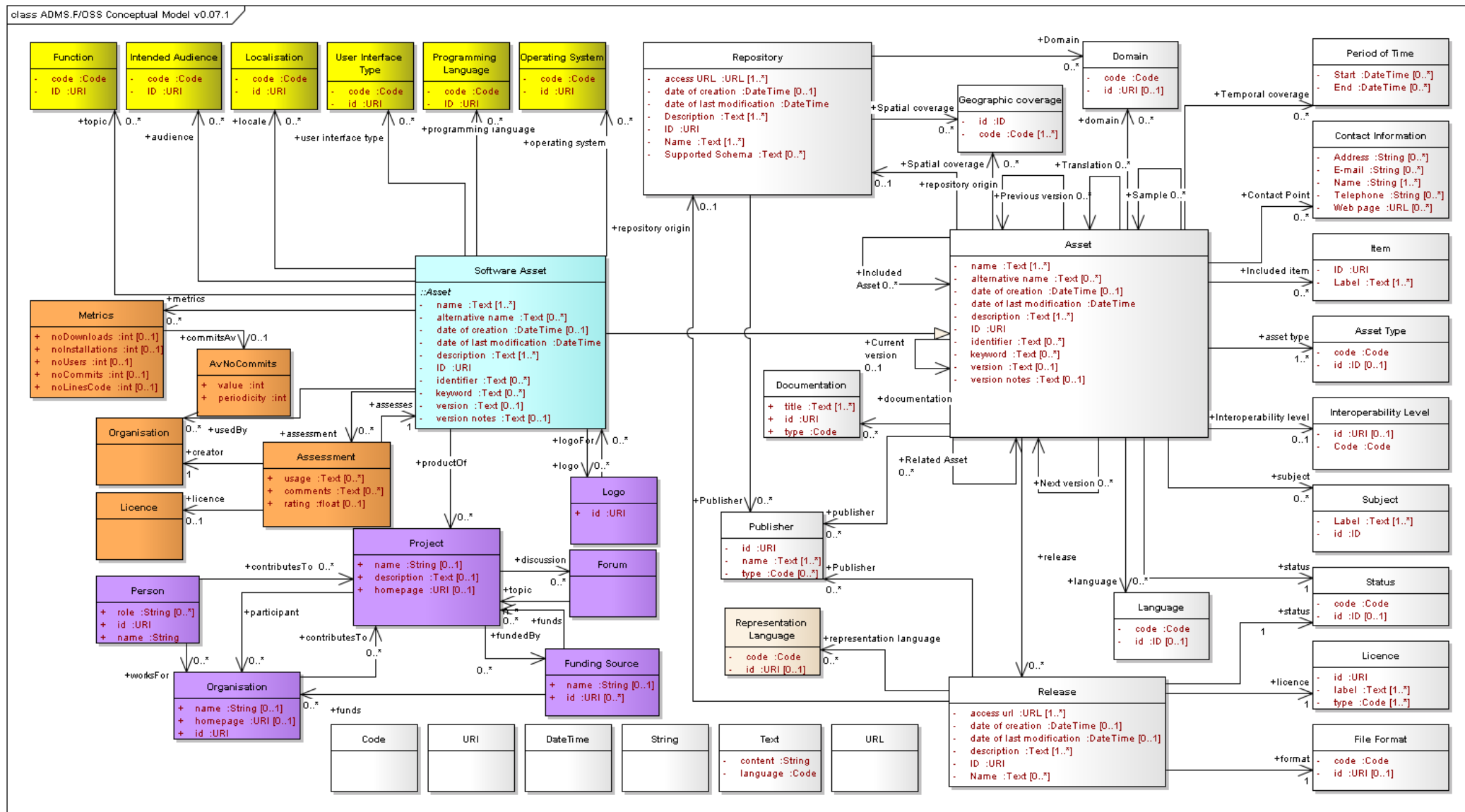
## 3.NAMESPACES

At the time of publication, an active discussion is ongoing concerning the namespace to be used for ADMS and ADMS.F/OSS. Factors such as long term stability and change control are paramount. For the time being, we are simply using 'example.org' as a place holder, to be replaced in the near future.

With that in mind, we define the namespaces and suggested prefixes for ADMS and ADMS.F/OSS as simply:



Prefix	Namespace
adms	<code>http://example.org/ns/adms#</code>
admsf	<code>http://example.org/ns/admsf#</code>



Key: Grey - Classes inherited from ADMS. Light Blue - the software asset itself. Purple - the project, people and funding behind the software asset. Light brown - usage and assessment. Yellow - what the software is for, technical aspects etc.

## 4. CONCEPTUAL MODEL

The conceptual model presented in Figure 1 is independent of any technology that may be used to represent it. It describes an extension to ADMS that provides the minimal set of classes, relationships and properties necessary to describe software assets.

This specification is arranged in sections as reflected in the colour coding used in Figure 1.

After introducing the Software Asset Class, it reviews the ADMS relationships and properties providing notes designed to interpret them in the context of ADMS.F/OSS (section 4.2). The document then works through the relationships, secondary classes and properties in three sections:

- Technical description of the software asset - what is it designed to do, what operating system and programming language is used etc. (coloured yellow in Figure 1 and described in Section 4.3)
- Usage and Assessment - who uses the software and what comments have they made about it (coloured light brown in Figure 1)
- The project, people and funding behind the software (coloured purple in Figure 1)

### 4.1 THE SOFTWARE ASSET CLASS

The Software Asset class is the key class for ADMS.F/OSS. It is a sub class of the ADMS Asset class and therefore inherits all the latter's properties and relationships. These are defined in the ADMS specification and are summarised in the following section with notes on their use within the ADMS.F/OSS context.

### 4.2 ADMS ASSET RELATIONSHIPS AND PROPERTIES INHERITED BY ADMS.F/OSS SOFTWARE ASSET CLASS

Relationship	Class	Cardinality	Notes (where applicable)
repositoryOrigin	Repository	[0..1]	The forge
release	Release	[0..*]	A software package that can be downloaded
spatialCoverage	Geographic Coverage	[0..1]	The geographic locations for which the software is applicable. This is likely to be left unused in ADMS.F/OSS

domain	Domain	[0..*]	The public sector for which the software is relevant.
assetType	Asset Type	[1..*]	This relationship is mandatory in ADMS. See Section 4.3.3 for a controlled vocabulary
language	Language	[0..*]	This may not be relevant to a software asset for which there are separate relationships for programming language (4.3.5) and locale (4.3.3)
publisher	Publisher	[0..*]	
subject	Subject	[0..*]	This may be relevant but note the provision of the topic/Function relationship and class (4.3.1)
status	Status	[0..*]	Alpha, beta, RC etc. @@@Controlled voc for this? @@
relatedAsset	Asset	[0..*]	
documentation	Documentation	[0..*]	This may include screenshots and videos as well as textual documentation.

**Table 1 ADMS relationships inherited by the Software Asset Class**

Property	Data type	Cardinality	Notes
name	text	[1..*]	The software must have a name. See section 4.7 for notes multiple languages
alternativeName	text	[0..*]	Any number of alternative names may be supplied.
dateOfCreation	dateTime	[0..1]	Dates (and time if relevant) should be conformant with ISO



			8601:2004. All xsd date and time formats meet this criteria.
dateOfLastModification	dateTime	[0..*]	
description	text	[1..*]	A free text description of the software must be provided.
ID	URI	[1..1]	Each software asset must have a URI.
identifier	string	[0..*]	Any number of identifiers, whether URIs or not, may also be assigned to the software asset.
keyword	text	[0..*]	
version	string	[0..1]	

**Table 2 ADMS properties inherited by the Software Asset Class**

## 4.3 TECHNICAL DESCRIPTION OF THE SOFTWARE

ADMS.F/OSS uses common terms to describe what a software asset does, its intended audience and the key technical parameters of programming language and operating system. Following the ADMS approach, for each relationship the associated class has two properties:

- a code - a value from a controlled vocabulary
- a URI.

Either or both properties may be used.

There are two terms within ADMS Core that are worthy of highlight in this respect too:

- licence - ADMS provides a licence relationship between a Release and a Licence;
- development status - this is covered by the ADMS relationship of status (see Table 1).

### 4.3.1 Topic

Attribute	Abstract Data Type	Cardinality
topic	Function	[0..*]

The topic relationship associates a Software Asset with its Function. The @@@TBA@@@ vocabulary provides a suitable classification system but further options tailored to the public sector are provided in section 4.6.

It should be noted that a software asset may be associated with any number of Function classes via the topic relationship.

### 4.3.2 Intended Audience

Attribute	Abstract Data Type	Cardinality
audience	Intended Audience	[0..*]

@@@ rely on the @@@ classification @@@

### 4.3.3 Locale

Attribute	Abstract Data Type	Cardinality
locale	Localisation	[0..*]

@@@ADMS uses RFC 3066 for this (the familiar en, es-mx etc. codes) and their associated DBpedia URIs for ID (check with ADMS discussion - there's been a lot of it)@@@

### 4.3.4 User Interface Type

Attribute	Abstract Data Type	Cardinality
uiType	User Interface Type	[0..*]

@@@ Rely on @@@ controlled vocabulary

### 4.3.5 Programming Language

Attribute	Abstract Data Type	Cardinality
programmingLanguage	ProgrammingLanguage	[0..*]

@@@ Rely on @@@ controlled vocabulary

### 4.3.6 Operating System

Attribute	Abstract Data Type	Cardinality
os	Operating System	[0..*]

@@@ Rely on @@@ controlled vocabulary

## 4.4 USAGE AND ASSESSMENT

ADMS records three distinct types of data that can be used to assess the suitability of a software asset:

- who uses it;
- what those users think of it;
- key metrics about the project.

Relationships exist between a Software Asset and classes that represent all three of these.

### 4.4.1 Used By

Attribute	Abstract Data Type	Cardinality
usedBy	Organisation	[0..*]

An important piece of information for anyone considering using a piece of software is "who else uses this?" In the context of the public sector, the important information is likely to be which other organisation use this software. The usedBy relationship associates a Software Asset with any organisation that uses it.

The Organisation class is not defined in ADMS.F/OSS as several vocabularies already exist for this purpose, notably Friend of a Friend [FOAF] and the Organization Ontology [ORG].

## 4.4.2 Assessment and Assesses

Attribute	Abstract Data Type	Cardinality
assessment	Assessment	[0..*]
assesses	Software Asset	[1..*]

assessment and assesses are inverse relationships that associate a Software Asset with an Assessment class (see section 4.4.4). A Software Asset may be connected to any number of assessments and each assessment must be connected to at least one Software Asset that it assesses. A single assessment that describes the experiences of using more than one Software Asset is perfectly acceptable.

## 4.4.3 Metrics

Attribute	Abstract Data Type	Cardinality
metrics	Metrics	[0..*]

metrics associates a Software Asset with a Metrics class (see section 4.4.5).

@@@What is the correct cardinality for metrics? (a software asset may be available from more than one forge...)

## 4.4.4 The Assessment Class

This represents a review of the software and has a number of relationships and properties. The Dublin Core creator relationship is used to link such a review to the organisation that provided it which should also be associated with the Software Asset via the usedBy relationship (see beginning section 4.4).

### Usage

Attribute	Abstract Data Type	Cardinality
usage	Text	[0..*]

A free text description of what the software was used for. This should be an objective description of the original intention of the user organisation at the time of selection and installation.

## Comments

Attribute	Abstract Data Type	Cardinality
comments	Text	[0..*]

A free text description of the experience of using the software. This should be a subjective description, ideally giving details of where the software was good and bad.

## Rating

Attribute	Abstract Data Type	Cardinality
rating	string	[0..*]

A rating for the software. Typically this will be given using a 5 star rating scale where 1 is poor and 5 is excellent.

@@@Do we want to tighten this and specify a 5 star scheme?? @@@

## Licence

Attribute	Abstract Data Type	Cardinality
licence	Licence	[0..*]

The licence relationship associates an Assessment with information about whether and how that Assessment may be published. The Licence class itself is not defined as part of ADMS.F/OSS.

@@@ More detail to add here?? @@@

### 4.4.5 The Metrics Class

The properties of this class record a variety of objective facts about the software. It is associated with a Software Asset via the metrics relationship.

@@@ Should we say anything about how numbers should be captured? @@@

### Number of Downloads

Attribute	Abstract Data Type	Cardinality
noDownloads	integer	[0..*]

The number of times a software asset has been downloaded.

### Number of Installations

Attribute	Abstract Data Type	Cardinality
noInstallations	integer	[0..*]

The number of times a software asset has been installed.

### Number of Users

Attribute	Abstract Data Type	Cardinality
noUsers	integer	[0..*]

The number of users of a software asset.

### Number of Commits

Attribute	Abstract Data Type	Cardinality
noCommits	integer	[0..*]

The number of times code for the asset has been committed to the forge.

### Number of Lines of Code

Attribute	Abstract Data Type	Cardinality
-----------	--------------------	-------------

noLinesCode	integer	[0..*]
-------------	---------	--------

The number of lines of code within the asset (not including any dependencies).

## Commit Average

Attribute	Abstract Data Type	Cardinality
commitAv	AvNoCommits	[0..*]

The commitAv relationship associates a Metrics Class with an AvNoCommits Class.

## Average Number of Commits Class

This class has two properties: the average number of commits itself and the time period over which the average is calculated.

Attribute	Abstract Data Type	Cardinality
value	integer	[1..1]
periodicity	string	[1..1]

Periodicity is recorded using one of the following values:

daily

weekly

monthly

yearly

@@@ Need to check this is sensible!! @@@

## 4.5 THE PROJECT AND COMMUNITY

By its very nature, open source software is a collaborative effort. ADMS.F/OSS provides classes and relationships to describe the project that created the software, the participating organisations and funding sources.

## 4.6 FUNCTIONAL CLASSIFICATION FOR THE PUBLIC SECTOR

As noted in section 4.3.1, ADMS.F/OSS provides a *topic* relationship that associates a software asset with its function. @@@ provides a generic classification system for any software and the terms there are likely to be applicable. However, for the public sector, we offer the following additional terms which were developed by Centro de Transferencia de Tecnología [CTT].

It should be noted that a software asset may be associated with any number of Function classes via the topic relationship. Where the @@@ and CTT classifications overlap, provide both.

@@@ provide example showing that the string in the left hand column is a value for the code property @@@

	Examples
<b>Web sites and virtual offices</b>	
<b>Citizens Attention</b>	Integrated offices, information phone lines, citizens folder, where is my transaction with the administration.
<b>Web Tools</b>	Searches, forum, geo-reference, etc
<b>Electronic Processing</b>	The applications that perform the electronic processing such us registries, management of grants, etc.
<b>Support to Electronic Processing</b>	Products and services that makes possible electronic management such us digital signature, exchange of data, payment gateway, etc.
<b>Management for internal Procedures</b>	Requests for materials, room reservations, etc.
<b>Management of Finances</b>	Procurement, budgets, comptroller
<b>Management of Human Resources</b>	Personnel management, payroll, time control, training, vacation
<b>Management of knowledge and support to taking decisions</b>	Content Managers, dashboards, data ware house, etc
<b>Infrastructure for Communications</b>	Networks and management tools
<b>Infrastructure for Security</b>	DMZ, proxies, DNS, IPS, backup, antivirus, etc
<b>Infrastructure for Messaging</b>	email, chat, Twitter, social networks, etc
<b>Infrastructure for Managing Identities</b>	Digital signatures, LDAP, PKI, identity management, etc
<b>Services and Systems Management</b>	Monitoring, statistics, managing of data processing centres managing of request of users, of bugs, service interruptions
<b>Development and running Platforms</b>	
<b>Desktop Environment</b>	Applications, virtualizations, models for PCs
<b>Normalisation and Regularisation</b>	Methodologies, recommendations, specifications, etc.
<b>Common Services for Public Administrations</b>	Common services offered, generally for free, to the rest of public administrations

## 4.7 LANGUAGES

@@@ Needs updating for ADMS.F/OSS as this example is from the core vocabularies@@@



Where data such as names exist in multiple languages, each version of the data should be included and each one associated with the relevant language identifier. RFC 3066 [RFC 3066] provides a commonly used set of identifiers for natural languages. This is the set recognised by UN/CEFACT and XML Schema.

Languages are represented by two character codes, optionally followed by a locale definition such as "de" meaning German and "de-at" meaning "German as spoken in Austria."

XML Example:

```
<Location>
  <geographicName xml:lang="en">London</geographicName>
  <geographicName xml:lang="fr">Londres</geographicName>
</Location>
```

RDF Example:

```
[] a locn:Location ;
   locn:geographicName "London"@en ;
   locn:geographicName "Londres"@fr .
```

## 5. PROPERTIES CONSIDERED AND EXCLUDED

## 6. THE RDF SCHEMA (TO DO)

To follow.

## 7. THE XML SCHEMA (TO DO)

Most, if not all the properties listed in this specification exist in the UN/CEFACT CCL.

## 8. USAGE GUIDELINES

## 9. BACKGROUND AND OBJECTIVES

As noted in the introduction, this Core Vocabulary is one of set produced under Action 1.1 of the ISA Programme [A11]. Further background is available in "Towards Open Government Metadata" [TOGM] which offers an overview and context for the work.

The natural course of action for any practitioner or team given the task of recording information about a natural person is to write a list of the data elements they need (or already have) and to work within the specific context of their project. This often works in that it demonstrably meets the project's needs. The problems only arise when one team wants to exchange data with another. It's at that point that the choice of, say, 'first name' over 'given name' and 'surname' over 'family name' becomes an obstacle. Such terms are well defined in a variety of vocabularies and their use cannot be regarded as 'wrong', however, it's easy for simple choices to lead to unintended difficulties further down the road.

The aim of providing Core Vocabularies via the Joinup Platform is not to force teams to use a particular set of terms, or to require the re-engineering of data sets to use them (which can be prohibitively expensive). Rather the aim is to make it easy to see and use the terms that crop up across multiple domains; terms that, when used by public sector agencies, will make data more interoperable.

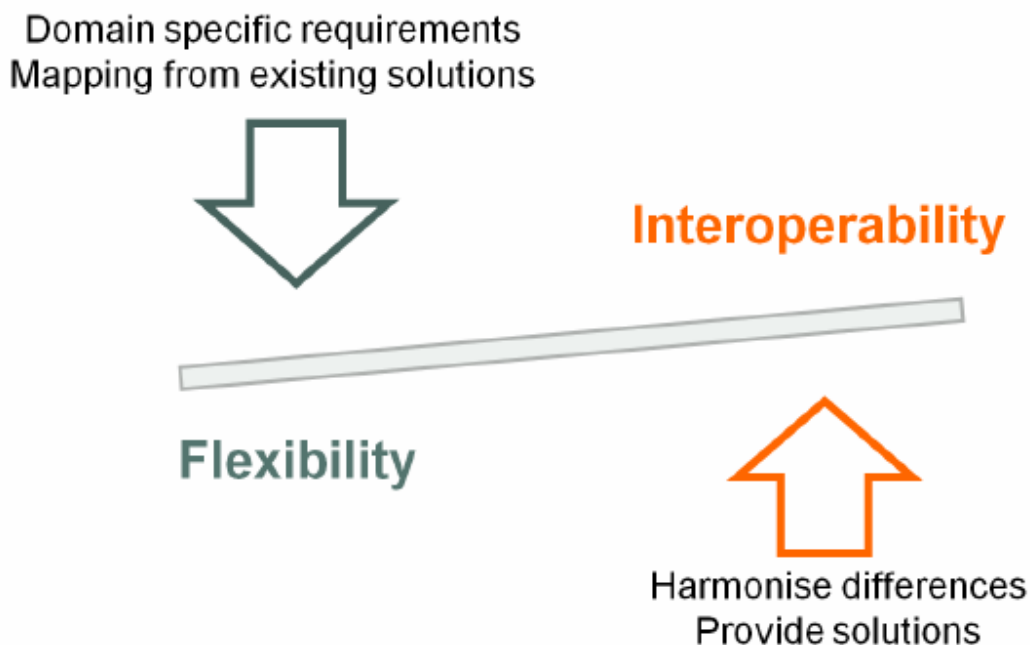


Figure 2 The struggle between enabling interoperability and giving flexibility

Identifiers are a case in point. In an international context, a person's passport number is likely to be critically important. This is not so within a university where the likelihood is that an in-house

identifier will be assigned. Rather than one agency defining a term for 'passport number' and another for 'student number', both can use the core vocabulary term 'identifier', preferably with some additional contextual information. As Figure 2 illustrates, there is a balance to be struck between flexibility and interoperability.

## 10. APPROACH & COMMUNITY

The process and methodology followed in the development is set out in detail in the Process and Methodology for Developing Core Vocabularies [PMDCV].

Specific acknowledgement is due to:

## 11. CHANGE CONTROL

ADMS.F/OSS is published by DG DIGIT. Review comments and requests for changes can be made via the mailing list which is archived at [http://joinup.ec.europa.eu/mailman/private/adms\\_foss-wg/](http://joinup.ec.europa.eu/mailman/private/adms_foss-wg/).

## 12. REFERENCES

- [A11] Action 1.1 Improving semantic interoperability in European eGovernment systems [http://ec.europa.eu/isa/actions/01-trusted-information-exchange/1-1action\\_en.htm](http://ec.europa.eu/isa/actions/01-trusted-information-exchange/1-1action_en.htm)
- [ADMS] Asset Description Metadata Schema, Makx Dekkers (Editor), PwC EU Services, <http://joinup.ec.europa.eu/asset/adms/home>
- [CTT] Centro de Transferencia de Tecnología, <http://administracionelectronica.gob.es/ctt>
- [DBpedia] DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. <http://dbpedia.org/>
- [DC] DCMI Metadata Terms, Dublin Core Metadata Initiative. <http://dublincore.org/documents/dcmi-terms/>
- [FOAF] Friend of a Friend

---

<http://xmlns.com/foaf/spec/>

- [ISA] Interoperability Solutions for European Public Administrations,  
<http://ec.europa.eu/isa/>
- [ISO 3166-1] ISO 3166 code lists, ISO. [http://www.iso.org/iso/iso\\_3166\\_code\\_lists](http://www.iso.org/iso/iso_3166_code_lists)
- [ISO 8601] Data elements and interchange formats -- Information interchange --  
Representation of dates and times, ISO 8601:2004.  
[http://www.iso.org/iso/catalogue\\_detail?csnumber=40874](http://www.iso.org/iso/catalogue_detail?csnumber=40874)
- [JOINUP] The Joinup Platform is operated by the European Commission designed to  
enable the sharing and reuse open-source software, semantic assets and other  
interoperability solutions for public administrations. See  
<http://joinup.ec.europa.eu/>
- [ORG] An Organization Ontology, Dave Reynolds, October 2010. Soon to be  
republished by W3C at <http://www.w3.org/TR/gld-org/>
- [PMDCV] Process and Methodology for Developing Core Vocabularies, 22 November  
2011. <https://joinup.ec.europa.eu/elibrary/document/isa-deliverable-process-and-methodology-developing-core-vocabularies>
- [RFC 3066] H. Alvestrand, ed. RFC 3066: Tags for the Identification of Languages 1995.  
Available at: <http://www.ietf.org/rfc/rfc3066.txt>
- [TOGM] Towards Open Government Metadata, Vassilios Peristeras, DG DIGIT, ISA  
Unit, September 2011  
[https://joinup.ec.europa.eu/sites/default/files/towards\\_open\\_government\\_metadata\\_0.pdf](https://joinup.ec.europa.eu/sites/default/files/towards_open_government_metadata_0.pdf)
- [XSD] XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28  
October 2004. <http://www.w3.org/TR/xmlschema-2/#date>