



ASSET DESCRIPTION METADATA SCHEMA FOR SOFTWARE 1.00

ADMS.SW 1.00



JOINING UP GOVERNMENTS



DOCUMENT METADATA

Property	Value
Release date	29/06/2012
Status:	Final
Version:	1.00

Statement of copyright:

This specification is released under the "ISA Open Metadata Licence v1.1" that can be retrieved from <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-licence-v11>.

This specification was prepared for the ISA programme by:

PwC EU Services

Disclaimer:

The views expressed in this draft specification are purely those of the authors and may not, in any circumstances, be interpreted as stating an official position of the European Commission.

The European Commission does not guarantee the accuracy of the information included in this study, nor does it accept any responsibility for any use thereof.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.

All care has been taken by the author to ensure that s/he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from her/his or their legal representative.

This draft specification can be downloaded from the ISA website:

https://joinup.ec.europa.eu/asset/adms_foss/release/release100.

TABLE OF CONTENTS

1. Introduction.....	5
2. Conformance Statement	6
2.1 Data Publishing Conformance.....	6
2.2 Data Consuming Conformance	6
3. Use Cases.....	7
3.1 Explore and search for software.....	7
3.2 Facilitate the setup of Institutional F/OSS contributions portals.....	11
3.3 Extract description metadata from software tags	11
4. Conceptual Model	13
4.1 Data Types	16
4.2 Main Concepts.....	17
4.2.1 Concept: Software Project.....	17
4.2.2 Concept: Software Release.....	19
4.2.3 Concept: Software Package.....	22
4.2.4 Concept: Software Repository.....	24
4.3 Supporting Concepts	25
4.3.1 Concept: Agent.....	25
4.3.2 Concept: Contact Information.....	25
4.3.3 Concept: Data Set	26
4.3.4 Concept: Documentation	28
4.3.5 Concept: Licence.....	28
4.4 Software Classification Concepts	28
4.4.1 Concept: Intended Audience	29
4.4.2 Concept: Locale.....	29
4.4.3 Concept: Operating System	29
4.4.4 Concept: Programming Language	30
4.4.5 Concept: Status	30
4.4.6 Concept: Topic	30
4.4.7 Concept: User Interface Type	31
4.5 RADion Classification Concepts.....	31
4.5.1 Concept: File Format.....	31
4.5.2 Concept: Geographic Coverage.....	31
4.5.3 Concept: Language	32
4.5.4 Concept: Theme	32
4.5.5 Concept: Theme Taxonomy	32
5. Controlled Vocabularies	33



5.1	File format Vocabulary.....	33
5.2	Geographic coverage Vocabulary	34
5.3	Intended Audience Vocabulary.....	34
5.4	Language Vocabulary.....	36
5.5	Locale Vocabulary	36
5.6	Licence Vocabulary	37
5.7	Licence Type vocabulary.....	37
5.8	Operating System.....	40
5.9	Programming language	41
5.10	Status Vocabulary	41
5.11	Theme Vocabulary	42
5.12	Topic Vocabulary.....	42
5.13	User Interface Type	43
6.	The RDF Schema	44
7.	Acknowledgement.....	46
7.1	ADMS.SW Working Group	46
7.2	The SEMIC Team.....	48
8.	References.....	49

1. INTRODUCTION

ADMS.SW is designed to help make it easier to discover all types of software, but particularly free and open source software (F/OSS), catalogued or located in many *different* software catalogues, repositories, and forges, particularly for use in public administrations. What software is available to handle a particular function? Who maintains the software and who uses it already?

ADMS.SW has been created under the Interoperability Solutions for European Public Administrations [ISA] Programme of the European Commission. This programme encourages public administrations in Europe to share and reuse interoperability solutions as free and open source software (F/OSS) and give better visibility to this software through by better interlinking on-line software repositories. This can be achieved by creating an [enhanced software description metadata specification](#) that will allow software forges to exchange more information about their software projects and software releases.

The process and methodology followed in the development of ADMS.SW is set out in detail in the document "[Process and Methodology for Developing Core Vocabularies](#)" [PMDCV]. Further background is available in the studies "[Vision for an enhanced software description metadata schema and software catalogue for e-Government](#)" [Vision] and "[Report on existing Software Forges](#)" [Forges] which offers an overview and context for the work.

The intention is not to create a new metadata vocabulary, but to identify and re-use existing methods for describing software wherever possible. In particular, ADMS.SW draws on the following work:

- Description of a Project [DOAP];
- The Trove System [TROVE];
- The Data Catalog Vocabulary [DCAT];
- The Software Package Data Exchange [SPDX] specification;
- The ISO/IEC 19770 series on Software Asset Management including in particular ISO/IEC 19770-2 Software Identification Tag, and ISO/IEC 19770-3 Software Entitlement Tag (still in development);
- The Asset Description Metadata Schema [ADMS]; and
- The Repository, Asset, Distribution [RADion] Schema.

2. CONFORMANCE STATEMENT

Conformance with this specification can be achieved in a number of ways as detailed below.

2.1 DATA PUBLISHING CONFORMANCE

Conceptual conformance, meaning that data is published along with a human readable mapping from the terms used by the publisher to the terms defined by ADMS.SW.

Conceptual conformance+ meaning that data is published along with a machine processable mapping, such as an XSLT, from the terms used by the publisher to the terms defined by ADMS.SW.

RDF publishing conformance means that data is published using the RDF classes and properties specified in ADMS.SW.

2.2 DATA CONSUMING CONFORMANCE

RDF consuming conformance, meaning that a consumer of data published using the ADMS.SW vocabulary must understand all the terms defined in the RDF Schema; RDF data published using any term in the vocabularies will be consumed and processed accordingly.

3. USE CASES

This section describes the use cases for an enhanced software catalogue for e-Government system development.

3.1 EXPLORE AND SEARCH FOR SOFTWARE

This use case describes the interconnection of software catalogues through the exchange of software description metadata. This enables a search for software from a single point of access.

Business need: System developers and end-users need to be able to easily *explore* [FRSAD], *find*, *identify*, *select*, and *obtain* [FRBR] software developed by *different* organisations and originally catalogued or located in many *different* software catalogues, repositories, and forges.

- To **explore** software that is available in a particular subject area and to explore the relationships between software in order to understand the structure of a subject area and its terminology;
- To **find** software that correspond to the user's stated search criteria (i.e., to locate either a single software packages or a set of software packages in *multiple* repositories or catalogues as the result of a search using a known characteristics of the software);
- To **identify** software (i.e., to confirm that the software described corresponds to the software sought, or to compare two or more software packages with similar characteristics in *multiple* repositories or catalogues);
- To **select** software that is appropriate to the user's needs (i.e., to choose a software package that meets the user's requirements with respect to content, format, etc., or to reject a software solution as being inappropriate to the user's needs);
- To **obtain** access to the software package described (i.e., to access an entity electronically through an online connection).

Usage scenario: Working on a new e-Government project, a user might have information needs related to exploration and finding software solutions, for example a user is interested in the existence of software libraries that allow him to manipulate spatial datasets that comply to the INSPIRE specifications.

- **Without a common software metadata vocabulary:** a user might try a keyword-based search on a general-purpose internet search engine or multiple searches on existing software catalogues and software forges.
- **With the enhanced federation:** a user performs a single keyword-based and facet-based search on the enhanced catalogue. The catalogue provides detailed search results. To obtain the software, the user is directed to the URL on the software repository or forge (or another location) where the software can be retrieved.

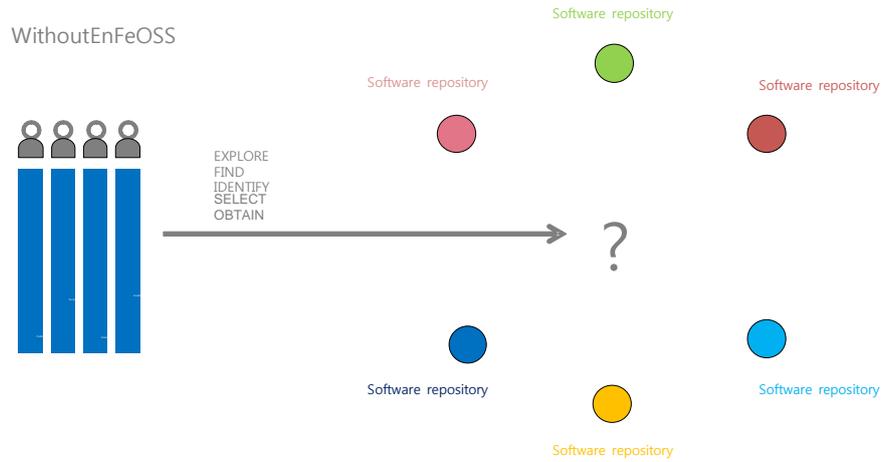


Figure 1 Without a common metadata vocabulary

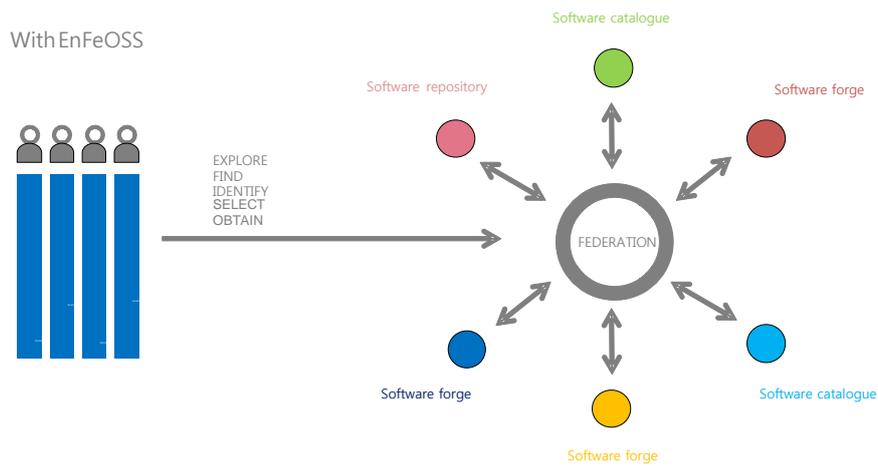


Figure 2 With a common metadata vocabulary

Similar to the Functional Requirements for Bibliographic Records [FRBR] the table below contains a list of conceivable asset metadata properties and relationships. Plotted against each property and relationship are the five generic user tasks (i.e., explore, find, identify, select, and obtain). The symbols used in the tables (■ □ ○) indicate the relative value of each attribute or relationship in supporting a specific user task focused on a particular entity. The symbol ■ signifies that an attribute or relationship is highly important for supporting the designated task; the symbol □ signifies moderate importance; and the symbol ○ signifies relatively low importance. The absence of a symbol indicates that the attribute or relationship has no discernible relevance to that particular user task or sub-task. The properties and relationships greyed out have not been included in the ADMS.SW conceptual model.

Table 1 Required information to support the user tasks to explore, find, identify, select, and obtain F/OSS (The properties and relationships greyed out have not been included in the ADMS.SW conceptual model)

Metadata category	Metadata property or relationship	Description	Explore	Find	Identify	Select	Obtain
descriptive metadata	title	the title of the software in multiple languages		■	■		■
	description	descriptive text in multiple languages		■	■		
	local identifier	local identifier for the software		■	■		■
	URI	uniform resource identifier		■	■		■
	version	version of the software release		■	■		■
	<i>related software</i>	related software		□			
	next version	a newer version of the software		□	□		□
	release	a release of the software					□
applicability	theme	the theme of the software	■	□		□	
	spatial coverage	geographic region in which the software can be used	■	□		□	
	multilingual	whether or not the software can be configured to have a multilingual user interface	□			□	
	language	natural language in which the software interface is available				□	
	related regulation	related regulations from which the software is derived	■				
provenance	origin	repository or catalogue that contains the primary description of the software	■	■			
	publisher	organisation responsible for the publication of the software	□	■	■		■
	publisher type	the kind of publisher	■				
	created	date of creation			■		
	modified	date of latest update			■		
people	developer	person who developed the software	□				
	documenter	person who documented the software					

Metadata category	Metadata property or relationship	Description	Explore	Find	Identify	Select	Obtain
	maintainer	person who maintains the software	<input type="checkbox"/>				
	helper	person who helps with the software					
	tester	Person who tests the software				<input type="checkbox"/>	
	translator	person who translates the software					
format	programming language	programming language of the software	■	<input type="checkbox"/>	■	■	
	theme	the topic, theme or function of the software	■	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
availability	licence	A legal document giving official permission to do something with a resource				■	
	licence class	the class of licences that govern (re-)use of releases (e.g. BSD)	<input type="checkbox"/>		<input type="checkbox"/>		
	license type	coarse type of rights and obligations that come with the license	<input type="checkbox"/>				
	status	status in the context of a particular workflow process			<input type="checkbox"/>	■	
	operating system	the platform for which a binary package exists		<input type="checkbox"/>	<input type="checkbox"/>		
	access URL	URL of the software (release)					■
	(related) documentation	documentation of the software					○
	(related) homepage	an associated web page					<input type="checkbox"/>
interoperability credentials	related asset	a specification implemented by the software	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
usage credentials	used by	the organisations that use the software	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
	used in public service type	the electronic public service type in which the software is used			<input type="checkbox"/>	<input type="checkbox"/>	
metrics	#commits	the number of code commits, as an indicator of the project's activity				<input type="checkbox"/>	
	#downloads	the number of downloads of the asset (release)				<input type="checkbox"/>	
	#installations	the number of installations of the software				<input type="checkbox"/>	
	#users	the number of users of the software				<input type="checkbox"/>	

Metadata category	Metadata property or relationship	Description	Explore	Find	Identify	Select	Obtain
	#end users	the number of end users of the software				<input type="checkbox"/>	
	#lines of code	the number of lines of code				<input type="checkbox"/>	
	#effort	the effort spent on producing a new software release, not counting the effort spent on previous software releases.				<input type="checkbox"/>	
	#cumulative effort	the total effort spent (expressed in personhours) on producing a software release, including the effort spent on previous releases.				<input type="checkbox"/>	

3.2 FACILITATE THE SETUP OF INSTITUTIONAL F/OSS CONTRIBUTIONS PORTALS

A (public) forge may host different projects developing software, where different actors collaborate, from different institutions. In the same project, there can be different institutional partners that cooperate for a common development. Sometimes such projects will be hosted at a particular partner 's forge, sometimes on another's, or sometimes on a public forges, which has no particular link with each partner (settling in advance any dispute on the predominant role of any of these partners).

Still, each of the participating institutions may wish to exhibit a “portal” of its contributions in F.OSS projects, alongside other information, sometimes as a “portfolio” of F.OSS contributions. It is thus important that metadata relating to sponsors / funders / proprietors of the developed software can be managed in the case of cooperation, allowing the same project or software to appear in several institutional portals. Alternatively, the hosting forges may have the capacity to properly credit institutions participating to projects (or funding them, etc.).

3.3 EXTRACT DESCRIPTION METADATA FROM SOFTWARE TAGS

This use case describes the process of extracting description metadata from a software tag.

Business need: software tagging can be particularly useful to reduce the burden of maintaining software description metadata. The creation and maintenance of software description metadata should follow a “create-once use-everywhere” principle; software publishers or the owners of software catalogues should not be manually re-entering the same metadata descriptions, once formatted in a machine-readable way. This can be most easily achieved when the metadata and the software package are kept together. This is the benefit of software tagging: an XML or RDF-XML file containing the description metadata is associated with the software package; the file is either embedded in or stored at the same location as the software package. The following specifications apply software tagging:

- The **ISO/IEC 19770-2:2009 standard** is an XML data structure that allows software publishers to ensure that the software packages they produce can be *uniquely identified and uniformly described* through certified or uncertified software identification tags [19770-2]. The software identification tags can be created independently by any software publisher and digitally signed using certificates provided by any certification authority. The software identification tag is designed to link to a separate software entitlement tag (conceptually a software license tag), which will be specified by the ISO/IEC 19770-3 standard currently at the Committee Draft stage.
- The idea of software tagging is also envisioned by the **Software Package Data Exchange (SPDX)** community. The SPDX is a specification to tag open-source software packages to, facilitate the compliance checking process with free and open source software licenses.

Usage scenario (software repository harvests metadata from software tag): the software description metadata in a software repository follows the lifecycle of the software package and the description metadata is kept synchronised between the tag and the software repository:

1. **Metadata creation:** when a software package is created, its description metadata is stored in a software tag. When the software package is added to the software repository, the software repository retrieves description metadata from the tag.
2. **Metadata update:** the software repository periodically harvests the description metadata from the software tag. If the metadata has been updated, it will update the metadata in the repository.
3. **Metadata deletion:** when the software package is removed, the software catalogue updates its description metadata accordingly.

Usage scenario (software forge generates software identification tag): when a software forge is used for releasing software packages, the software forge could generate a software identification tag for the software release. In this case, the authentic source of metadata is maintained by the software forge.

4. CONCEPTUAL MODEL

The conceptual model presented in this section defines the concepts, properties, and relationships that are needed to describe software and meet the requirements of the above mentioned used cases.

Hereby it is important to remain structural correspondence with relevant related work. Such structural correspondence facilitates comprehensibility, tool implementation and data integration. To ensure that the [Data Catalog Vocabulary](#) (DCAT), the [Asset Description Metadata Schema](#) (ADMS), and the [Asset Description Metadata Schema for Software](#) (ADMS.SW) are seeded on the same structure, the [RADion](#) vocabulary was created [RADion]. RADion is shorthand for **Repository, Asset, and Distribution** – the three structural elements that RADion abstracts from. Figure 3 is a UML representation of RADion v1.00.

In ADMS.SW, the concepts Software Repository, Software Release and Software Package are defined as specialisations of the more general concepts Repository, Asset and Distribution specified by RADion and presented in Figure 4. The concepts, properties, and relationships in Figure 4 are further clarified in this section. It is divided in five parts:

1. **Data types:** the primitive and compound data types used in the conceptual model;
2. **Main concepts:** a definition of the concepts Software Project, Software Release, Software Distribution, and Software Repository, together with their properties and relationships;
3. **Supporting concepts:** concepts such as Agent, Contact Information, Documentation, and Licence, which are related to software;
4. **Software classification concepts:** a conceptual model of the Trove classification system for software [TROVE];
5. **RADion classification concepts:** generic classifying concepts for software inherited from the RADion conceptual model.

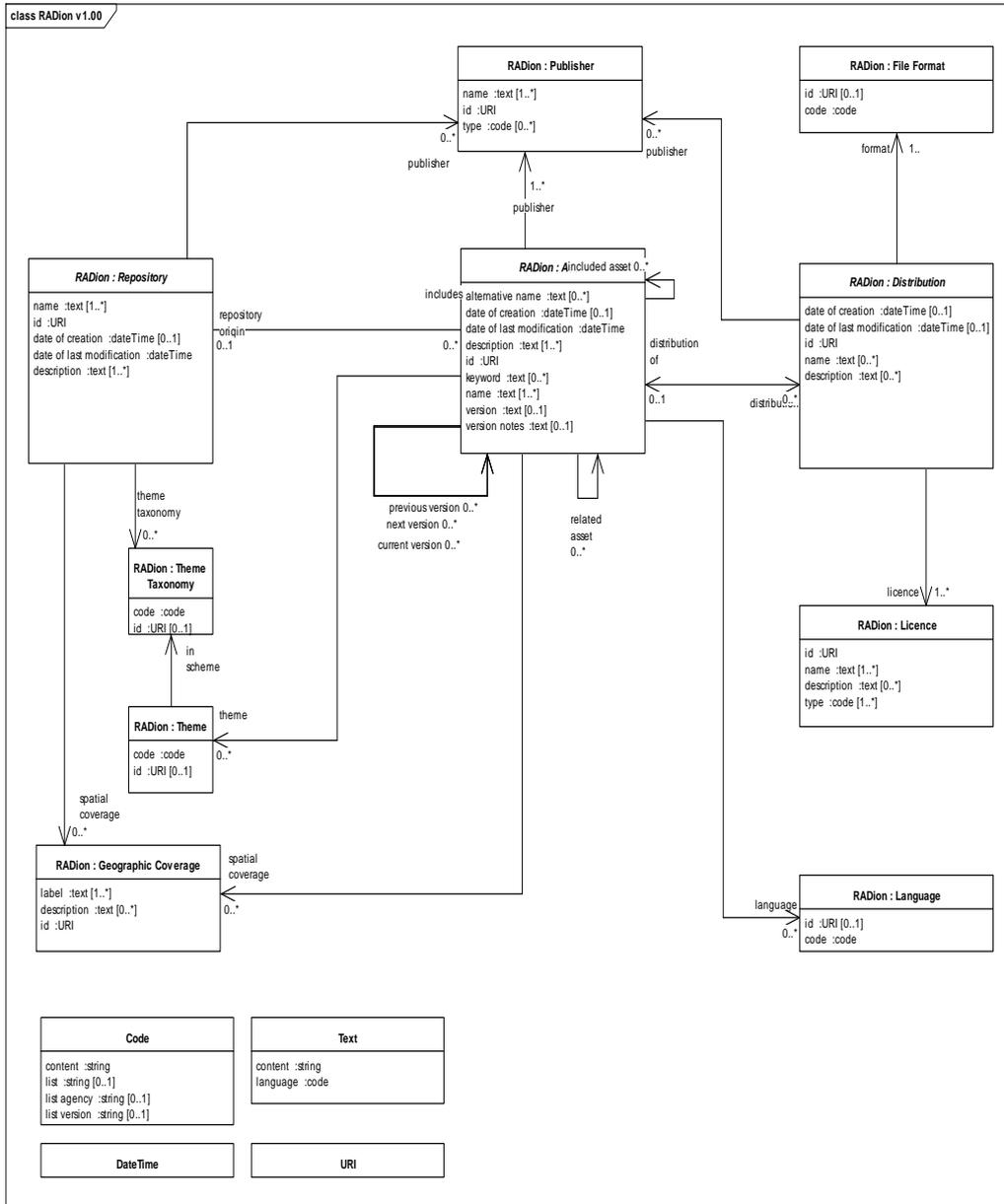


Figure 3 UML Class Diagram of the Repository - Asset - Distribution abstract model (RADion) v1.00

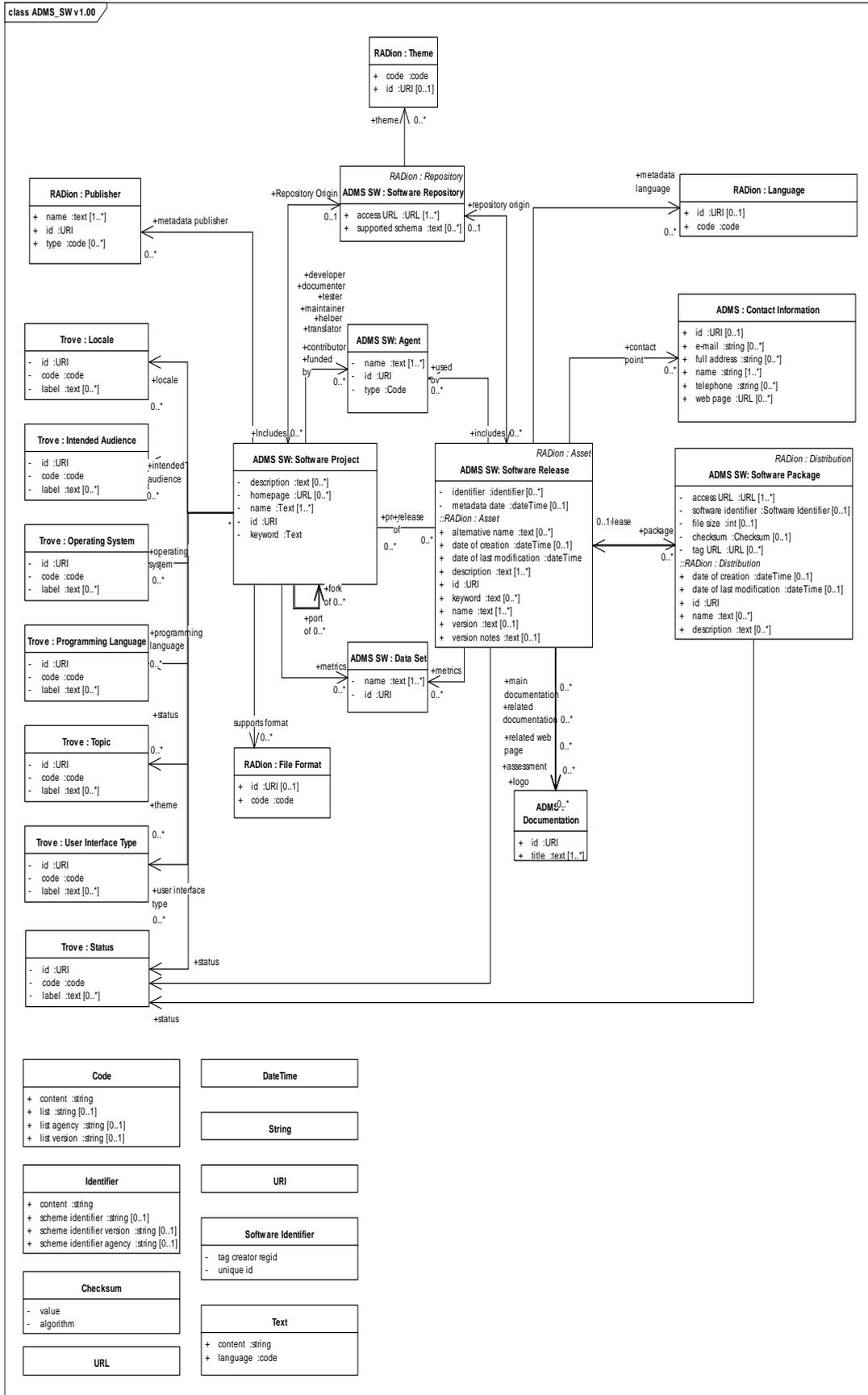


Figure 4 UML Class diagram of the ADMS.SW extension to RADion

4.1 DATA TYPES

The table below defines the primitive and compound data types that are used in the ADMS.SW conceptual model.

<i>Data type</i>	<i>Description</i>
Code	<p>Complex type (based on UN/CEFACT Code. Type¹) consisting of:</p> <ul style="list-style-type: none"> • a content string which is the code (data type String) • an optional identifier for the code list (data type String) • an optional identifier for the agency that manages the code list (data type String) • an optional identifier for the version of the code list (data type String)
Checksum	<p>A value that allows validating the integrity of a file's content. According to the SPDX specification [SPDX] it is a complex type consisting of:</p> <ul style="list-style-type: none"> • the algorithm • a checksum value.
DateTime	String; syntax conforming to ISO8601 ^{2,3}
Identifier	<p>Complex type (based on UN/CEFACT Identifier. Type⁴) consisting of:</p> <ul style="list-style-type: none"> • a content string which is the identifier (data type String) • an optional identifier for the identifier scheme (data type String) • an optional identifier for the version of the identifier scheme (data type String) • an optional identifier for the agency that manages the identifier scheme (data type String)
Software Identifier	<p>Complex type (based on the software identifying system in the ISO19770-2:2009 software identification tag XML data structure [ISO19770-2]) consisting of:</p> <ul style="list-style-type: none"> • a unique token within the regid domain of the tag creator and/or subdomain; and • a tag creator regid. <p>Example in XML representation:</p> <pre><swid:software_id> <swid:unique_id>AcrobatPro-AS1-Win-GM-MUL</swid:unique_id> <swid:tag_creator_regid>regid.1986-12.com.adobe</swid:tag_creator_regid> </swid:software_id></pre>

¹ UNECE United Nations Economic Commission for Europe. UN Centre for Trade Facilitation and Electronic Business (UN/CEFACT). Core Components Data Type Catalogue Version 3.1. <http://www.unece.org/fileadmin/DAM/cefact/codesfortrade/CCTS/CCTS-DTCatalogueVersion3p1.pdf>

² ISO 8601:2004. Data elements and interchange formats -- Information interchange -- Representation of dates and times. http://www.iso.org/iso/catalogue_detail?csnumber=40874

³ Id. http://dotat.at/tmp/ISO_8601-2004_E.pdf

⁴ UNECE United Nations Economic Commission for Europe. UN Centre for Trade Facilitation and Electronic Business (UN/CEFACT). Core Components Data Type Catalogue Version 3.1. <http://www.unece.org/fileadmin/DAM/cefact/codesfortrade/CCTS/CCTS-DTCatalogueVersion3p1.pdf>

<i>Data type</i>	<i>Description</i>
String	String of UNICODE ⁵ characters
Text	Complex type consisting of: <ul style="list-style-type: none"> • a content string (data type String) • an optional language code from RFC5646⁶ (data type code)
URI	String; syntax conforming to RFC 3986 ⁷ or RFC3987 ⁸ Note: An IRI (Internationalized Resource Identifier) is a Unicode string that conforms to the syntax defined in RFC 3987 [IRI]. IRIs are a generalization of URIs [URI]. Every absolute URI and URL is an IRI.
URL	String; syntax conforming to RFC 1738 ⁹

4.2 MAIN CONCEPTS

In this section the main concepts in the ADMS.SW conceptual model are defined together with their properties and relationships.

4.2.1 Concept: Software Project

A Software Project is a time-delimited undertaking with the objective to produce one or more software releases, materialised as software packages. Some projects are long-running undertakings, and do not have a clear time-delimited nature or project organisation. In this case, the term 'software project' can be interpreted as the result of the work: a collection of related software releases that serve a common purpose.

An example of a Software Project is the Apache HTTP Server Project¹⁰.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
description	descriptive text for the Project Note: a project can have more than one description. In most cases, this will be attributed to the fact that the project description is available in multiple language.	Text	0..*

⁵ UNICODE. <http://www.unicode.org/standard/standard.html>

⁶ Internet Engineering Task Force (IETF). RFC 5646. Tags for Identifying Languages. <http://www.ietf.org/rfc/rfc5646.txt>

⁷ Internet Engineering Task Force (IETF). RFC 3986. Uniform Resource Identifier (URI): Generic Syntax. <http://www.ietf.org/rfc/rfc3986.txt>

⁸ Internet Engineering Task Force (IETF). RFC5890. Internationalized Domain Names for Applications (IDNA). <http://tools.ietf.org/html/rfc5890>

⁹ Internet Engineering Task Force (IETF). RFC 1738. Uniform Resource Locators (URL). <http://www.ietf.org/rfc/rfc1738.txt>

¹⁰ Apache HTTP Server Project <http://httpd.apache.org/>

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
homepage	homepage for the Project	URI	0..*
id	URI for the Project Example: the URI of the Apache HTTP Server Project could be http://httpd.apache.org/ .	URI	1..1
keyword	word or phrase to describe the Release	Text	0..*
name	name of the Project Note: in cases that a software project has parallel names, for example if more than one official name exists, or if the name exists in more than one official language, this field can be repeated for all name variants	Text	1..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
release	software release that the project has produced.	Software Release	0..*
contributor	contributing person or organisation for the project	Agent	0..*
funded by	source of funding (person or organisation) for the project	Agent	0..*
fork of	software project of which the current project is a descendant	Software Project	0..*
developer	person who developed the software	Agent	0..*
documenter	person who documented the software	Agent	0..*
maintainer	person who maintains the software	Agent	0..*
helper	person who helps with the software	Agent	0..*
tester	Person who tests the software	Agent	0..*
translator	person who translates the software	Agent	0..*
metrics	a data set of metrics about the software	Data Set	0..*
theme	topic, theme or function of the software	Theme or Topic	0..*
intended audience	intended audience of the software	Intended Audience	0..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
locale	a locale of the software	Locale	0..*
user interface type	user interface type of the software	User Interface Type	0..*
programming language	programming language of the software	Programming Language	0..*
repository origin	Software forge or repository that contains the primary description of the Software Release	Software Repository	0..1
operating system	operating system of the software	Operating System	0..*
status	status of the software	Status	0..*
supports format	data format that is supported by the software	File Format	0..*

4.2.2 Concept: Software Release

A Software Release is an abstract entity that reflects the intellectual content of the software at a particular point in time and represents those characteristics of the software that are independent of its physical embodiment. This abstract entity corresponds to the FRBR entity *expression* (the intellectual or artistic realization of a work)¹¹. A release is typically associated with a version number.

An example of a Software Release is the Apache HTTP Server 2.22.22 (httpd)¹² release.

The Software Release class is a subclass of the RADion Asset class and therefore inherits all the latter's properties and relationships. These are defined in the RADion specification and are summarised in the following section with notes on their use within the ADMS.SW context.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
-----------------	--------------------	------------------	--------------------

¹¹ IFLA. Cataloguing Section. Functional Requirements for Bibliographic Records, section 3. Entities. http://archive.ifla.org/VII/s13/frbr/frbr_current3.htm

¹² Apache HTTP Server 2.22.22 (httpd) <http://httpd.apache.org/download.cgi#apache22>

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
alternative name	<p>alternative name for the Release. Any number of alternative names may be supplied.</p> <p>Note: this information may be used to provide additional access points, e.g. allowing indexing of any acronyms, nicknames, shorthand notations or other identifying information under which a user might expect to find the Release</p>	Text	0..*
date of creation	creation date of this version of the Release	DateTime	0..1
date of last modification	date of latest update of the Release	DateTime	0..1
description	descriptive text for the Release	Text	1..*
id	URI for the Release	URI	1..1
identifier	<p>any identifier for the Release.</p> <p>Any number of identifiers, whether URIs or not, may also be assigned to the Release.</p>	identifier	0..*
keyword	word or phrase to describe the Release	Text	0..*
metadata date	date of the most recent update of the metadata for the Release	DateTime	0..1
name	<p>name of the Software Release. The software must have a name.</p> <p>Note: in cases that a Software Release has parallel names, for example if more than one official name exists, or if an organisation or country has more than one official language, this field can be repeated for all name variants</p>	Text	1..*
version	version number or other designation of the Release	Text	0..*
version notes	additional notes about the specific version of the software, for example description of changes between this version and the previous version of the Software Release.	Text	0..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
report	report on or assessment of the asset	Documentation	0..*
contact point	contact point for further information about a Release.	Contact Information	0..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
included asset	<p>an Asset that is contained in the Software Release being described.</p> <p>Example: a Linux distribution is being described, that contains other libraries and submodules.</p>	Asset	0..*
metrics	a data set of metrics about the software	Data Set	0..*
language	<p>language of the Software Release. Software Releases and the projects that create them are typically described throughout in a given language. Note that this is quite separate from Locale.</p>	Language	0..*
logo	the logo of the project	Documentation	0..1
main documentation	the main documentation or specification of the Asset	Documentation	0..*
metadata language	<p>language of the metadata for the Asset.</p> <p>Note: Properties like name and description are language-tagged strings. The metadata language tag indicates the different languages in which the asset description metadata can be expected to be available.</p>	Language	0..*
metadata publisher	organisation making the metadata for the Asset available	Publisher	0..*
current version	current or latest version of the Release	Software Release	0..1
next version	newer version of the Asset	Software Release	0..*
previous version	older version of the Asset	Software Release	0..*
project	project that has produced the Asset	Project	0..*
publisher	organisation making the Asset available. It is perhaps worth noting that the publisher might be the project or a participant in the project.	Publisher	1..*
related asset	another piece of software or another interoperability asset (e.g. a domain model or code list) that is related to the current Asset	Asset	0..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
related documentation	documentation that contains information related to the asset. This may include screenshots and videos as well as textual documentation.	Documentation	0..*
related web page	a Web page that contains information related to the asset	Documentation	0..*
package	implementation of the Software Release that can be downloaded.	Software Package	0..*
repository origin	Software forge or repository that contains the primary description of the Software Release	Software Repository	0..1
spatial coverage	specific geographic relevance of the software - if any. This inherited property is likely to be left unused in ADMS.SW	Geographic Coverage	0..*
status	status of the Asset in the context of a particular workflow process	Status	0..*
theme	theme or sector to which the Asset applies. This is expected to be provided as part of a SKOS concept scheme	Theme	0..*
used by	agent that uses the Asset	Agent	0..*

4.2.3 Concept: Software Package

A Software Package represents a particular physical embodiment of a Software Release, which is an example of the FRBR entity manifestation (the physical embodiment of an expression of a work). A Software Package is typically a downloadable computer file (but in principle it could also be a paper document) that implements the intellectual content of a Software Release. A particular Software Package is associated with one and only one Software Release, while all Packages of an Asset share the same intellectual content in different physical formats.

An example of a Software Package is `httpd-2.2.22.tar.gz`¹³, which represents the Unix Source of the Apache HTTP Server 2.22.22 (httpd) software release.

Software often has at least two kinds of physical embodiments: a source code package and a binary package. Binary packages are sometimes compiled for different operating systems or are released under different licences, e.g. in case of dual licensing. Also scripting languages need some sort of packaging for installation systems used by end users.

¹³<http://httpd-2.2.22.tar.gz>, <http://apache.mogo.be/httpd/httpd-2.2.22.tar.gz>

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
access URL	URL of the Software Package Note: more than one URL may be available, for example if mirror sites are maintained.	URL	1..*
date of creation	creation date of the Package	DateTime	0..1
date of last modification	date of latest update of the Package	DateTime	0..1
description	descriptive text for the Package	Text	0..*
id	URI for the Package	URI	1..1
name	name of the Package	Text	0..*
software identifier	identifier of the software package according to the ISO19770-2:2009 software identifying system	Software Id	0..1
tag URL	URL from which a software tag file can be obtained for the Software Package. Note: several software tag standards exist such as the ISO19770-2:2009, the ISO19770-3 and the SPDX specification. These standards do not require the software tags to be available on via the Web.	URL	0..*
file size	The size of the package file in bytes.	Integer	0..1
checksum	The checksum property provides a mechanism that can be used to verify that the contents of a Package have not been changed.	Checksum	0..1

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
format	format in which the Package is available (e.g. EXE, ZIP) according to a controlled vocabulary	File Format	1..1
licence	conditions or restrictions for (re-)use of the Package.	Licence	1..*
release	the Software Release that this Software Package embodies. This is the reverse relationship of package	Software Release	0..1
publisher	organisation making the Package available	Publisher	0..*
status	status of the Package in the context of a particular workflow process	Status	0..*

4.2.4 Concept: Software Repository

A Software Repository is a system or service that provides facilities for storage and maintenance of descriptions of Software Projects, Software Releases and Software Packages, and functionality that allows users to search and access these descriptions. A Software Repository will typically contain descriptions of several Software Projects, Software Releases and related Software Packages.

An example of a Software Repository is the Apache Software Foundation Project Catalogue¹⁴.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
access URL	URL of the Repository	URL	1..*
date of creation	creation date of the Repository	DateTime	0..1
date of last modification	date of latest update of the Repository	DateTime	0..1
description	descriptive Text for the Repository	Text	1..*
id	URI for the Repository	URI	1..1
name	name of the Repository Note: in cases that a Repository has parallel names, for example if more than one official name exists, or if an organisation or country has more than one official language, this field can be repeated for all name variants	Text	1..*
supported schema	Schema (semantic asset) according to which the Repository can provide data, e.g. ADMS.SW version 1.0	URI	0..*

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
includes	an Asset for which a description is included in the Repository. This is the reverse relationship of repository origin	Asset	0..*
publisher	organisation or individual making the Software Repository available	Agent	0..*
spatial coverage	geographic region or jurisdiction to which the Repository applies	Geographic Coverage	0..*

¹⁴ Apache Software Foundation Project Catalogue, <http://projects.apache.org/>

<i>Relationship</i>	<i>Description</i>	<i>Target</i>	<i>Cardinality</i>
theme taxonomy	The theme taxonomy relationship associates a Repository with a classification scheme for the assets within it	Theme Taxonomy	0..*

4.3 SUPPORTING CONCEPTS

This section defines a number of supporting concepts in the ADMS.SW conceptual model together with its properties and relationships.

4.3.1 Concept: Agent

An Agent is anything that can undertake activities or perform actions; it can represent a natural person, and organisation, or a group.

Several vocabularies have a concept agent, notably Friend of a Friend [FOAF], Dublin Core [DCTERMS] and the Organization Ontology [ORG]. ADMS.SW reused this concept.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
id	URI identifying the Person or Organisation; if no URI is available, the URL of the organisation's or person's home page may be used as an alternative	URI	1..1
name	Name of the agent Note: An Agent may have one of more Names, e.g. if the organisation has names in different languages as may be the case in countries with more than one official language	Text	1..*
type	Value from a list of controlled terms; see Section 5.	code	0..*

4.3.2 Concept: Contact Information

Contact point for further information about an Asset.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
full address	The complete address written as a string, with or without formatting.	String	0..*
e-mail	An e-mail address for questions and feedback	String	0..*

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
name	A name of the contact point, e.g. organisation or department name, function name or name of a person	String	1..*
telephone	A telephone number for questions and feedback	String	0..*
web page	A Web page where contact information is shown, or a Web form where questions and feedback can be entered	URL	0..*

Note: Although both e-mail and web page are optional, one of the two should be provided.

4.3.3 Concept: Data Set

The Data Set concept represents a multidimensional data structure that contains a variety of measures (e.g. number of lines of code) about the software according to a number of dimensions (e.g. time). It is associated with a Software Project and Software Release via the metrics relationship.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
id	URI identifying the Data Set	URI	1..1
name	Name of the Data Set Note: A Data Set may have one of more Names, e.g. if the organisation has names in different languages as may be the case in countries with more than one official language	Text	1..*

The description of a generic multi-dimensional structure is outside the scope of ADMS. For this, the SDMX¹⁵, or RDF Data Cube Vocabulary¹⁶ specifications could be used. However, the multidimensional data set should contain

The table below lists a number of measures that can be expected in a software metrics data set. ADMS.SW does not prescribe how any particular metric is calculated.

<i>Measure</i>	<i>Description</i>	<i>Data Type</i>
number of downloads	The number of times a software package has been downloaded.	Integer

¹⁵ Statistical Data and Metadata Exchange (SDMX), <http://sdmx.org/>

¹⁶ RDF Data Cube Vocabulary, <http://dvcs.w3.org/hg/qld/raw-file/default/data-cube/index.html>

<i>Measure</i>	<i>Description</i>	<i>Data Type</i>
number of installations	The number known instances of software package that have been installed.	Integer
number of users	The number of known users of a software release.	Integer
number of commits	The number of times code for the asset has been committed to the forge.	Integer
number of lines of code	The number of lines of code within the asset (not including any dependencies). Note: observations for the measure 'number of lines of code' should include the dimension 'programming language'.	Integer
time to close support tickets	The time between opening and closing a support ticket in milliseconds.	DateTime
effort	The effort spent (expressed in personhours) on producing a new software release, not counting the effort spent on previous software releases.	Integer
cumulative effort	The total effort spent (expressed in personhours) on producing a software release, including the effort spent on previous releases.	Integer

The table below lists a number of dimensions that can be expected in a software metrics data set.

<i>Dimension</i>	<i>Description</i>	<i>Range</i>
agent dimension	The agent to which a measure applies	Agent
package dimension	The software package to which a measure applies	Software Package
release dimension	The software release to which a measure applies	Software Release
time dimension	The time interval to which a measure applies	Period of Time
programming language dimension	The programming language to which a measure applies. Note: this dimension must be associated with observations for the number of lines of code measure.	Programming Language

4.3.4 Concept: Documentation

Documentation represents an online accessible resource that further describes an Asset or gives guidelines for its use. It can be an online accessible document, a homepage, etc.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
id	URI identifying the Documentation, or the URL where it can be accessed	URI	1..1
title	Title of the Documentation Note: Documentation may have more than one title, e.g. in different languages	Text	1..*

4.3.5 Concept: Licence

A licence is a document with the conditions or restrictions that apply to the use of a Package, e.g. whether it is in the public domain, or that some restrictions apply like in cases attribution is required, or that it can only be used for non-commercial purposes etc.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
description	Descriptive text for the Licence	Text	0..*
id	URI identifying the Licence, or the URL where the text of the licence is available	URI	1..1
name	A short phrase describing the Licence	Text	1..*
type	Value from a list of controlled terms; see section 5.7 for proposed values	Code	1..*

4.4 SOFTWARE CLASSIFICATION CONCEPTS

This section represents the common classifiers of the Trove software map [TROVE]. This is a common classification schema to classify open-source software. Initial credits for implementing the Trove software map go to SourceForge. The Trove software map is also embedded in Fusionforge, based on the GPL branch of GForge which was in turn based on the last GPL version of sourceforge. In February 2012, SourceForge made a wiki¹⁷ page available about the Trove software map under the [CC by](#) licence.

¹⁷ Trove Software Map, <http://sourceforge.net/apps/trac/sourceforge/wiki/Software%20Map%20and%20Trove>

4.4.1 Concept: Intended Audience

Concept used to classify software by its intended audience.

The use of a common classification scheme is strongly recommended, see Section 5 for proposed values.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	Code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.4.2 Concept: Locale

Concept used to classify software with one or more languages in which its UI is available.

The use of a common classification scheme is strongly recommended, see Section 5 for proposed values. It is recommended to use RFC 3066 for this (the familiar en, es-mx etc. codes) and their associated DBpedia URIs for ID.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	Code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.4.3 Concept: Operating System

Operating System is a Concept to classify software according to the operating system(s) for which it is built.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	Code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.4.4 Concept: Programming Language

Concept used to classify software by its programming language.

The use of a common classification scheme is strongly recommended, see Section 5 for proposed values.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.4.5 Concept: Status

Concept used to classify software by its status.

The use of a common classification scheme is strongly recommended, see Section 5 for proposed values.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.4.6 Concept: Topic

Concept used to classify the function of software [TROVECAT]. It should be noted that software may be associated with any number of Topic concepts via the topic relationship.

The use of a common classification scheme is strongly recommended, see Section 5 for proposed values.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the concept	URI	0..1

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
label	language-tagged label in human language	Text	0..*

4.4.7 Concept: User Interface Type

Concept used to classify software by its user interface type.

The use of a common classification scheme is strongly recommended, see Section 6 for proposed values.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the concept	URI	0..1
label	language-tagged label in human language	Text	0..*

4.5 RADION CLASSIFICATION CONCEPTS

This section lists a number of other classifiers mentioned in the RADion and ADMS specification

4.5.1 Concept: File Format

Technical format in which a Package is available, for example “application/zip” for a compressed zip archive or “tar.gz” for a compressed tar archive.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the File Format	URI	0..1

4.5.2 Concept: Geographic Coverage

Specific geographic relevance of the software – if any.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
description	Descriptive text for the Geographic Coverage	Text	0..*

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
id	URI identifying the Geographic Coverage	URI	1..1
label	A short phrase describing the Geographic Coverage	Text	1..*

4.5.3 Concept: Language

Language of a software release if it contains textual information, for example the language in which a specification is written. This is different from the Locale of a software, which is the language in which a user interface is available.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the Language	URI	0..1

4.5.4 Concept: Theme

The concept Theme represents a specific subject to which an Asset can be associated (a skos:Concept).

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the Theme	URI	0..1

4.5.5 Concept: Theme Taxonomy

A Theme Taxonomy is a classification scheme of a repository for the assets within it.

<i>Property</i>	<i>Description</i>	<i>Data Type</i>	<i>Cardinality</i>
code	Value from a list of controlled terms	code	1..1
id	URI identifying the Theme Taxonomy	URI	0..1

5. CONTROLLED VOCABULARIES

This section identifies a number of controlled vocabularies to be used for specific concepts in the ADMS.SW model. In this section, the term “vocabulary” is used as shorthand for various types of controlled vocabularies, including taxonomies (collections of controlled category labels or notations representing concepts in a hierarchical structure), thesauri (networked collections of controlled terms representing concepts in a networked structure) and other types of knowledge organisation systems.

In general, use of controlled collections of terms is recommended as far as possible. Where such collections do not exist, repository publishers should consider creating and maintaining such collections to ensure consistent description. The specification does not mandate any controlled vocabulary for classifying software. It may only suggest or recommend useful controlled vocabularies to enhance interoperability. Consider the following levels of software metadata interoperability:

- **Level 1:** Every software catalogue uses its own classification system, which is not documented in a machine-readable format.
- **Level 2:** Every software catalogue uses its own classification system, which is documented on the Web, in a machine-readable format such as SKOS.
- **Level 3:** Every software catalogue uses its own classification system, which is documented on the Web, in a machine-readable format such as SKOS. There exist automated mappings between these classification systems.
- **Level 4:** Every software catalogue uses the same universal classification system, which is documented on the Web, in a machine-readable format such as SKOS.

Clearly, level 4 is unrealistic; there may always be good reasons for a software forge or catalogue to use its own classification system. It is much better to spend time and energy in attaining level 2 and 3.

An initial proposal was received by Elena Muñoz Salinero, based on a detailed study (Anexo IV: Estudio Taxonomías Forjas, 2011) by the Spanish F/OSS Competence Centre CENATIC [CENATIC]. This study compares all taxonomies implemented by the Spanish forges, OSOR and some relevant forges (SourceForge, Launchpad, GoogleCode, Github). The conclusion of this study is that many of the forges share many taxonomies and it was proposed a minimum set of characteristics to be shared in the interchanging of projects and of course, and optional set of characteristics.

5.1 FILE FORMAT VOCABULARY

The proposed vocabulary for File Format is the list of IANA MIME Media Types¹⁸. The website mediatypes.appspot.com provides Purl.org URIs for the IANA MIME types¹⁹.

¹⁸ IANA Internet Assigned Numbers Authority. MIME Media Types. <http://www.iana.org/assignments/media-types/index.html>

¹⁹ <http://mediatypes.appspot.com/application/zip>

An example is “application/zip” to identified compressed zip files for which the URI <http://purl.org/NET/mediatypes/application/zip> was minted.

5.2 GEOGRAPHIC COVERAGE VOCABULARY

Several vocabularies are available for geographic coverage. ISO 3166 Alpha 2 codes²⁰ can be used for countries, while ISO 3166-2²¹ specifies official subdivisions of countries. NUTS, the Nomenclature of territorial units for statistics maintained by Eurostat can be used. Other possible vocabularies are GeoNames, DBPedia²², FAO Geopolitical Ontology²³, UN/LOCODE²⁴ and The Getty Thesaurus of Geographic Names²⁵.

See for a related activity the ISA work on the Core Location Vocabulary²⁶

5.3 INTENDED AUDIENCE VOCABULARY

The Trove system [TROVE] provide URIs as suitable values for intended audience along two hierarchies: by end-user class and by sector.

<i>Label</i>	<i>Code</i>	<i>Id</i>
by End-User Class	by_enduser	http://sourceforge.net/api/truve/index/rdf#534
Architects	architects	http://sourceforge.net/api/truve/index/rdf#863
Quality Engineers	enduser_qa	http://sourceforge.net/api/truve/index/rdf#537
End Users/Desktop	endusers	http://sourceforge.net/api/truve/index/rdf#2
Developers	developers	http://sourceforge.net/api/truve/index/rdf#3
System Administrators	sysadmins	http://sourceforge.net/api/truve/index/rdf#4
Advanced End Users	enduser_advanced	http://sourceforge.net/api/truve/index/rdf#536
Management	management	http://sourceforge.net/api/truve/index/rdf#725
Testers	testers	http://sourceforge.net/api/truve/index/rdf#865
Security Professionals	secpros	http://sourceforge.net/api/truve/index/rdf#866

²⁰ ISO 3166-1:2006. Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes.

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39719

²¹ ISO 3166-2:2007. Codes for the representation of names of countries and their subdivisions -- Part 2: Country subdivision code.

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39718

²² DBPedia. <http://dbpedia.org/About>

²³ Food and Agriculture Organization of the United Nations. FAO Country Profiles <http://www.fao.org/countryprofiles/geoinfo.asp?lang=en>

²⁴ UNECE United Nations Economic Commission for Europe. UN Centre for Trade Facilitation and Electronic Business (UN/CEFACT). United Nations Code for Trade and Transport Locations (UN/LOCODE). <http://www.unece.org/cefact/locode/welcome.html>

²⁵ The Getty Research Institute. Getty Thesaurus of Geographic Names®.

<http://www.getty.edu/research/tools/vocabularies/tgn/index.html>

²⁶ European Commission. ISA. Joinup. Core Location Vocabulary. https://joinup.ec.europa.eu/asset/core_location/description

<i>Label</i>	<i>Code</i>	<i>Id</i>
by Industry or Sector	by_industrysector	http://sourceforge.net/api/trove/index/rdf#535
Automotive	automotive	http://sourceforge.net/api/trove/index/rdf#768
Agriculture	agriculture	http://sourceforge.net/api/trove/index/rdf#765
Non-Profit Organizations	nonprofit	http://sourceforge.net/api/trove/index/rdf#618
Aerospace	aerospace	http://sourceforge.net/api/trove/index/rdf#599
Government	government	http://sourceforge.net/api/trove/index/rdf#569
Engineering	audienceengineering	http://sourceforge.net/api/trove/index/rdf#729
Information Technology	informationtechnology	http://sourceforge.net/api/trove/index/rdf#363
Financial and Insurance Industry	financialinsurance	http://sourceforge.net/api/trove/index/rdf#361
Healthcare Industry	healthcareindustry	http://sourceforge.net/api/trove/index/rdf#362
Science/Research	scienceresearch	http://sourceforge.net/api/trove/index/rdf#367
Customer Service	customerservice	http://sourceforge.net/api/trove/index/rdf#359
Education	education	http://sourceforge.net/api/trove/index/rdf#360
Manufacturing	manufacturing	http://sourceforge.net/api/trove/index/rdf#365
Telecommunications Industry	telecommunications	http://sourceforge.net/api/trove/index/rdf#368
Legal Industry	legalindustry	http://sourceforge.net/api/trove/index/rdf#364
Religion	religion	http://sourceforge.net/api/trove/index/rdf#366
Security	secindustry	http://sourceforge.net/api/trove/index/rdf#867

The audience relationship associates a software project with the Audience class. For the public sector, ADMS.SW provides the following hierarchical classification developed by Centro de Transferencia de Tecnología [CTT] based on a study by CENATIC [CENATIC]. On the basis of the CENATIC study the following minimal list is proposed.

<i>Label</i>	<i>Code</i>	<i>Id</i>
Citizens	Citizens	http://purl.org/adms/intendedaudience/Citizens
Developer Community	Developers	http://purl.org/adms/intendedaudience/Developers
Individual User	Individuals	http://purl.org/adms/intendedaudience/Individuals
Public administrations	PublicAdministrations	http://purl.org/adms/intendedaudience/PublicAdministrations

<i>Label</i>	<i>Code</i>	<i>Id</i>
National public administrations	NationalPublicAdministrations	http://purl.org/adms/intendedaudience/NationalPublicAdministrations
Regional public administrations	RegionalPublicAdministrations	http://purl.org/adms/intendedaudience/RegionalPublicAdministrations
Local public administrations	LocalPublicAdministrations	http://purl.org/adms/intendedaudience/LocalPublicAdministrations
Enterprise	Enterprise	http://purl.org/adms/intendedaudience/Enterprise
Self Employed Individuals (sei)	SelfEmployedIndividuals	http://purl.org/adms/intendedaudience/SelfEmployedIndividuals
Small and Medium Enterprises (sme)	SME	http://purl.org/adms/intendedaudience/SME
Large Enterprise	LargeEnterprise	http://purl.org/adms/intendedaudience/LargeEnterprise

5.4 LANGUAGE VOCABULARY

The proposed vocabulary for Language is the code list defined by IETF RFC 5646²⁷. URIs for languages are maintained by lexvo.org²⁸ (for ISO639-3²⁹) and the Library of Congress (for ISO639-2³⁰).

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."

5.5 LOCALE VOCABULARY

The proposed vocabulary for Locale is the same as for Language.

²⁷ Internet Engineering Task Force. RFC 5646. Tags for Identifying Languages. <http://www.ietf.org/rfc/rfc5646.txt>

²⁸ Lexvo.org <http://www.lexvo.org/>

²⁹ SIL International. ISO 639 Code Tables. <http://www.sil.org/iso639-3/codes.asp>

³⁰ Library of Congress. ISO 639.2 Registration Authority. Codes for the Representation of Names of Languages Part 2: Alpha-3 Code. <http://www.loc.gov/standards/iso639-2/langhome.html>

5.6 LICENCE VOCABULARY

On its website³¹, the Open Software Initiative [OSI], maintains a controlled vocabulary of open-source licences. The SPDX³² website also maintains a controlled vocabulary of open-source software licences derived from that.

5.7 LICENCE TYPE VOCABULARY

In addition to a reference to a specific licence a property is defined to classify the conditions and restrictions that are related to the specified licence.

The ADMS v1.00 specification [ADMS] proposes a list of licence types that was updated by the ADMS.SW Working Group. Please note that these licence types are intended to classify licences (mostly F/OSS licences) that are granted to an undetermined licensee, i.e. the right is granted to anyone.

<i>Label</i>	<i>Code</i>	<i>Id Definition</i>
Attribution	Attribution	http://purl.org/adms/licencetype/Attribution In general, most licences request to respect author's credits. Example: Article 5 EUPL: "the licensee shall keep intact all copyright notices.... in every distributed copy..." For the most permissive licences, this "Attribution obligation" (combined with a warranty disclaimer) is the main or sole condition for using, copying, performing and (re)distributing the work. Examples: CC by, MIT, BSD, ISA Open Metadata v1.1.
Public domain	PublicDomain	http://purl.org/adms/licencetype/PublicDomain Unlike in the US, most EU Member State's laws ignore volunteer dedication to Public domain (authors cannot resign their moral rights). Falling in public domain results only from copyright expiration. However, there are licences dedicating the work to the public domain by waiving all rights under copyright law, including all related and neighbouring rights, "to the extent allowed by law". Examples: CCO (Creative Commons Zero).

³¹ Open Source Initiative (OSI), Open Source Licenses by category, <http://opensource.org/licenses/category>

³² Software Package Data Exchange® (SPDX™), <http://www.spdx.org/licenses/>

Label	Code	Id Definition
Share alike / copyleft - not compatible/interoperable with other copyleft licences;	ShareAlike-NotCompatible	http://purl.org/adms/licencetype/ShareAlike-NotCompatible Share alike / copyleft means that, in case of redistribution of the work, this (same) licence must be reused. Not compatible / not interoperable with other copyleft licences means that the licence is copyleft on both source code and binaries, and is "self centric": the distribution of larger works integrating components covered by other licences must also be done under this licence only. This has been considered as invasive (sometimes "viral"). In the case of different copyleft provisions, licence conflict makes distribution legally impossible. Examples: GPL, AGPL, OSL.
Share alike / copyleft on source code or with compatibility exceptions for larger work and interoperability	ShareAlike-Compatible	http://purl.org/adms/licencetype/ShareAlike-Compatible Share alike / copyleft means that, in case of redistribution of the work, this (same) licence must be reused. Compatibility / interoperability exist in two cases: <ol style="list-style-type: none"> 1. The scope of the copyleft applies to the source code, not on binaries (therefore if a larger work is derivative of sources covered by various licences, then combined, compiled and statically linked binaries could distributed under any convenient licence). Examples: LGPL, EPL, APSL 2.0, CDDL, Ms-PL ... 2. There is full copyleft on the work (code and binaries), but the licence has implemented interoperability conditions and exceptions for distributing larger works under a controlled set of "secondary licences". Examples: EUPL v1.1, MPLv2.
Non-commercial use only	NonCommercialUseOnly	http://purl.org/adms/licencetype/NonCommercialUseOnly Licensees may copy, distribute, display and perform the work, and make derivative works based on it only for non-commercial purposes. These licences could present some interest for public sector. These licences are NOT Open Source /OSI approved: principle 6 of the Open Source Definition (OSD).
No derivative work	NoDerivativeWork	http://purl.org/adms/licencetype/NoDerivativeWork Licensees may copy, distribute, display and perform only verbatim copies of the work, and not derivative works based on it. These licences are more useful for art or literacy work than for software (except for specific security or warranty reasons). These licences are NOT Open Source /OSI approved: principle 3 of the Open Source Definition (OSD).

<i>Label</i>	<i>Code</i>	<i>Id</i> <i>Definition</i>
Royalties required	RoyaltiesRequired	http://purl.org/adms/licencetype/RoyaltiesRequired The management of royalties (price to pay per year, per user, per computer for using the software) is a frequent characteristic of proprietary licences (in large deals). Royalties are un-manageable in open source licensing, because re-distribution cannot be restricted: principle 1 of the Open Source Definition (OSD). Explicit exclusion of royalties (for example Article 2 EUPL) makes no obstacle for selling the software (i.e. a lump sum, n years contribution) or for cost sharing agreements (i.e. supporting a % of the development budget).
Reserved names / endorsement / official status	ReservedNames-Endorsement-OfficialStatus	http://purl.org/adms/licencetype/ReservedNames-Endorsement-OfficialStatus Except for attribution in copyright notices, some licences reserve the use of the name (of the author, of its employer /institution) because licensees cannot take advantage of an endorsement of the licensor in case they produce their own derivative. For example, an economic operator cannot promote its product as “official and based on the work of the European Commission”. Example: Article 5 EUPL: “This Licence does not grant permission to use the trade names, trademarks, service marks, or names of the Licensor...”.
Nominal cost	NominalCost	http://purl.org/adms/licencetype/NominalCost Requesting from licensees a nominal value or cost refers to a contribution (i.e. to the development costs of a software, or of a standard) expressed as a lump sum (that is, in units of a currency) paid in a given year or series of years by initial licensees. This is generally compatible with open source licensing (at the contrary of managed royalties).
Grant back	GrantBack	http://purl.org/adms/licencetype/GrantBack This is a licence under which licensor grants to recipient (or licensee) the right to use/improve/redistribute the work under the condition that the licensee grants back to the licensor a licence with respect to any improvements he/she made. This condition is mostly used in patent portfolio exchanges. It is compatible with open source software licensing: copyleft is a form of grant back (only in case of re-distribution). Stronger grant back allows a licensor to retain control of and access to any later developments associated with its technology. It is often considered bureaucratic and less efficient than pure share-alike/copyleft provisions.

<i>Label</i>	<i>Code</i>	<i>Id</i> <i>Definition</i>
Jurisdiction within the EU	JurisdictionWithinTheEU	http://purl.org/adms/licencetype/JurisdictionWithinTheEU Licence provision stating that the applicable law (and in some case also the competent court) must be (the law) of a EU member State. Judges have the facility to ask a prejudicial question to the unique European Court of Justice, as most relevant national provisions are derivative of EU Treaties and Directives). Example: Article 15 of the EUPL v1.1.
Other restrictive clauses	OtherRestrictiveClauses	http://purl.org/adms/licencetype/OtherRestrictiveClauses As the label states, this is to evaluate case by case. In open source licensing, additional agreements (i.e. a warranty, support services, jurisdiction/arbitration, applicable law, etc.) are valid if they do not restrict the rights granted by the licence (according to the principles of the Open Source Definition – OSD).
Known patent encumbrance	KnownPatentEncumbrance	http://purl.org/adms/licencetype/KnownPatentEncumbrance Patent encumbrance is a characteristic of specific products (software or standards) that cannot be used, reproduced and redistributed freely under any copyright licence, because a patent licence is also needed. Specific open source licences fight patent encumbrance (Art 11 GPL v3) or neutralise it to “the extent necessary to make use of the rights granted under this Licence” (Art 2 EUPL).
Unknown IPR	UnknownIPR	http://purl.org/adms/licencetype/UnknownIPR As the label says, IPR is unknown. Sole valid recommendation: Please do not use/reuse/redistribute such work, until clearly falling in public domain.

5.8 OPERATING SYSTEM

The operating system relationship associates a Software Project with an Operating System class. Both DBpedia and the Trove categories provide URIs that are suitable values for this such as:

- <http://sourceforge.net/api/trove/index/rdf#691> (J2ME)
- http://dbpedia.org/resource/Windows_7 (Windows 7)

On the basis of the CENATIC study the following minimal list is proposed.

<i>Label</i>	<i>Code</i>	<i>Id</i>
--------------	-------------	-----------

<i>Label</i>	<i>Code</i>	<i>Id</i>
Linux distributions	Linux	http://dbpedia.org/page/Linux
Windows	Microsoft_Windows	http://dbpedia.org/page/Microsoft_Windows
MAC OS	Mac_OS	http://dbpedia.org/page/Mac_OS
Multiplatform JAVA	JavaOS	http://dbpedia.org/page/JavaOS
Mobile devices	Mobile_operating_system	http://dbpedia.org/page/Mobile_operating_system

5.9 PROGRAMMING LANGUAGE

Both DBpedia and the Trove categories provide URIs that are suitable values for this, such as:

- <http://dbpedia.org/resource/Perl> (Perl)
- [http://live.dbpedia.org/page/Euphoria_\(programming_language\)](http://live.dbpedia.org/page/Euphoria_(programming_language)) (Euphoria)
- [http://live.dbpedia.org/page/Forth_\(programming_language\)](http://live.dbpedia.org/page/Forth_(programming_language)) (Forth)
- <http://live.dbpedia.org/page/Fortran> (Fortran)
- [http://live.dbpedia.org/page/Java_\(programming_language\)](http://live.dbpedia.org/page/Java_(programming_language)) (Java)
- <http://sourceforge.net/api/trove/index/rdf#178> (Python)

5.10 STATUS VOCABULARY

The Trove system [TROVE] provides URIs as suitable values for status, listed below.

<i>Label</i>	<i>Code</i>	<i>Id</i>
1 - Planning	planning	http://sourceforge.net/api/trove/index/rdf#7
2 - Pre-Alpha	prealpha	http://sourceforge.net/api/trove/index/rdf#8
3 - Alpha	alpha	http://sourceforge.net/api/trove/index/rdf#9
4 - Beta	beta	http://sourceforge.net/api/trove/index/rdf#10
5 - Production/Stable	production	http://sourceforge.net/api/trove/index/rdf#11
6 - Mature	mature	http://sourceforge.net/api/trove/index/rdf#12
7 - Inactive	inactive	http://sourceforge.net/api/trove/index/rdf#358

5.11 THEME VOCABULARY

Many candidate vocabularies are available for Theme. Examples are the Classification of Functions of Government (COFOG)³³, the European Commission's ECLAS³⁴ Thesaurus and EuroVoc, GEMET, AGROVOC, ZBW's STW Thesaurus for Economics³⁵, the Library of Congress' Subject Headings (LCSH)³⁶ and Thesaurus for Graphic Materials (TGM)³⁷ and others. The use of these vocabularies is closely linked to the theme to which an Asset is related. It is recommended that terms should be assigned from a vocabulary that is most widely used in the domain covered.

5.12 TOPIC VOCABULARY

The Trove system [TROVE] provide URIs as suitable values for topic, such as:

- <http://sourceforge.net/api/truve/index/rdf#826> (statistics)
- <http://sourceforge.net/api/truve/index/rdf#827> (knowledge management)
- <http://sourceforge.net/api/truve/index/rdf#828> (unattended installation)
- <http://sourceforge.net/api/truve/index/rdf#830> (voting)

For the public sector, we offer the following terms which were developed by Centro de Transferencia de Tecnología [CTT], based on a CENATIC study [CENATIC]. It should be noted that a Software Project may be associated with any number of Function classes via the topic relationship. Where the CTT classifications and other classifications overlap, best practice is to provide both. The intention is to use a set of persistent URLs for this classification.

<i>Label</i>	<i>Code</i>	<i>Description</i>
	virtual	
Citizens Engagement	citizen	Integrated offices, information phone lines, citizens folder, where is my transaction with the administration.
Web Tools	webTools	Searches, forum, geo-reference, etc
Electronic Processing	eProc	The applications that perform the electronic processing such us registries, management of grants, etc.
Support to Electronic Processing	eProcSupport	Products and services that makes possible electronic management such us digital signature, exchange of data, payment gateway, etc.
Management for internal Procedures	internalProc	Requests for materials, room reservations, etc.
Management of Finances	financial	Procurement, budgets, comptroller

³³ COFOG, <http://unstats.un.org/unsd/cr/registry/regcst.asp?Cl=4&Top=1&Lq=1>

³⁴ ECLAS, the Commission libraries' union catalogue. http://ec.europa.eu/libraries/doc/catalogues/index_en.htm

³⁵ Leibniz Information Centre for Economics ZBW. STW Thesaurus for Economics. <http://zbw.eu/stw/versions/latest/about>

³⁶ The Library of Congress. Subject & Genre/Form Headings. <http://www.loc.gov/aba/cataloging/subject/>

³⁷ The Library of Congress. Thesaurus for Graphic Materials. <http://id.loc.gov/vocabulary/graphicMaterials.html>

<i>Label</i>	<i>Code</i>	<i>Description</i>
Management of Human Resources	hr	Personnel management, payroll, time control, training, vacation
Management of knowledge and support to taking decisions	decisionMaking	Content Managers, dashboards, data ware house, etc
Infrastructure for Communications	comms	Networks and management tools
Infrastructure for Security	security	DMZ, proxies, DNS, IPS, backup, antivirus, etc
Infrastructure for Messaging	messaging	email, chat, Twitter, social networks, etc
Infrastructure for Managing Identities	eID	Digital signatures, LDAP, PKI, identity management, etc
Services and Systems Management	systems	Monitoring, statistics, managing of data processing centres managing of request of users, of bugs, service interruptions
Development and running Platforms	platforms	
Desktop Environment	desktop	Applications, virtualizations, models for PCs
Normalisation and Regularisation	standards	Methodologies, recommendations, specifications, etc.
Common Services for Public Administrations	publicServices	Common services offered, generally for free, to the rest of public administrations

5.13 USER INTERFACE TYPE

The Trove system [TROVE] provide 61 URIs as suitable values for user interface type, such as:

- <http://sourceforge.net/api/truve/index/rdf#237> (Web)
- <http://sourceforge.net/api/truve/index/rdf#232> (KDE)
- <http://sourceforge.net/api/truve/index/rdf#479> (Qt)
- <http://sourceforge.net/api/truve/index/rdf#314> (handhelds)
- <http://sourceforge.net/api/truve/index/rdf#459> (Command-line)

The following high-level values are proposed.

- Shell (command line)
- Graphical interface
- Auditory interface
- Web browser

Other prefixes and namespaces used in the document are as follows

Prefix	Namespace
dcterms	http://purl.org/dc/terms/
doap	http://usefulinc.com/ns/doap#
foaf	http://xmlns.com/foaf/0.1/
schema	http://schema.org/
sf	http://sourceforge.net/api/sfelements.rdf#
skos	http://www.w3.org/2004/02/skos/core#
spdx	http://spdx.org/rdf/terms#
vcard	http://www.w3.org/2006/vcard/ns#
qb	http://purl.org/linked-data/cube#

7. ACKNOWLEDGEMENT

Specific acknowledgement is due to:

- Adam Arndt
- Olivier Berger
- Rüdiger Czieschla
- Alfonso De Cala
- Jose Angel Diaz
- Roberto Galoppini
- Bart Hanssens
- Roland Mas
- Elena Muñoz Salinero
- Serafín Olcoz
- Alain Peyrat
- Alisson Randal
- Patrice-Emmanuel Schmitz
- Savino Sguera
- Rashid Tariq
- Sander van der Waal
- Szabolcs Szekacs
- Tony Wasserman

7.1 ADMS.SW WORKING GROUP

<i>First Name</i>	<i>Surname</i>	<i>Organisation</i>	<i>Country</i>	<i>Additional Info</i>
Miguel Angel	Alvarez	University of Seville	ES	
Phil	Archer	W3C	UK	Member of SEMIC.eu team
Adam	Arndt	Danish Agency for Digitisation	DK	ISA Member State representative
Olivier	Berger	INF department of TELECOM SudParis Institut Mines Telecom	FR	Research Engineer
Juan	Conde	Guadalinx	ES	
Rüdiger	Czieschla	Stadt Freiburg Open-source Business Alliance	DE	Head of IT dept. - City of Freiburg, Germany
Alfonso	De Cala	Junta de Andalucia	ES	Member of osor.eu national forges community
Jose Angel	Diaz	CENATIC	ES	
Edd	Dumbill	O'Reilly Media	US	creator of DOAP
Carlos	E. Jimenez	Independent Consultant	ES	e-Government & Interoperability Consultant
Roberto	Galoppini	SourceForge Senior Director of Business Development at Geeknet	IT	SourceForge
Greg	Grossmeier	Creative Commons	US	
Björn	Hagström	Swedish eGovernment Delegation	SW	

Morten Kristoffer	Hansen	Danish National IT and Telecom Agency; Center for Digitalisering	DK	Member of osor.eu national forges community; Project leader, Videncenter for open source og åbne standarder
Bart	Hanssens	FEDICT	BE	Member State Representative
Roland	Mas	Adullact	FR	
Jaque	Miguel	CENATIC	ES	
Elena	Muñoz	Ministry of Territorial Policy and Public Administration	ES	Spanish repository at national level
Serafín	Olcoz		ES	Director of Informatics and Telecommunications for the Basque Government
Juan Antonio	Ortega		ES	
Dolores	Pallero Espadero	Regional Government of Extremadura	ES	First Vice President and Spokeswoman o the Regional Government of Extremadura
Thodoris	Papadopoulos	Ministry of Public Reform and E-Governance	GR	
Bruce	Perens	Open Source Initiative (OSI)	US	Created the Open Source Definition and published the first formal announcement and manifesto of open source. He co-founded the Open Source Initiative (OSI) with Eric S. Raymond
Alain	Peyrat	Alcatel	FR	FusionForge contributor
Simon	Phipps	Open Source Initiative	UK	
Pop	Ramsamy	CENATIC	ES	
Allison	Randal	Canonical	US	
Tariq	Rashid	Home Office - Uk Government	UK	Architect
Nikos	Roussos	KEDKE OSEPA	GR	Expert in Free Software, organization leader at Osepa project
Patrice-Emmanuel	Schmitz	Unisys	BE	Independent Expert
Savino	Sguera	nexB	IT	Software Engineer at nexB
Gofran	Shukair	DERI University of Galway	IE	External expert
Szabolcs	Szekacs	European Commission - DG DIGIT B2	HU	project officer
Ann	Tran	The Open Technology Foundation	AU	
Sander	Van Der Waal	OSS Watch Simal	NL	Service Manager of OSS Watch
Neven	Vrček	University of Zagreb	HR	External Expert

7.2 THE SEMIC TEAM

- Phil Archer, W3C
- Michiel De Keyzer, PwC
- Makx Dekkers, AMI Consult
- Débora Di Giacomo, PwC
- João Rodrigues Frade, PwC
- Stijn Goedertier, PwC
- Saky Kourtidis, PwC
- Thomas Rössler, W3C
- Kelly Liljemo, PwC
- Niels Van Hee, PwC
- Rigo Wenning, W3C

8. REFERENCES

- [ADMS] Asset Description Metadata Schema,
<http://joinup.ec.europa.eu/asset/adms/description>
- [CENATIC] Anexo IV: Estudio Taxonomías Forjas, CENATIC, 2011 at
<http://fng.morfeo-project.org/wp-content/uploads/2011/07/Anexo-IV.-Estudio-taxonomias-forjas-v01r001.pdf>
- [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/](http://dbpedia.org/)
- [DCAT] The Data Catalog Vocabulary
<http://www.w3.org/TR/vocab-dcat/>
- [DCTERMS] DCMI Metadata Terms, Dublin Core Metadata Initiative.
<http://dublincore.org/documents/dcmi-terms/>
- [DEBTAGS] Debian Package Tags, <http://debtags.aliioth.debian.org/vocabulary/>
- [DOAP] Description of a Project
<http://trac.usefulinc.com/doap>
- [FOAF] Friend of a Friend
<http://xmlns.com/foaf/spec/>
- [FRBR] Functional requirements for bibliographic records. IFLA. (2008). Retrieved from <http://www.ifla.org/publications/functional-requirements-for-bibliographic-records>
- [FRSAD] Functional Requirements for Subject Authority Data (FRSAD) . IFLA. (2010). Retrieved from <http://www.ifla.org/en/news/functional-requirements-for-subject-authority-data-frsad-final-report-published>
- [Forges] ISA Deliverable: Report on existing Software Forges at
<https://joinup.ec.europa.eu/elibrary/document/isa-deliverable-report-existing-software-forges>
- [ISA] Interoperability Solutions for European Public Administrations,
<http://ec.europa.eu/isa/>

-
- [ISO 19770-1] Software asset management -- Part 1: Processes and tiered assessment of conformance
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=56000
- [ISO 15288:2008] Systems and software engineering -- System life cycle processes
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43564
- [ISO 19770-2] Software asset management -- Part 2: Software identification tag
http://www.iso.org/iso/catalogue_detail.htm?csnumber=53670
<http://standards.iso.org/iso/19770/-2/2009/schema.xsd>
- [ISO 3166-1] ISO 3166 short country names code lists, ISO.
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
- [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/>
- [OSI] Open Software Initiative, Licence by category
<http://opensource.org/licenses/category>
Opensource.org site content is licensed under a [Creative Commons Attribution 2.5](https://creativecommons.org/licenses/by/2.5/) License.
- [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>
- [RADion] RADion – Repository, Asset, Distribution.
<http://joinup.ec.europa.eu/asset/radion/release/01>
- [RFC 3066] H. Alvestrand, ed. RFC 3066: Tags for the Identification of Languages 1995. Available at: <http://www.ietf.org/rfc/rfc3066.txt>
- [RFC3986] RFC 3986 Uniform Resource Identifier (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter January 2005

-
- [RFC3987] RFC 3987, "Internationalized Resource Identifiers (IRIs)", M. Dürst , M. Suignard
- [SIOC] SIOC Core Ontology Specification, Uldis Bojārs, John Breslin (Eds)
<http://rdfs.org/sioc/spec/>
- [SKOS] SKOS Simple Knowledge Organization System Reference
W3C Recommendation 18 August 2009
<http://www.w3.org/TR/skos-reference/>
<http://www.w3.org/2004/02/skos/>
- [SPDX] Software Package Data Exchange (SPDX), <http://www.spdx.org/licenses/>
(©Linux Foundation and its Contributors 2010-2011 [Creative Commons Attribution License 3.0 unported](#))
- [TROVE] The Trove System is made available and maintained by Sourceforge.net
(© SourceForge 2012 CC by)
<http://sourceforge.net/apps/trac/sourceforge/wiki/Software%20Map%20and%20Trove>
- [TROVECAT] The Trove classification of categories of software is published as RDF/XML at
<http://sourceforge.net/api/trove/index/rdf>
- [Vision] ISA Deliverable: The vision for an enhanced software description metadata schema and software catalogue for e-Government at
<https://joinup.ec.europa.eu/elibrary/document/isa-deliverable-vision-enhanced-software-description-metadata-schema-and-software->
- [XSD] XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004. <http://www.w3.org/TR/xmlschema-2/#date>