European Commission

e-Government Core Vocabularies handbook

Using horizontal data standards for promoting interoperability

ISA

This report was prepared for the ISA Programme by:
PwC EU Services

Disclaimer:
The views expressed in this report 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.

Cover picture : © Thinkstock - mon5ter

pwc

This report was prepared for the ISA Programme by:
PwC EU Services

Europe Direct is a service to help you find answers to your questions about the European Union.

Freephone number (*):
00 800 6 7 8 9 10 11

(*) The information given is free, as are most calls (though some operators, phone boxes or hotels may charge you).

More information on the European Union is available on the Internet (http://europa.eu).
Luxembourg: Publications Office of the European Union, 2015

ISBN 978-92-79-50129-6 (print)
ISBN 978-92-79-50130-2 (web)
doi:10.2799/945287 (print)
doi:10.2799/52286 (web)

Printed in Belgium
PRINTED ON ELEMENTAL CHLORINE-FREE BLEACHED PAPER (ECF)

Disclaimers

The views expressed in this report 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 report has been carefully compiled by PwC, but no representation is made or warranty given (either express or implied) as to the completeness or accuracy of the information it contains. PwC is not liable for the information in this presentation or any decision or consequence based on the use of it. PwC will not be liable for any damages arising from the use of the information contained in this presentation. The information contained in this presentation is of a general nature and is solely for guidance on matters of general interest. This presentation is not a substitute for professional advice on any particular matter. No reader should act on the basis of any matter contained in this publication without considering appropriate professional advice.

CONTENTS

1 Introduction

1.1 The need for semantic interoperability

In Europe, citizens and businesses increasingly live, work, and conduct business across borders. Their increased mobility must be supported by cross-border public services, such as the registration of a foreign branch, obtaining a licence to conduct business in another country, or getting a birth certificate. In such an environment, public administrations must be capable of interacting efficiently and effectively across borders through the seamless exchange of data. Unfortunately, the environment in which data exchange takes place amongst EU Member States is complex, creating many semantic interoperability conflicts during the execution of European public services. Such *semantic interoperability conflicts* are caused by discrepancies in the interpretation of administrative procedures and legislation, the lack of commonly agreed data models, the absence of universal reference data, etc.

In order to overcome these conflicts, the ISA Programme has created the *Core Vocabularies* specifications in the period 2011-2013 with international Working Groups of Member State representatives and experts. The Core Vocabularies are defined as *simplified, re-usable and extensible data models that capture the fundamental characteristics of a data entity in a **context-neutral** and **syntax-neutral** fashion* [1].

1.2 Scope

This handbook provides a *generic* approach for designing and mapping data models based on the Core Vocabularies. The approach is syntax-neutral, i.e. *independent of any technical representation*, and can be used together with other methodologies for creating information system data models, information exchange data models or linked data models. The handbook therefore does not cover the following aspects:

- The handbook does not provide a complete methodology for syntax binding and toolset for creating data model specifications, as the Core Vocabularies intend to remain syntax-neutral.
- The handbook does not recommend any methodology, data model library, or tool above another. Such choices are heavily dependent on the domain and the task at hand.

1.3 The Core Vocabularies as an enabler for reducing semantic interoperability conflicts

The process of providing cross-border public services across EU Member States is complex, due to the heterogeneity of the actors, information and services of the different Member States. The complexity of exchanging data may lead to *semantic interoperability conflicts* [2]. The Core Vocabularies can be used to reduce these semantic conflicts in two ways:

- **The design of new data models that extend the Core Vocabularies:** new data models that make use of the Core Vocabularies as building blocks, either at conceptual level or at syntax-level, guarantee a minimum of cross-sector interoperability.
- **The mapping of existing data models to the Core Vocabularies:** existing data models that have mappings to the Core Vocabularies allow using the Core Vocabularies as a common foundational data model allowing to bridge different data models.

The remainder of this section uses the classification of semantic conflicts in [2] and illustrates how the Core Vocabularies can alleviate these.

1.3.1 Reducing schema-level conflicts

Schema-level conflicts are caused by differences in logical structures and/or inconsistencies in metadata. The Core Vocabularies can be used to alleviate different types of conflicts at schema level [2]:

- ***Naming:*** different names for the same concept (synonyms), or the same name for different concepts (homonyms). For example, the term 'citizen' or 'national' can be synonyms in a specific context. Naming conflicts can be overcome by agreeing on the use of common names when designing new data models (where possible reusing the terms from the Core Vocabularies), or by mapping existing data models (where possible to the Core Vocabularies).

- ***Entity identifier:*** different identifiers for the same object. For example, a person can be identified by his ID number, as well as by his National Register number. Entity identifier conflicts can be overcome by agreeing on the use of common identifiers for information exchanges. In cases where no common identifiers are used, the Core Vocabularies data type 'Identifier' allows documenting the context of identifiers used in information exchanges, such as the issuing authority and/or type, allowing different identifiers to be mapped.

- ***Schema-isomorphism:*** the same concept is described by different attributes. For example, the concept 'person' can be described using different attributes. These conflicts can be partially alleviated by agreeing on the attributes using the Core Vocabularies as a minimal common set.

- ***Generalisation:*** similar concepts, but with a broader/narrower scope, for example, a legal entity is described in one model as a generic "organisation", whereas in another model this is described as a "registered organisation" with a legal status. The Core Vocabularies provide common generalisations for similar concepts. When mapping the Core Vocabularies to existing data models, a given concept in the Core Vocabularies may have narrow matches with similar context-specific concepts. In this case, a comment has to be provided with the mapping describing how the concept is constrained in this specific context at hand. These mappings create bridges between data models that are pockets of semantic interoperability. Similarly, when designing new data models, a context-specific information requirement may require including a more specific concept that has narrow matches with a given concept in the Core Vocabularies.

- ***Aggregation:*** multiple attributes in one model aggregated together as one attribute in the other model. For example, one model describes both "family name" and "first name", while another model combines these properties into one term "full name". The Core Vocabularies may alleviate this problem, as they set a common granularity for data models. In addition, aggregated properties for less granular data models could be mapped to each relevant property of the Core Vocabularies. Mappings between data models and the Core Vocabularies may help unveil such interoperability conflicts.

- ***Schematic discrepancies:*** a set of attributes of one entity in one model is organised in a different structure of entities in another model. This conflict is due to the design of the data model. For example, for the description of a legal entity in UBL, the Party Legal Entity ABIE (aggregated business information entity) class is used as part of the Party ABIE. The structure of the legal entity class in UBL is different from the Core Vocabularies LegalEntity conceptual data model. Intended to be syntax-neutral, the use of the Core Vocabularies may alleviate schematic discrepancies, as they focus on mappings of atomic elements (class, property, relationship) in the conceptual model of the Core Vocabularies to other data models. Each element (class, property, or relationship) in a data model may have a mapping to a corresponding element in the Core Vocabularies. At instance level, each fact expressed using elements for which there exists an exact match or a broad match can then be converted without (or with limited) loss of meaning.

1.3.2 Reducing data-level conflicts

Data-level conflicts are caused by differences occurring in data domains due to multiple possible representations and interpretations of similar data [2].

The Core Vocabularies mainly concern conceptual data models and do not much affect the representation and interpretation of the data. The generic methodology presented in Section 4 for designing context-specific domain data models or information exchange specifications, foresees adding additional specifications regarding the use of controlled vocabularies (code lists) and business rules. Multiple types of data-level conflicts exist, i.e. differences in the value, representation, units and precision of the data [2].

- ***Data value:*** different meanings for the same value, or different values for the same meaning. For example, in a controlled vocabulary, the term "Gender" can have different code lists.
- ***Data representation:*** data is represented using different formats. For example, date formats can be represented as DMY, MDY, YMD; decimal marks can be represented by a space, comma, full stop, apostrophe, etc.
- ***Data units:*** data is expressed in different units of measurement. For example, data can be measured in the metric system of units or in imperial and US customary units: mass can be measured in kg or lbs; distances can be measured in kilometres or miles; etc.
- ***Data precision:*** data is expressed using different granularity, for example, when assigning grades using a three-point scale (A – B – C) or a five-point scale (very good – good – neutral – bad – very bad).

Naming	Entity identifier	Schema isomorphism	Generalisation	Aggregation	Schematic discrepancies
• Homonyms: e.g. 'citizen' and 'national' • Synonyms: 'resident' and inhabitant'	• Citizens identified by ID card number or national number	• Different attributues on ID cards in different states	• Birth certificate of one state can contain all infor of birth and family certificates of another state	• 'full name' or 'surname'; 'middle name'; 'first name'	• Detailed information cannot be exchanged due to schematic differences (ex. different xml schemas)

Figure 1: Examples of schema-level conflicts

1.4 Use cases for the Core Vocabularies

This handbook aims to be a practical user guide on how to use the Core Vocabularies, covering the following two use cases:

- The design of new data models that extend the Core Vocabularies; and
- The mapping of existing data models to the Core Vocabularies.

Data models that map to or extend the Core Vocabularies guarantee a minimum level of cross-sector and/or cross-border interoperability that can be attained by public administrations.

1.4.1 Use case: design new data models that extend the Core Vocabularies

The Core Vocabularies can become the basis for the design of context-specific data models. These can include:

- ***Domain data models:*** the Core Vocabularies can be used as a *domain data model*. One example here is to use the Core Vocabularies as the foundation of a common publication format for data in *base registries* like cadastres, business registers and service catalogues. These domain data models can also be used as a default starting point for designing the conceptual and logical data models in newly developed information systems.
- ***Information exchange data models:*** the Core Vocabularies can become the basis of a context-specific information exchange data model used to exchange data between information systems. One example here is the use of the Core Vocabularies for defining information exchange data models such as a 'Business Activity Registration Request', which could be used for registering new business activities for a foreign branch of a legal entity in another EU Member State.

For a data model to be conformant to the Core Vocabularies, a mapping of the data model to the conceptual model of the Core Vocabularies must be published. These conformance guidelines are explained in Section 3.5.

1.4.2 Use case: mapping existing data models to the Core Vocabularies

The Core Vocabularies can be used to annotate new and existing data models with mappings to the Core Vocabularies, as explained in Section 3.5. This handbook shows how mappings should be created and published. These mappings can in turn be used for:

- ***Alignment of data models:*** The analysis of existing relationships between data models, through their mapping with the Core Vocabularies, provides insights in possible alignments. Such alignments are needed in order to resolve semantic interoperability conflicts between different data models.
- ***Reconciliation of data sources:*** Data sources of which the data models are mapped to the Core Vocabularies possess a minimum of semantic alignment. This means that two data sources with information on persons, legal entities, or addresses could be (semi)-automatically merged. The mappings can also be used by implementers of interoperability solutions to easily extract basic information from artefacts of conformant information exchanges.

1.5 Structure of the document

This handbook gives an overview of how organisations can put the Core Vocabularies into practice. The document is structured as follows:

- Chapter 2 provides an overview of the Core Vocabularies indicating how they were *created* and *maintained* and how they can be used under a permissive licence;
- Chapter 3 explains the *data modelling concepts* used to define the Core Vocabularies. It also defines *mappings relationships*. These mapping relationships are important for claiming conformance; and
- Chapter 4 contains a detailed methodology for the two main use cases of the handbook, indicating how to *design data models* using the Core Vocabularies and *how to document mappings*.

2 Overview of the Core Vocabularies

This section contains a brief overview of the Core Vocabularies, indicating how they were developed and how they are maintained.

2.1 Overview

The ISA programme of the European Commission facilitated international working groups to forge consensus on four Core Vocabularies. The Core Vocabularies have been developed following the 'Process and methodology for developing Core Vocabularies' [3].

The Core Vocabularies are subject to an open change and release management process, supported by the ISA Programme of the European Commission.

A short description of these vocabularies is included in Table 1. The latest release of the Core Vocabularies can be retrieved via Joinup[1].

[1] https://joinup.ec.europa.eu/asset/core_vocabularies/description

Vocabulary	Description
CORE BUSINESS VOCABULARY	The Core Business Vocabulary is a simplified, reusable and extensible data model that captures the fundamental characteristics of a legal entity, e.g. the legal name, the activity, address, legal identifier, company type, and its activities. On 8 January 2013, the RDF representation of the Core Business Vocabulary has been formally published on the W3C standards track as a Public Working Draft. It has been revised and renamed into Registered Organization Vocabulary[2]. The development of the Core Business Vocabulary was chaired by DG MARKT/F2, involved in the Business Register Information System (BRIS) project.
CORE Location VOCABULARY	The Core Location Vocabulary is a simplified, reusable and extensible data model that captures the fundamental characteristics of a location, represented as an address, a geographic name or a geometry. The Core Location Vocabulary is aligned with the INSPIRE data specifications. The development of the Core Location Vocabulary was chaired by JRC/H6 Digital Earth and Reference Data involved in the implementation of the INSPIRE Directive
CORE PERSON VOCABULARY	The Core Person Vocabulary is a simplified, reusable and extensible data model that captures the fundamental characteristics of a person, e.g. the name, the gender, the date of birth, etc. The development of the Core Person Vocabulary was chaired by Eurojust, which promotes interoperability in the judicial domain via the EPOC IV project.
CORE PUBLIC SERVICE VOCABULARY	The Core Public Service Vocabulary is a simplified, reusable and extensible data model that captures the fundamental characteristics of a service offered by public administration. Such characteristics include the title, description, inputs, outputs, providers, locations, etc. of the public service. The development of the Core Public Service Vocabulary was chaired by Theodoros Papadopoulos, a representative from the Greek Ministry of Administrative Reform and E-Governance.

Table 1: Overview of the ISA Core Vocabularies

The Core Vocabularies are *context-neutral* semantic building blocks that can be extended into context-specific data models, in a way that is similar to the approach used by the UN/CEFACT Core Component Technical Specification [4] or the US National Information Exchange Model [5]. Hereby it is insightful to consider three levels of abstraction, depicted in Figure 2:

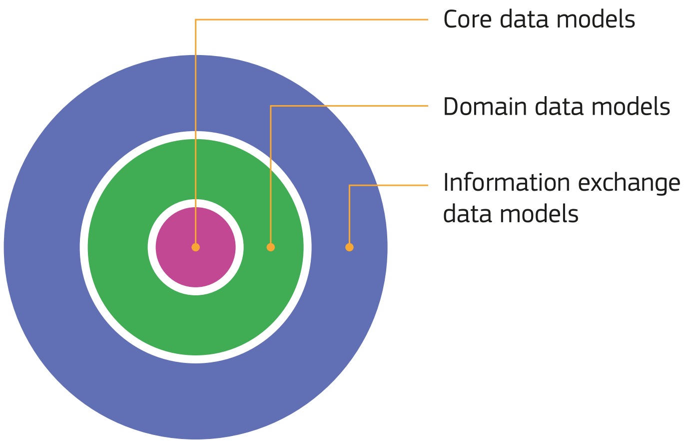

Figure 2: Three levels of abstraction for data models

- ***Core data model:*** a context-neutral data model that captures the fundamental characteristics of an entity. The Core Vocabularies are at this level.
- ***Domain data models:*** a data model modelling a particular domain (e.g. the justice domain, the healthcare domain) that identifies the entities involved and their relationships.
- ***Information exchange data models:*** a data model that defines and describes the structure and content of information that is exchanged between organisations in a specific information exchange context.

[2] RegOrg: http://www.w3.org/TR/vocab-regorg/

The use of the Core Vocabularies as a common building block for developing context-specific data models guarantees a *minimum* of semantic consistency. When the Core Vocabularies are extended to create domain models and information exchange data models, additional meaning (semantics) is added to the specifications, due to the *contextualisation*.

2.2 Representation formats

The Core Vocabularies are by definition syntax-neutral, indicating that they focus on fundamental characteristics of data entities rather than on the specific representation. The specifications of the Core Vocabularies are published[3] as a spreadsheet and a consolidated UML model capturing the meaning of the Core Vocabularies at conceptual level. The spreadsheet contains an identifier, label, and definition for each term in the Core Vocabularies.

The Core Vocabularies are expressed in three formats: a conceptual model and spreadsheet, XML schema and RDF schema.

- Conceptual model: the conceptual model of the Core Vocabularies allows understanding the meaning of the data model. Generally, the conceptual data model is the most important. It is represented in the form of a spreadsheet and as a static model[4] in the Unified Modelling Language (UML).
- XML Schema: the XML schemas represent the above concepts by following the UBL naming and design rules (NDRs) and by being based on the UBL Unqualified Data Types.
- RDF Schema: the RDF schemas represent the above concepts based on existing RDF vocabularies, including the Dublin Core Terms[5], FOAF[6], SKOS[7], etc.

2.3 Licensing conditions

The Core Vocabularies are licensed under the ISA Open Metadata Licence v1.1 [6].

[3] https://joinup.ec.europa.eu/asset/core_vocabularies/description

[4] http://joinup.ec.europa.eu/site/core_vocabularies/Core_Vocabularies_v1.1/Core_Vocabularies_v1.1.htm

[5] http://dublincore.org/documents/dcmi-terms/

[6] http://xmlns.com/foaf/spec/

[7] http://www.w3.org/2004/02/skos/

3 Data modelling concepts

This chapter explains how the conceptual model of the Core Vocabularies is structured.

3.1 Modelling conventions

The conceptual model of the Core Vocabularies follows the following conventions.

- The names of all elements are title-cased (i.e., the first letter of each word is written in uppercase) and consist solely of alphanumeric characters and spaces.
- The identifier of a class or data type is the name of the element, where spaces have been removed. For example, the identifier of the "Legal Entity" class is "LegalEntity".
- The identifier of a property, association, or attribute is the concatenation of the identifier of the parent class or data type, and the name of the element, where spaces have been removed. For example, the identifier of the "Company Type" property of the "Legal Entity" class is "LegalEntityCompanyType".
- To avoid identifier clashes, a class or data type name must not be a prefix of another class or data type name. For example, a class named "Address Thoroughfare" would be identified by "AddressThoroughfare", which would conflict with the identifier of the "Thoroughfare" property of the "Address" class.
- Each element is defined by a crisp, one-line definition. The definition starts with a capital letter and ends with a period.
- A description may provide complementary information concerning the usage of the element or its relation to relevant standards. For example, a description may contain recommendations about which controlled vocabularies to use. Descriptions may contain multiple paragraphs separated by blank lines. The descriptions should not paraphrase the definitions.
- Examples are provided as a comma-separated list. Each example value is enclosed in quotes and is optionally followed by a short explanation enclosed in parentheses.
- The UML modelling conventions are based on the conventions mentioned above. Furthermore, the UML diagrams follow the specifications described below.
- All data types are defined in the package "DataTypes". We make the distinction between primitive types (consisting of singular values) and composite types (composed of multiple attributes).
- The terminology between the Core Vocabularies and UML is mapped. For example, property is mapped to property, association is mapped to association, etc.

3.2 Model characteristics

The Core Vocabularies are *simplified, re-usable and extensible data models that capture the fundamental characteristics of a data entity in a **context-neutral** and **syntax-neutral** fashion*.

This means that the scope of the Core Vocabularies contains:

- ***No controlled vocabularies:*** the Core Vocabularies do not propose value vocabularies, i.e. code lists, taxonomies, etc. This is left to the context-specific vocabularies to specify.
- ***No range restrictions:*** the Core Vocabularies do not include cardinality constraints on associations or attributes. This is left to the context-specific data models to specify on the basis of domain-dependent business rules.
- ***No temporal aspects:*** the Core Vocabularies do not propose a mechanism to deal with temporal aspects. For example, if a country allows a person to change his given name, the Core Vocabularies do not include a mechanism to record the given name of a person at a specific point in time. This is left to the context-specific data models to specify.
- ***No administrative metadata:*** the Core Vocabularies do not include any administrative guidelines on what should be done with data. For example, it does not contain any guidance on how to deal with personally identifiable information in the context of personal data protection regulations. This is left to the context-specific data models to specify.

3.3 Customising the conceptual model

The Core Vocabularies are not meant to be complete to fit the requirements of any domain data model or information exchange data model. This means that using the Core Vocabularies as a basis for a data model specification involves extending or restricting them.

There are various ways to do so, each with their own consequences for interoperability:

- ***Adding new properties or associations to a class***

The most common extension is to add new properties and associations to existing classes. For example, a data model might need to include a property indicating whether a person has a driving licence.

The consequences for interoperability are limited, as systems that do not understand the new properties and associations can usually just drop them and still benefit from the information contained in the common properties.

- ***Removing irrelevant properties and associations from a class***

Some properties may be irrelevant in a particular domain or for a specific information system. For example, the date of death of a person will not be useful when providing services only to living people. In such cases, the irrelevant properties or associations may be removed from the resulting data model.

The direct consequence for interoperability is that a system might be missing information that it would expect from Core Vocabularies-conform models. Hence, removing properties and associations reduces the number of systems with which the data model is interoperable. However, it is the context of information exchange that determines whether the absence of properties and associates can be tolerated or not.

On the other hand, if no new properties are added and only some properties removed, there is a guarantee that systems will understand all the remaining properties.

Note also that when both adding and removing properties is allowed, this can lead to a situation where no *common* properties remain. This is however unlikely because of the fundamental character of the core properties. In general, it is desirable for all IT systems aiming at interoperability to understand at least these “core” elements (where applicable). If some “core” information is present in the systems participating in a given information exchange but not needed in all communication scenarios between those systems, it may make sense to first define a more general data model for communication between these systems that includes all core elements, and make sure that all systems “understand” this general model. That general model can then be restricted as needed for any given communication scenario.

- ***Specialising classes, properties, or associations***

When using the Core Vocabularies in an information system or information exchange data model, the context refines the meaning of the elements. For example, in a clinical context, a person might be a patient or a doctor. The two specialisations will probably have different additional properties and associations, but share the common properties of the Person class in the Core Vocabularies.

Specializing elements may lead to generalization conflicts. Indeed, a patient in a clinical context might not always correspond to a criminal in a judicial context, even though both are specializations of the Person class. However, the common properties and associations defined in the Core Vocabularies, such as the name, should still be valid for the person referred to by the criminal or the patient. Hence, care must be taken when specializing elements so that the resulting elements can still be understood with the semantics of the original elements using the mapping relations “narrow match” as described in Section 3.4.

- ***Replacing classes, properties, or associations***

Sometimes, an element from the Core Vocabularies might be close to the requirements of a data model, but still not match them exactly. In such cases, one might replace the element with a new one.

Replacing elements has a huge impact on interoperability and is strongly discouraged. Schema-level conflicts, such as schema isomorphism conflicts and schematic discrepancies may occur. To mitigate those issues, it is recommended to use another name (in order to avoid naming conflicts) and to document the difference in the “close match” or “related match” mapping.

3.4 Mapping relations

The Core Vocabularies do not recommend any particular technical representation (syntax binding) of the conceptual model. Instead, mappings to other existing standard data models are provided. In the same manner, mappings to the data models of information systems can be provided. Mappings serve two purposes:

- ***Documentation:*** experts with deep knowledge of one data model can leverage that knowledge to understand another data model by using the mapped core concepts as entry points. Such mappings can also give insights in using existing syntax bindings when designing new data models.
- ***Reconciliation of data sources:*** interoperability solutions can use the mappings to extract basic information from conformant data models.

The mapping of the Core Vocabularies to another data model contains relations between classes, properties, associations, data types and attributes of the Core Vocabularies on the one hand and elements of the other data model on the other hand. The Core Vocabularies borrow the exact, close, related, broad and narrow match mapping relations of the SKOS vocabulary [7].

Intuitively, two elements have an *exact match* if their definitions are strictly equivalent. For example, the Person class is defined as "a natural person" in the Core Vocabularies, and as "an individual human being" in the UN/CEFACT CCL. Both classes cover the same set of individuals and thus have an exact match. The match is at the semantic level. Note that an exact match does not imply that the Person classes are structurally equal, i.e., they may define additional or remove existing properties and associations; schema-isomorphism is still possible.

If the definitions are not strictly equivalent and the differences are negligible, the *close match* relation is used. For example the "Address Post Name" ("the key postal division of the address, usually the city") is a close match of UN/CCL's "Address. City Name. Text" ("the name, expressed as text, of the city, town or village of this address"). In most cases, the two properties will contain the same value, but there might be situations where the key postal division is neither a city nor a town nor a village.

The *narrow match* relation indicates that the definition of one element is more general than the definition of the other element. For example, the "Address Locator Designator" ("a number or a sequence of characters that uniquely identifies the locator within the relevant scope") has a narrow match with the UN/CCL's "Address. Building Number. Text" ("the number or alphanumeric designation, expressed as text, of a building or house at this address"). The definition of Address Locator Designator includes all designators of buildings and houses, as well as many other designators. The inverse relation of a narrow match is a *broad match*.

The *related match* relation is the loosest mapping relation. It indicates that two elements are somehow related. As any element can be somehow related to any other element, this relation should only be used if the relation is meaningful. For example, "Person Birth Name" ("the full name of the person at the time of her birth, irrespective of any subsequent changes") is a related match of UN/CCL's "Person. Maiden Name. Text" ("a family name, expressed as text, of a person before first marriage"), because many persons do not change name before their first marriage.

To formally define the mapping relations, we need to distinguish between an element and its subjects, i.e., the real-world things that are represented by the instances of the element.

An **element** is a class, a property, or an association.

Classes are instantiated into **instances** (individuals) representing a real-world physical or conceptual thing. Classes are used to express **facts** about the generic characteristics of an individual instance. For example, the fact that an individual 'Gotlobb Frege is a (instance of the class) Person'. **Properties** or **associations** are used to construct **facts** about individuals. The property 'Family name' can be used in the fact 'Frege is the family name of Gotlobb Frege'.

For example, if John Doe is a natural person, he is an instance of the Person class. The fact of having "John" as given name is an instance of the *Given Name* property. Similarly, the fact of being born in the country Germany is an instance of the *Country Of Birth* property.

The mapping relations are defined in terms of the set of subjects covered by the elements. Figure 3 shows a graphical representation of the mapping relations.

A **has an exact match** B if the set of subjects of A is equal to the set of subjects of B. The definitions of A and B are equivalent.

A **has a close match** B if the set of subjects of A is mostly equal to the set of subjects of B. The number of subjects of A not included in B, and vice-versa, is negligible.

A **has a related match** B if there is a meaningful intersection between the subjects of A and the subjects of B.

A **has a narrow match** B if the set of subjects of A is a superset of the set of subjects of B. The definition of A generalizes the definition of B.

A **has a broad match** B if B has a narrow match A.

These mapping relations have consequences for reconciling data sources, i.e. the conversion of a data source from one data model into another data model:

- ***Exact match:*** Facts that are expressed for elements (classes, properties, or relationships) with an *exact match* relationship can be *converted in both directions* between data models without loss of meaning. For example, if there is an exact match between the element 'foaf:familyName' and 'nc:PersonSurName', such facts in data sources using either the FOAF or NIEM Core data models can be converted in both directions without loss of meaning.
- ***Narrow/broad match:*** Facts that are expressed for elements (classes, properties, or relationships) with a *narrow/broad* match relationship can only be *converted into only the direction of the more general data model element*. For example the 'LegalEntity' concept in the Core Vocabularies has a broad match with 'Organization' in the UN/CEFACT Core Component Library. This means that all facts in one data source about individuals being instances of the class 'LegalEntity' can be converted into another data model, stating that they are individuals of the class 'Organization'. The opposite conversion is not true.
- ***Close match:*** Elements with a close match relationship can be *converted in both directions* between data models with a minimal loss of meaning for some individuals.
- ***Related match:*** Elements with a related match relationship can only be converted with considerable error. Conversion is not advised. Such mappings can still be valuable to make semantic conflicts between data models better visible.

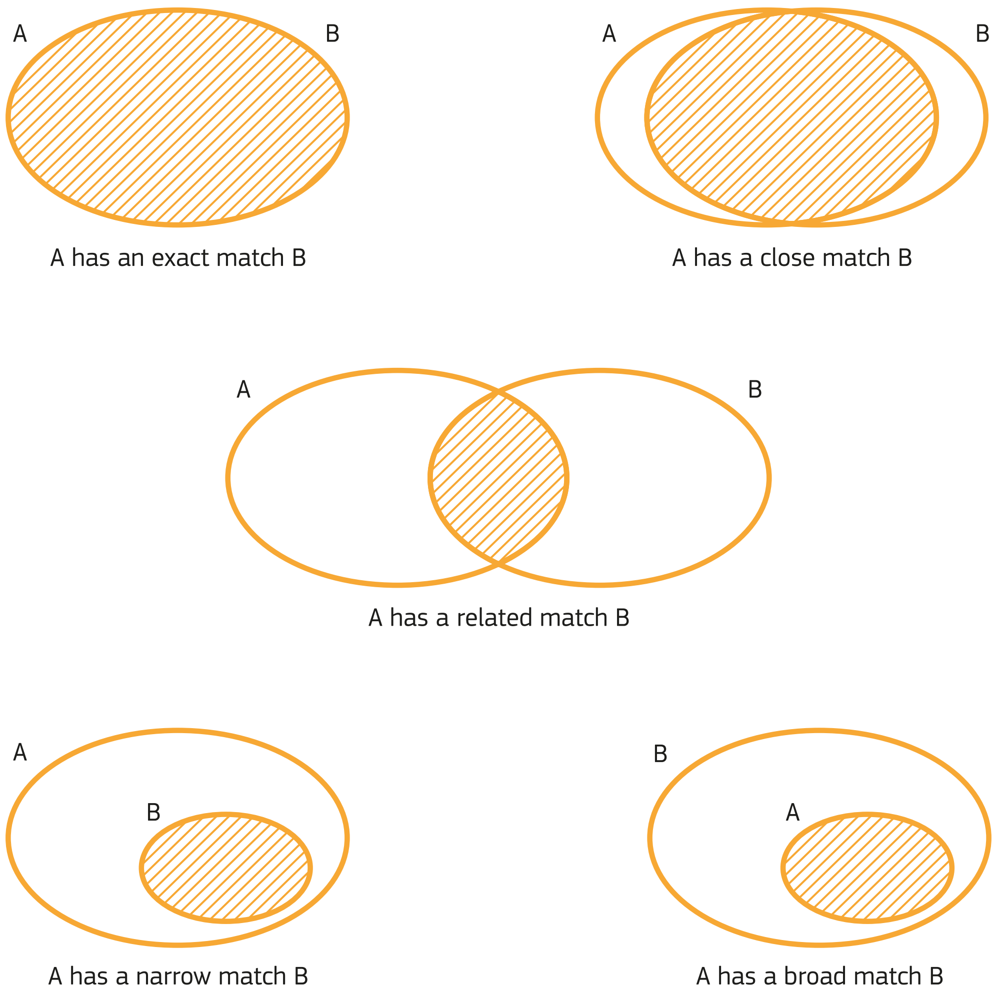

Figure 3: Mapping relations: defined in terms of the set of subjects covered by the elements [7]

3.5 Claiming conformance

For a data model to conform to the Core Vocabularies, a mapping to the conceptual model of the Core Vocabularies must be published as a Core Vocabulary self-conformance statement. This mapping shall comply with the following rules:

1. Each data element in the data model shall have a identifier, a label, and a definition:

- ***Identifier (preferably HTTP URI):*** The identifier that is used to uniquely identify an element. This should preferably be an HTTP URI that is dereferenceable.
- ***Label:*** a meaningful label that represents the meaning of the element.
- ***Definition:*** a meaningful definition that univocally defines the element.
- ***Data model:*** an identifier of the data model.

2. Each data element in the data model must have a mapping to the Core Vocabularies that contains the following information:

- ***Core Vocabulary identifier (preferably an HTTP URI):*** The identifier of the element of the Core Vocabularies. This should preferably be an HTTP URI that is dereferenceable;
- ***Mapping relation:*** mapping relation (one of "Exact match", "Close match", "Broad match", "Narrow match", "Related match", "No match"). See section 3.3. Only mapping relations defined in Section 3.3 shall be used.
- ***Identifier (preferably an HTTP URI):*** identifier of the element of the information exchange specification (empty if and only if Relation is set to "No match");
- ***Core Vocabulary Version:*** The version of the Core Vocabulary MUST be specified.
- ***Mapping comment:*** additional information about how the elements differ. A comment describing how the elements differ MUST be provided for broad, narrow, and related matches. Such comment MAY be provided for exact and close matches.

These mappings MUST be encoded in a *spreadsheet* (mandatory) and possibly as an *XML schema or RDF schema annotation* (optional), as explained in Section 4.5.

4 Design data models and create mappings using the Core Vocabularies

This chapter describes how Core Vocabularies can be used to:

- design a new data model and either bind it to an existing syntax or create a new syntax for it;
- create mappings from a data model to the Core Vocabularies' conceptual data model.

To do so, there are three options:

1. select a standard syntax that can support the defined data model; or
2. in case no suitable standard syntax is found, create a new syntax; or
3. a combination of both, binding to a standard syntax and creating new elements.

When a standard syntax is selected, it should support most of the requirements of the data model, but it also may provide for additional requirements, including data elements that are not needed. This excess of information can create confusion on eventual implementers leading to different interpretations of the syntax preventing interoperability.

When no standard syntaxes are found suitable to handle the data model, stakeholders shall create a new syntax.

In both cases, a methodology can be used as to understand how to use the Core Vocabularies. Figure 4 depicts the different steps of the methodology on the use of the Core Vocabularies:

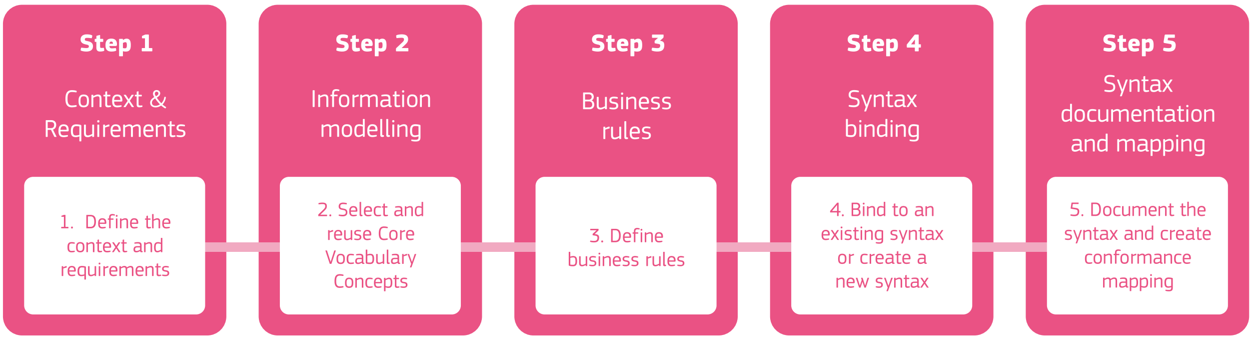

Figure 4: Methodology on using the Core Vocabularies

The sections below describe these methodological steps.

4.1 Step 1: Context and requirements

The aim of this initial step is to define the context and elicit a set of technology-neutral requirements for the data model to be designed.

- ***Define the context:*** The context of a data model can be specified in writing by stating the information exchange context and/or the particular policy domain. Other related aspects of context include the geo-political context, the administrative procedures, the applicable regulations and policies, etc. One example of context for a data model could be described as "the *cross-border exchange of electronic health records among patients and health professionals in EU Member States in the context of the Directive on the application of patients' rights in cross border health care (2011/24/EU)."* It is important to provide an accurate description of the context. For example, the term "Person" identified in the Core Vocabularies can be widely interpreted when no context is specified. On a basic level, it contains general properties that are applicable in all cases, i.e. Full Name, Date Of Birth, Gender, etc. When considering a specific context, for instance, the cross-border exchange of electronic health records among EU Member States, the underlying term "Person" needs to be specified as "Patient" in order to assure an unambiguous interpretation.
- ***Elicit information requirements:*** This can be done by using *common requirement analysis techniques*, including: describing the business process model, expressing the goals, depicting examples and/or gathering other high-level requirements.

Examples for this step are given in Annex I Example: Design a data model with bindings to an existing syntax - Step 1: Context and requirements and in ANNEX II Example: Create a data model with a new syntax - Step 1: Context and requirements.

4.2 Step 2: Information modelling

This step is used to create a conceptual data model that covers the information requirements derived from the first step. The output of this step is the conceptual data model aligned with the Core Vocabularies.

The conceptual data model shall contain the following:

- ***Information requirement identifier:*** the unique identifier for the information requirement used to link it to the high level requirements.
- ***Type of business term:*** Identifies if the information requirement corresponds to a class, a property or an association.
- ***Business term:*** The information requirement name. It has to follow the Core Vocabulary terminology when possible.
- ***Business term definition:*** The explanatory definition of the business term.
- ***Core Vocabulary identifier:*** The global and unique identifier of the Core Vocabulary concept for the business term for those business terms where a corresponding Core Vocabulary term has been found.

Once the conceptual data model is defined, use the Core Vocabularies to:

- ***Check for alignment.*** The Core Vocabularies shall be used as a pattern to build similar data elements of the new conceptual data model. For instance, if the concept of Person is needed in the conceptual data model, we shall analyse the Core Person Vocabulary semantic concept in order to check whether the Core Person attributes can be used as a pattern to fulfil the new conceptual data model requirements.
- ***Enhance names and semantics.*** The semantics provided by the Core Vocabularies shall be reused for those data elements that have an exact match as described in section 3.3.

In order to perform these tasks:

- ***Download the Core Vocabulary*** spreadsheet from the Core Vocabulary repository[8].
- ***Identify common concepts.*** Compare the concepts of the new data model with the concepts defined in the Core Vocabulary in order to find matches.
- ***Align concepts and classes*** using the Core Vocabularies as described in Section 3.3:
  - Name data elements concepts with the Core Vocabulary terms when the match is exact. For non-exact matches, this activity allows finding synonyms, refined names in the context of the conceptual data model. Synonyms can be used but the link to the corresponding Core Vocabulary term shall be maintained.
    - Align data element descriptions using the Core Vocabulary descriptions.
    - Align the data model classes to the Core Vocabulary classes. The conceptual data model classes should be a specialization of the Core Vocabulary classes by:
      - Adding new properties representing new concepts in the new context.
      - Removing properties from the Core Vocabularies that are not used in the new context.
      - Replacing properties and associations only when needed. Replacing properties or associations for narrow matches is not encouraged as it can have a negative impact on interoperability.
- ***Identify in the conceptual data model*** the link to the Core Vocabulary using the Core Vocabulary identifier to ease the syntax binding and documentation steps.

Examples for this step are given in Annex I Example: Design a data model with bindings to an existing syntax - Step 2: Information modelling and in ANNEX II Example: Create a data model with a new syntax - Step 2: Information modelling.

[8] https://joinup.ec.europa.eu/asset/core_vocabularies/description

4.3 Step 3: Business rules

The previous step has defined the information requirements. There are still action assertions, constraints and derivations concerning some aspects of the conceptual data model that have to be defined. They can be summarized as:

- Integrity constraints on the information model;
- Inferences and mathematical calculations;
- Conditional business rules and co-occurrence constraints; and
- Sets of allowed values for coded data elements.

The Core Vocabularies themselves do not define cardinalities for attributes nor business rules or additional constraints.

The outcome of the third step is an enhanced data model with the cardinalities and constraints and the lists of sets of values that restrict the possible values for coded elements.

Examples for this step are given in Annex I Example: Design a data model with bindings to an existing syntax - Step 3: Business rules and in ANNEX II Example: Create a data model with a new syntax - Step 3: Business rules.

4.4 Step 4: Bind to an existing syntax or create a new syntax

During the syntax binding process, the information requirements are bound to actual elements with a given syntax. When there is a standard syntax supporting a conceptual data model, it is recommended to maximally use the existing standard syntax. If no standard syntax is available, then a new syntax element can be created.

The process to create a syntax binding is as follows:

1. **Choose a representation format:** the information requirements can be implemented in different ways depending on the use case, for example:
   - ***XML Schema*** when creating an information exchange model or domain model.
   - ***Linked Data*** (RDF) when creating an information exchange model or domain model [8].
   - ***Data Definition Language*** (SQL) when creating a database.
2. **Choose standard syntax bindings and naming and design rules:** there are several standard syntaxes that can be used depending on the domain of the conceptual data model and the selected representation format.

   These standard syntaxes provide support for different domains such as transportation or procurement. For specific domains, other standard syntaxes can be used such as for example HL7 in the health domain or XBRL for financial reporting.

   In addition to standard syntax bindings, naming and design rules (NDRs) have to be chosen to create the actual syntax. For example, there are several naming and design rules according to which XSD Schemas can be created:

   - UBL methodology;
   - CEFACT methodology;
   - NIEM methodology;
   - ISO20022 methodology.

   In most cases, the selection of the standard syntax to be used indicates also the naming and design rules to be followed.
3. **Use existing mappings where available:** The Core Vocabularies provide guidance in the syntax binding process as they pre-define a set of mappings to existing standard syntaxes. Currently, the following standard syntaxes have mappings to the Core Vocabularies:
   - Core Vocabularies RDF Schemas;
   - OASIS Universal Business Language 2.1;
   - UN/CEFACT CCL 13B; and
   - NIEM 3.0.

These mappings are available for download on Joinup[9]. These pre-defined mappings provide a consistent way to map the same concepts to the same syntax elements cross-projects and cross-domains. Use the Core Vocabulary mappings as follows:

- Download the Core Vocabulary spreadsheet from the Core Vocabulary repository[10].
- Identify the classes, properties and associations in the conceptual data model that refer to a *Core Vocabulary identifier*.
- Select the sheet "Mappings" from the Core Vocabulary spreadsheet. This sheet has the following information:
  - ***Core Vocabulary Identifier:*** The identifier of the Core Vocabulary term.
  - ***Relation:*** The type of relation with the Core Vocabulary as described in section 3.3.
  - ***Foreign identifier:*** The identifier of the data element in the standard syntax.
  - ***Foreign source:*** The name of the standard syntax.
  - ***Comment:*** The additional comment to describe the relationship.
- Filter the sheet selecting the chosen syntax in the *Foreign source column*.
- Use the *Identifier* to find the corresponding class, property or association.
- Use the *Foreign identifier* as the mapping.

Annex I contains an example on how to use the Core Vocabulary syntax bindings to a standard syntax.

4. **Use standard syntax where available:** The information requirements that do not have a correspondence to a Core Vocabulary concept shall be mapped to the proper element in the standard syntax.

   Use the semantics of the standard syntax to identify the mapping.

5. **Mint new terms where needed:** If an information requirement cannot be bound to the standard syntax, it will be necessary to mint new terms.

   If the representation format is XML, the standard syntax should have extension points to allow adding new terms. New terms shall be used to create a new schema and this new schema shall be included into the extension point as defined by the selected standard syntax.

6. **Create a specific schema (validation artefacts):** The outcome of this step is a schema, a specification that defines the new syntax.

   - ***XML Schemas:*** Standardization Definition Organizations (SDOs) provide *validation artefacts* for their standard syntaxes, following predefined XML Naming and Design Rules (NDR). XSD Schema is the main type of validation artefact provided by SDOs. They are used to validate that a particular XML document instance fulfils the structural and type constraints defined by the standard. Using the syntax binding process, additional constraints are added on top of the standard restrictions. Additional validation artefacts shall be created to allow users to verify that instances fulfil the new data model restrictions. The validation artefacts can be created using different technologies:
     - ***Restricted XSD Schema.*** An XSD schema restricted to the elements and attributes from the standard syntax actually used for the new data model.
     - ***Schematron validation file.*** An artefact that checks for the presence of required data elements from the new model, and ensures there are no elements not belonging to the data model.

[9] https://joinup.ec.europa.eu/asset/core_vocabularies/description

[10] https://joinup.ec.europa.eu/asset/core_vocabularies/description

Annex II contains an example on how to create a new syntax using the Core Vocabulary library and the UBL methodology and its naming and design rules. Annex III lists several tools that can be used to create XML validation artefacts.

- ***RDF Schemas:*** Unlike XML, RDF Schema is intended for definition and not for validation purposes. The Cookbook for translating Data Models to RDF Schema includes a number of guidelines for creating RDF Schemas [8].
- ***Data Definition Language (SQL):*** The end-product for a relational database representation is an SQL Data Definition Language (DDL) script that can be run to create a relational database structure that meets the information requirements and chosen syntax. Organisations use their own data base engineering methodologies, using logical data model design and naming conventions.

In summary, the Core Vocabularies help the implementers provide an appropriate mapping of the core concepts, which implies improving the interoperability of the conceptual data model, and leads to a consistent use of standard syntaxes.

Examples for this step are given in Annex I Example: Design a data model with bindings to an existing syntax - Step 4: Syntax binding (using an existing syntax) and in ANNEX II Example: Create a data model with a new syntax - Step 4: Syntax binding (create a new syntax).

4.5 Step 5: Syntax documentation and mapping

The aim of this step is to create documentation of the syntax that allows users to implement it, and at the same time allows the owner to claim conformance of the data model to the Core Vocabularies as described in Section 3.5. The syntax documentation takes the form of:

- ***A mapping spreadsheet (mandatory):*** a spreadsheet documenting the syntax documentation and the mapping of the syntax to the Core Vocabularies.
- ***Schema annotations (optional):*** documentation provided as part of the validation artefacts.

Mapping spreadsheet (mandatory)

The syntax documentation must be done using the spreadsheet that can be downloaded via the following link: http://mapping.semic.eu

In this spreadsheet the sheet 'mappings' conforms to the mapping information described in in Section 3.5. It already includes a number of sample mappings. Table 2 contains a screenshot with information on the mappings. These mappings must be made publicly accessible online as *a Core Vocabulary self-conformance statement*.

Core Vocabularies internal identifier	Mapping relation	Target Vocabulary internal identifier	Data model	Target identifier	Mapping comment
Person	Has exact match	KoSIT - XÖV / NatuerlichePerson_v1.1	KoSIT - XÖV	NatuerlichePerson_v1.1	missing attributes: Ehenamen, titel, anrede, nam
PersonIdentifier	Has exact match	KoSIT - XÖV / NatuerlichePerson.identifikationsnummer_v1.1	KoSIT - XÖV	NatuerlichePerson.identifikationsnummer_v1.1	
PersonFullName	Has no match	KoSIT - XÖV /	KoSIT - XÖV		
PersonGivenName	Has exact match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.vorname	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	An adequate mapping is not possible because th
PersonFamilyName	Has close match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.familienna	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	An adequate mapping is not possible because th
PersonPatronymicName	Has broad match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.weitererNa	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	An adequate mapping is not possible because th
PersonAlternativeName	Has narrow match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.kuenstlern	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	An adequate mapping is not possible because th
PersonAlternativeName	Has narrow match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.ordensnam	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	An adequate mapping is not possible because th
PersonGender	Has exact match	KoSIT - XÖV / NatuerlichePerson.geschlecht_v1.1	KoSIT - XÖV	NatuerlichePerson.geschlecht_v1.1	
PersonBirthName	Has exact match	KoSIT - XÖV / NatuerlichePerson.nameNatuerlichePerson_v1.1.geburtsnar	KoSIT - XÖV	NatuerlichePerson.nameNatuerlichePerson	Based on the definition of the property there m
PersonDateOfBirth	Has exact match	KoSIT - XÖV / NatuerlichePerson.geburt_v1.1.datum	KoSIT - XÖV	NatuerlichePerson.geburt_v1.1.datum	Geburt_v1.1 comprises datum, zusatz and gebu
PersonDateOfDeath	Has exact match	KoSIT - XÖV / NatuerlichePerson.tod_v1.1.datum	KoSIT - XÖV	NatuerlichePerson.tod_v1.1.datum	Tod_v1.1 comprises datum, zusatz and sterbeo
PersonCountryOfBirth	Has exact match	KoSIT - XÖV / NatuerlichePerson.geburt_v1.1.geburtsort	KoSIT - XÖV	NatuerlichePerson.geburt_v1.1.geburtsort	geburtsort utilizes the Anschrift_v1.1 concept a
PersonCountryOfDeath	Has exact match	KoSIT - XÖV / NatuerlichePerson.tod_v1.1.sterbeort	KoSIT - XÖV	NatuerlichePerson.tod_v1.1.sterbeort	geburtsort utilizes the Anschrift_v1.1 concept a
PersonPlaceOfBirth	Has exact match	KoSIT - XÖV / NatuerlichePerson.geburt_v1.1.geburtsort	KoSIT - XÖV	NatuerlichePerson.geburt_v1.1.geburtsort	geburtsort utilizes the Anschrift_v1.1 concept a
PersonPlaceOfDeath	Has exact match	KoSIT - XÖV / NatuerlichePerson.tod_v1.1.sterbeort	KoSIT - XÖV	NatuerlichePerson.tod_v1.1.sterbeort	geburtsort utilizes the Anschrift_v1.1 concept a
PersonCitizenship	Has exact match	KoSIT - XÖV / NatuerlichePerson.staatsangehoerigkeit_v1.1	KoSIT - XÖV	NatuerlichePerson.staatsangehoerigkeit_v1.1	
PersonResidency	Has no match	KoSIT - XÖV /	KoSIT - XÖV		
PersonAddress	Has exact match	KoSIT - XÖV / NatuerlichePerson.anschrift_v1.1	KoSIT - XÖV	NatuerlichePerson.anschrift_v1.1	

Table 2: Screenshot of the mapping spreadsheet

Schema annotations (optional)

The technical artefacts created in Step 4 shall include Schema annotations capturing the mapping to the Core Vocabularies in order to be self-descriptive:

1. **Using XML Schema annotations:** to include annotations for type definitions within the <xsd:annotation><xsd:documentation> elements. These annotations can be included as described in Table 3.
2. **Using RDF Schema annotations:** the Cookbook for translating Data Models to RDF Schema includes a number of guidelines for annotating RDF Schemas [8]. Table 4 shows an example of RDF Schema annotations.

The XSD Schemas shall use xsd:annotation to describe the mappings to the Core Vocabularies.

The annotation documentation will be used to convey the following descriptive and mapping metadata:

- ***Identifier:*** The uniform resource identifier that is used to uniquely identify an element. This should preferably be an HTTP URI that is dereferenceable.
- ***Label:*** a meaningful label that represents the meaning of the element.
- ***Definition:*** a meaningful definition that univocally defines the element.
- ***Core Vocabulary URI:*** The global and unique uniform resource identifier that uniquely identifies the corresponding element of the Core Vocabulary, as it is defined in the Core Vocabulary specification.
- ***Core Vocabulary Version:*** The version of the Core Vocabulary specification.
- ***Mapping relation:*** The mapping relation of the annotated element to the Core Vocabulary element (see section 3.3):
  - ***Has exact match***
  - ***Has close match***
  - ***Has related match***
  - ***Has broad match***
  - ***Has narrow match***
- ***Mapping comment:*** Explanatory comment on the mapping.

```
<xsd:annotation>
    <xsd:documentation>
        <cvmap:Mapping>
            <cvmap:URI>urn:x-syntax:dataelement:RequestingPerson</cvmap:URI>
            <cvmap:Label>Requesting Person</cvmap:Label>
            <cvmap:Definition>The name of the requestor</cvmap:Definition>
            <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/Person </
cvmap:CoreVocURI>
            <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
            <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
            <cvmap:MappingComment>All requestors are natural persons.</
cvmap:MappingComment>
        </cvmap:Mapping>
    </xsd:documentation>
</xsd:annotation>
```

Table 3: Example of annotation for an XML Schema

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix adms: <http://www.w3.org/ns/adms#>.
@prefix ex: <http://example.com/>.
@prefix cvmap: <http://data.europa.eu/cv/>.

ex:RequestingPerson
rdfs:label”Requesting Person”@en;
rdfs:comment “The name of the requestor”@en;
cvmap:mapping [
             a cvmap:Mapping;
      cvmap:coreVocURI <http://data.europa.eu/core-vocabularies/Person>;
             cvmap:mappingRelation cvmap:hasNarrowMatch;
             cvmap:mappingComment “ All requestors are natural persons.”@en
```

Table 4: Example of annotation for an RDF Schema

5 References

[1] European Commission, ISA Programme, eGovernment Core Vocabularies: The SEMIC.EU approach, 2011.

[2] V. Peristeras, N. Loutas, S. K. Goudos and K. Tarabanis, “A conceptual analysis of semantic conflicts in pan-European e-government services,” Journal of Information Science, vol. 34, pp. 877-891, 2008.

[3] European Commission, ISA Programme, “Process and Methodology for Core Vocabularies,” 2011.

[4] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), “Core Components Technical Specification Version 3.0,” 2009. [Online]. Available: http://www.unece.org/cefact/codesfortrade/ccts_index.html.

[5] U.S. Department of Justice, “NIEM User Guide,” 2008.

[6] ISA Programme, European Commission, “ISA Open Metadata Licence v1.1,” [Online]. Available: https://joinup.ec.europa.eu/category/licence/isa-open-metadata-licence-v11. [Accessed 1 October 2014].

[7] W3C, “SKOS Simple Knowledge Organization System Reference,” 2009.

[8] European Commission, ISA Programme, “Cookbook for translating Data Models to RDF Schemas”, 2013. [Online]. Available: http://joinup.ec.europa.eu/node/64619.

[9] European Commission, ISA Programme, “European Interoperability Framework (EIF) for European public services,” 2010. [Online]. Available: http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf.

[10] Object Management Group (OMG), “Unified Modeling Language (UML), v2.4.1,” 2011.

6 Annexes

ANNEX I
Example: Design a data model with bindings to an existing syntax

This annex describes how to bind a conceptual data model to an existing standard syntax using the Core Vocabularies. It explains how to design an "Order" electronic document based on the Core Vocabularies binding the requirements to the UBL Order document type.

The example depicts an ordering business process where the buyer submits an electronic order document to the seller, and the seller sends back an order response accepting or rejecting the order.

The scope can be depicted using a BPMN diagram, as shown in Figure 5.

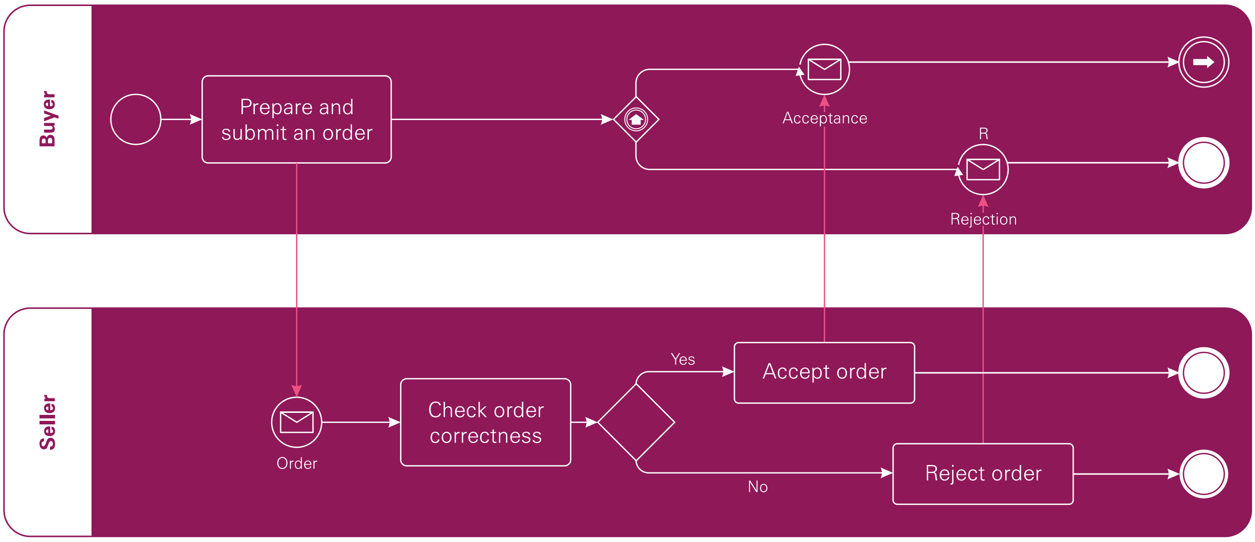

Figure 5: BPMN diagram: Ordering process

The seller drives the project. He wants to distribute a usage guideline to its community of buyers promoting an international standard syntax and a harmonized set of semantics to improve interoperability with other projects.

The rest of the annex describes the steps that have to be performed in order to create a syntax binding of the actual requirements to the UBL Order document using the Core Vocabularies to harmonize and find the correct mappings.

Step 1: Context and requirements

The seller gathers the requirements according to the goals and the scope of the project. The result of this first step is the list of business requirements.

- ***Goals:*** Table below gives an overview of the identified goals.

Goal ID	Goal Name	Goal Description
G1	Improve Business Process Performance	To simplify the ordering process for the distributors.
G2	Improve Interoperability	To harmonize the buyer business processes using international standards to reach more potential buyers.

Table 5: Example goals for the ordering project

- ***Scope:*** Jointly with the formulation of the goals and the business process definition, the scope can be explicitly expressed. The scope of the pilot is described in Table 15.

Scope statement
A distributor, using its own information system places an electronic order to the seller that receives and automatically processes it. The seller information system evaluates the order and the correctness of product references and sends back an acceptance when the order is correct. If there is an error in the order, the seller sends a rejection message to the distributor, in which case he will have to restart the ordering process again. On acceptance, the seller is bound to the delivery of the order under the conditions stated in the electronic document, and the distributor shall pay for the ordered items according to the order terms and conditions.

Table 6: Scope statement of the ordering project

- ***Specific requirements:*** The specific requirements are outlined in Table 7 below and linked to the goals.

Requirement identifier	Requirement name	Requirement statement	Rationale	Reference to goals
R1	Identification	The order has to be identified	The order shall be identifiable for future references from the despatch advice or the invoice.	G1, G2
R2	Buyer	The buyer has to be identified	The buyer is a distributor well known by the seller. Only the identifier is required.	G1, G2
R3	Customer	The final recipient of the goods.	The name of the final customer is needed to perform the delivery process.	G1
R4	Delivery location	The delivery location has to be identified through an address.	The location has to be identified to enable the delivery of the goods.	G1, G2

Table 7: High-level requirements of the ordering project (sample)

Step 2: Information modelling

The conceptual data model must be based on the high-level requirements described above and looking and comparing the conceptual data elements with the Core Vocabulary library.

Following the method explained in section 4.2 and in order to align with the Core Vocabularies, the conceptual data model must be created as follows:

- ***Identify common concepts.*** Compare the concepts of the new data model with the concepts defined in the Core Vocabulary repository[11]. The goal is to find matching concepts.

In our example, Table 8 identifies the new concepts that match the Core Vocabularies:

Requirement Identifier	Requirement name	Requirement statement	Core Vocabulary identifier
R2	Buyer	The identification of the buyer in the ordering process.	Legal Entity
R3	Customer	Information about the end customer, the final recipient of the ordered goods.	Legal Entity
R4	Delivery Address	Address where the goods have to be delivered	Address

Table 8: Matching concepts to Core Vocabularies

[11] https://joinup.ec.europa.eu/asset/core_vocabularies/description

- ***Align concepts and classes*** using the Core Vocabularies as described in Section 3.3:
  - ***Name matching.***
    - "Buyer" is a narrow match to "Legal Entity": the Buyer is "a legal entity in the role of a buyer".
    - "Customer" is a narrow match to "Legal Entity": the Customer is "a legal entity in the role of a customer".
    - "Delivery Address" is a narrow match to "Address": the Delivery Address is a specific place where the ordered items shall be delivered.
  - ***Align concept descriptions*** using the Core Vocabulary descriptions.

    The new concepts add context to the context-neutral Core Vocabulary terms, this means that the description of the Core Vocabularies is refined using the definitions of the new data elements.
  - ***Align the data model classes*** to the Core Vocabulary classes. Use the Core Vocabulary classes as a pattern to identify the properties that have to be used in the new syntax.

    Table 9 below shows the relevant classes defined by the Core Vocabularies. Based on the requirements identified in the project, the Core Vocabulary properties can be added or should not be used in the resulting data model. This is identified in the Action column.

White rows represent properties and blue rows associations to other classes. Because of the Business Information requires the Registered Address, the Core Vocabulary Address class must be also analysed. Based on these alignments the final data model shall be defined as specified in Table 9 below.

Requirement property	Core Vocabulary property	Action
Buyer	Legal Entity	
	LegalEntityLegalIdentifier	Remove
Buyer identifier	LegalEntityIdentifier	Keep
	LegalEntityLegalName	Remove
	LegalEntityAlternativeName	Remove
	LegalEntityCompanyType	Remove
	LegalEntityCompanyStatus	Remove
	LegalEntityCompanyActivity	Remove
	LegalEntityRegisteredAddress	Remove
	LegalEntityAddress	Remove
	LegalEntityLocation	Remove
Customer	Legal Entity	
	LegalEntityLegalIdentifier	Remove
	LegalEntityIdentifier	Remove
Customer name	LegalEntityLegalName	Keep
	LegalEntityAlternativeName	Remove
	LegalEntityCompanyType	Remove
	LegalEntityCompanyStatus	Remove
	LegalEntityCompanyActivity	Remove
	LegalEntityRegisteredAddress	Remove
	LegalEntityAddress	Remove
	LegalEntityLocation	Remove
Delivery address	Address	
	AddressFullAddress	Remove
	AddressPOBox	Remove

Requirement property	Core Vocabulary property	Action
Line1 and Line 2	AddressThoroughfare	Specialize
	AddressLocatorDesignator	Remove
	AddressLocatorName	Remove
City	AddressAddressArea	Specialize
	AddressPostName	Remove
Province	AddressAdminUnitL2	Specialize
	AddressAdminUnitL1	Remove
Postal code	AddressPostCode	Keep
Address ID	AddressAddressID	Keep

Table 9: Align ordering data model to the Core Vocabularies

IReqID	Type	Business Term	Definition	CoreVocID
	Class	Order	An electronic document used to place an order	
IR-001	Prop	Order identifier	Order reference for the buyer.	
IR-002	Prop	Issue date	Date of issuance of the order.	
IR-003	Prop	Requested delivery date	Latest date the order is requested to be delivered.	
IR-004	Prop	Special terms	Information about special handling requirements for the delivery of the goods.	
IR-005	Assoc	Buyer	Information about the distributor, which is the buyer.	LegalEntity
IR-006	Prop	Buyer identifier	Identifier of the buyer coded with the seller's identifier.	LegalEntityIdentifier
IR-007	Assoc	Customer	Information about the final customer, the recipient of the goods.	LegalEntity
IR-008	Prop	Customer name	Name of the customer to whom the order has to be delivered.	LegalEntityLegalName
IR-009	Assoc	Delivery Address	The address where the ordered items have to be delivered.	Address
IR-010	Prop	Delivery address identifier	Identifier of the address where the order has to be delivered.	AddressID
IR-011	Prop	Delivery address line 1	First line for the delivery address.	AddressThoroughFare
IR-012	Prop	Delivery address line 2	Second line for the delivery address.	AddressThoroughFare
IR-013	Prop	City	City name for the delivery address.	AddressPostName
IR-014	Prop	Province	Province name for the delivery address.	AddressAdminUnitL2
IR-015	Prop	Postal Zone	Postal zone for the delivery address.	AddressPostCode
IR-016	Assoc	Order line	Line of the order to specify the items and the ordered quantity.	
IR-017	Prop	Order line identifier	Reference identifier for the order line.	
IR-018	Prop	Item identifier	Seller's item identifier.	
IR-019	Prop	Item description	Textual description of the item.	
IR-020	Prop	Ordered quantity	Ordered quantity for the item.	
IR-021	Prop	Item price	Price for the item.	

Table 10: Information requirements aligned with the Core Vocabularies

Step 3: Business rules

The cardinalities of the data elements and new additional business rules that govern the behaviour of the order shall be defined in this phase.

IReqID	Type	Business Term	Definition	Card	CoreVocID
	Class	Order	An electronic document used to place an order		
IR-001	Prop	Order identifier	Order reference for the buyer.	1	
IR-002	Prop	Issue date	Date of issuance of the order.	1	
IR-003	Prop	Requested delivery date	Latest date the order is requested to be delivered.	0..1	
IR-004	Prop	Special terms	Information about special handling requirements for the delivery of the goods.	0..1	
IR-005	Assoc	Buyer	Information about the distributor, which is the buyer.	1	LegalEntity
IR-006	Prop	Buyer identifier	Identifier of the buyer coded with the seller's identifier.	1	LegalEntityIdentifier
IR-007	Assoc	Customer	Information about the final customer, the recipient of the goods.	0..1	LegalEntity
IR-008	Prop	Customer name	Name of the customer to whom the order has to be delivered.	1	LegalEntityLegalName
IR-009	Assoc	Delivery Address	The address where the ordered items have to be delivered.	1	Address
IR-010	Prop	Delivery address identifier	Identifier of the address where the order has to be delivered.	0..1	AddressID
IR-011	Prop	Delivery address line 1	First line for the delivery address.	0..1	AddressThoroughFare
IR-012	Prop	Delivery address line 2	Second line for the delivery address.	0..1	AddressThoroughFare
IR-013	Prop	City	City name for the delivery address.	0..1	AddressPostName
IR-014	Prop	Province	Province name for the delivery address.	0..1	AddressAdminUnitL2
IR-015	Prop	Postal Zone	Postal zone for the delivery address.	0..1	AddressPostCode
IR-016	Assoc	Order line	Line of the order to specify the items and the ordered quantity.	0..n	
IR-017	Prop	Order line identifier	Reference identifier for the order line.	1	
IR-018	Prop	Item identifier	Seller's item identifier.	1	
IR-019	Prop	Item description	Textual description of the item.	0..1	
IR-020	Prop	Ordered quantity	Ordered quantity for the item.	1	
IR-021	Prop	Item price	Price for the item.	0..1	

Table 11: Information requirements with cardinalities

There is an additional business rule:

Business rule ID	Business rules
BR-001	The delivery address can be specified using an identifier or the full address

Step 4: Syntax binding (using an existing syntax)

The information requirements shall be bound to the UBL syntax. The process to create a syntax binding has the following steps:

1. ***Reuse existing mappings:*** Table 12 below shows an excerpt of the Core Vocabulary to UBL mappings available in the Core Vocabulary repository[12].

The table identifies in bold the mappings for the Core Vocabularies that have to be used in the context of this project when binding to the UBL Library.

Identifier	Relation	Foreign identifier	Foreign source	Comment
Address	Exact match	Address. Details	OASIS UBL Common Library 2.1	
AddressFullAddress	No match		OASIS UBL Common Library 2.1	
AddressPOBox	Exact match	Address. Postbox. Text	OASIS UBL Common Library 2.1	
AddressThoroughfare	Broad match	Address. Street Name. Name	OASIS UBL Common Library 2.1	
AddressThoroughfare	Broad match	Address. Additional_ Street Name. Name	OASIS UBL Common Library 2.1	
AddressLocatorDesignator	Broad match	Address. Floor. Text	OASIS UBL Common Library 2.1	
AddressLocatorDesignator	Broad match	Address. Building Number. Text	OASIS UBL Common Library 2.1	
AddressLocatorName	Broad match	Address. Room. Text	OASIS UBL Common Library 2.1	
AddressLocatorName	Broad match	Address. Block Name. Name	OASIS UBL Common Library 2.1	
AddressLocatorName	Broad match	Address. Building Name. Name	OASIS UBL Common Library 2.1	
AddressAddressArea	Close match	Address. City Subdivision Name. Name	OASIS UBL Common Library 2.1	
AddressPostName	Close match	Address. City Name. Name	OASIS UBL Common Library 2.1	
AddressAdminUnitL2	Close match	Address. Country Subentity. Text	OASIS UBL Common Library 2.1	
AddressAdminUnitL1	Close match	Address. Country	OASIS UBL Common Library 2.1	
AddressPostCode	Exact match	Address. Postal_ Zone. Text	OASIS UBL Common Library 2.1	
AddressAddressID	Close match	Address. Identifier	OASIS UBL Common Library 2.1	
LegalEntity	Close match	Party Legal Entity. Details	OASIS UBL Common Library 2.1	In UBL, the Party Legal Entity ABIE (class) is used as part of the Party ABIE. Hence some properties of the Core Vocabularies' Legal Entity class can be mapped to properties of UBL's Party ABIE.
LegalEntityLegalIdentifier	Related match	Party Legal Entity. Company Identifier. Identifier	OASIS UBL Common Library 2.1	In UBL, the company identifier does not necessarily confer legal status to the legal entity.
LegalEntityIdentifier	Close match	Party Legal Entity. Company Identifier. Identifier	OASIS UBL Common Library 2.1	
LegalEntityIdentifier	Close match	Party. Party Identification	OASIS UBL Common Library 2.1	
LegalEntityLegalName	Exact match	Party Legal Entity. Registration_ Name. Name	OASIS UBL Common Library 2.1	
LegalEntityAlternativeName	Narrow match	Party. Party Name	OASIS UBL Common Library 2.1	
LegalEntityCompanyType	Close match	Party Legal Entity. Company Legal Form Code. Code	OASIS UBL Common Library 2.1	
LegalEntityCompanyStatus	Close match	Party Legal Entity. Company Liquidation Status Code. Code	OASIS UBL Common Library 2.1	
LegalEntityCompanyActivity	Exact match	Party. Industry Classification Code. Code	OASIS UBL Common Library 2.1	
LegalEntityRegisteredAddress	Exact match	Party Legal Entity. Registration_ Address. Address	OASIS UBL Common Library 2.1	
LegalEntityAddress	Broad match	Party. Postal_ Address. Address	OASIS UBL Common Library 2.1	
LegalEntityLocation	Broad match	Party. Physical_ Location. Location	OASIS UBL Common Library 2.1	

Table 12: Core Vocabulary mappings to UBL Standard syntax

[12] https://joinup.ec.europa.eu/asset/core_vocabularies/description

Using the mappings from the Core Vocabularies and the UBL Library the syntax binding for all information requirements is produced.

2. ***Map the rest of the terms to the syntax:*** Table below contains the syntax binding for the complete information requirements, including the bindings derived from the Core Vocabulary to UBL library bindings.

IReqID	Business Term	CoreVocID	Syntax Binding
	Order		ubl:Order
IR-001	Order identifier		ubl:Order/cbc:ID
IR-002	Issue date		ubl:Order/cbc:IssueDate
IR-003	Requested delivery date		ubl:Order/cac:Delivery/cac:RequestedDeliveryPeriod/cbc:EndDate
IR-004	Special terms		ubl:Order/cac:DeliveryTerms/cbc:SpecialTerms
IR-005	Buyer		ubl:Order/cac:BuyerCustomerParty
IR-006	Buyer identifier	LegalEntityIdentifier	ubl:Order/cac:BuyerCustomerParty/cac:PartyLegalEntity/cbc:CompanyID
IR-007	Customer		ubl:Order/cac:OriginatorCustomerParty
IR-008	Customer name	LegalEntityLegalName	ubl:Order/cac:OriginatorCustomerParty/ cac:PartyLegalEntity/ cac:RegistrationName/bc:Name
IR-009	Delivery Address		ubl:Order/cac:Delivery/cac:DeliveryAddress
IR-010	Delivery address identifier	AddressID	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:ID
IR-011	Delivery address line 1	AddressThoroughFare	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:StreetName
IR-012	Delivery address line 2	AddressThoroughFare	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:AdditionalStreetName
IR-013	City	AddressPostName	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:CityName
IR-014	Province	AddressAdminUnitL2	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:CountrySubentity
IR-015	Postal Zone	AddressPostCode	ubl:Order/cac:Delivery/cac:DeliveryAddress/cbc:PostalZone
IR-016	Order line		ubl:Order/cac:OrderLine/cac:LineItem
IR-017	Order line identifier		ubl:Order/cac:OrderLine/cac:LineItem/cbc:ID
IR-018	Item identifier		ubl:Order/cac:OrderLine/cac:LineItem/cac:Item/cac:SellersItemIdentification/ cbc:ID
IR-019	Item description		ubl:Order/cac:OrderLine/cac:LineItem/cac:Item/cbc:Description
IR-020	Ordered quantity		ubl:Order/cac:OrderLine/cac:LineItem/cbc:Quantity
IR-021	Item price		ubl:Order/cac:OrderLine/cac:LineItem/cac:Price/cbc:PriceAmount

Table 13: Syntax binding using Core Vocabularies

3. ***Create a specific schema (validation artefacts):*** The GEFEG.FX tool has been used to produce a Restricted XSD schema to help validating document instances and generating code according to the syntax binding.

The XSD Schema for the restricted Order is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Library:           OASIS Universal Business Language (UBL) 2.1
                     http://docs.oasis-open.org/ubl/
  Module:            xsd/maindoc/UBL-Order-2.1.xsd
  Generated on:      2014-11-01 02:42z
-->
<xs:schema xmlns="urn:oasis:names:specification:ubl:schema:xsd:Order-2"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
    xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
    targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:Order-2"
    elementFormDefault="qualified">
  <xs:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponen
ts-2" schemaLocation="Restricted=UBL-SEMIC%20Order-2.1_urn_oasis_names_specification_ubl_schema_
xsd_CommonAggregateComponents-2.xsd"/>
  <xs:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
schemaLocation="Restricted=UBL-SEMIC%20Order-2.1_urn_oasis_names_specification_ubl_schema_xsd_
CommonBasicComponents-2.xsd"/>
  <xs:element name="Order" type="OrderType">
    <xs:annotation>
      <xs:documentation>This element MUST be conveyed as the root element in any instance docu-
ment based on this Schema expression</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="OrderType">
    <xs:sequence>
      <xs:element ref="cbc:UBLVersionID" minOccurs="0"/>
      <xs:element ref="cbc:CustomizationID" minOccurs="0"/>
      <xs:element ref="cbc:ProfileID" minOccurs="0"/>
      <xs:element ref="cbc:ProfileExecutionID" minOccurs="0"/>
      <xs:element ref="cbc:ID"/>
      <xs:element ref="cbc:IssueDate"/>
      <xs:element ref="cac:BuyerCustomerParty"/>
      <xs:element ref="cac:SellerSupplierParty"/>
      <xs:element ref="cac:OriginatorCustomerParty" minOccurs="0"/>
      <xs:element ref="cac:Delivery" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="cac:DeliveryTerms" minOccurs="0"/>
      <xs:element ref="cac:OrderLine" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Step 5: Syntax documentation and mappings

The Restricted XSD Schema for the common classes contains the annotations for the mappings to the Core Vocabularies:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
  targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
  xmlns:cvmap="http://data.europa.eu/core-vocabularies/"
  elementFormDefault="qualified"  version="2.1">
  <xs:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
    schemaLocation="Restricted=UBL-SEMIC%20Order-2.1_urn_oasis_names_specification_ubl_schema_
xsd_CommonBasicComponents-2.xsd"/>
  <xs:element name="Address" type="cac:AddressType"/>
  <xs:element name="BuyerCustomerParty" type="cac:CustomerPartyType"/>
  <xs:element name="Delivery" type="cac:DeliveryType"/>
  <xs:element name="DeliveryAddress" type="cac:AddressType"/>
  <xs:element name="DeliveryTerms" type="cac:DeliveryTermsType"/>
  <xs:element name="Item" type="cac:ItemType"/>
  <xs:element name="LineItem" type="cac:LineItemType"/>
  <xs:element name="OrderLine" type="cac:OrderLineType"/>
  <xs:element name="OriginatorCustomerParty" type="cac:CustomerPartyType"/>
  <xs:element name="Party" type="cac:PartyType"/>
  <xs:element name="PartyLegalEntity" type="cac:PartyLegalEntityType"/>
  <xs:element name="Price" type="cac:PriceType"/>
  <xs:element name="RequestedDeliveryPeriod" type="cac:PeriodType"/>
  <xs:element name="SellersItemIdentification" type="cac:ItemIdentificationType"/>
  <xs:element name="SellerSupplierParty" type="cac:SupplierPartyType"/>
  <xs:complexType name="AddressType">
    <xs:annotation>
      <xs:documentation>
        <cvmap:Mapping>
          <cvmap:URI>urn:x-mylibrary:dataelement:Address</cvmap:URI>
          <cvmap:Label>Address</cvmap:Label>
          <cvmap:Definition>An address representing a location.</cvmap:Definition>
          <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/Address</cvmap:CoreVocURI>
          <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
          <cvmap:MappingRelation>exact</cvmap:MappingRelation>
        </cvmap:Mapping>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="cbc:ID" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:AddressID</cvmap:URI>
              <cvmap:Label>Address Identifier</cvmap:Label>
              <cvmap:Definition>A globally unique identifier for this instance of the
                address.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressID</
```

```
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>exact</cvmap:MappingRelation>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="cbc:StreetName" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:AddressLine1</cvmap:URI>
              <cvmap:Label>Address Line 1</cvmap:Label>
              <cvmap:Definition>First line for the delivery address.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressThorougFare</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
              <cvmap:MappingComment>A first line for address information.</cvmap:MappingComment>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="cbc:AdditionalStreetName" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:AddressLine2</cvmap:URI>
              <cvmap:Label>Address Line 2</cvmap:Label>
              <cvmap:Definition>Second line for the delivery address.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressThorougFare</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
              <cvmap:MappingComment>A second line to add address information.</
cvmap:MappingComment>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="cbc:CityName" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:City</cvmap:URI>
              <cvmap:Label>City</cvmap:Label>
              <cvmap:Definition>City name for the delivery address.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressPostName</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
              <cvmap:MappingComment>Refers specifically to the city name.</cvmap:MappingComment>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
```

```
      </xs:element>
      <xs:element ref="cbc:PostalZone" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:PostalZone</cvmap:URI>
              <cvmap:Label>Postal Zone</cvmap:Label>
              <cvmap:Definition>The post code, a.k.a. postal code, ZIP code, etc.</
cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressPostCode</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>exact</cvmap:MappingRelation>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="cbc:CountrySubentity" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:Province</cvmap:URI>
              <cvmap:Label>Province</cvmap:Label>
              <cvmap:Definition>Province for the delivery address.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/AddressAdminUnitL2</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
              <cvmap:MappingComment/>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CustomerPartyType">
    <xs:annotation>
      <xs:documentation>
        <cvmap:Mapping>
          <cvmap:URI>urn:x-mylibrary:dataelement:Customer</cvmap:URI>
          <cvmap:Label>Customer</cvmap:Label>
          <cvmap:Definition>Information about the final customer, the recipient of the
            goods.</cvmap:Definition>
          <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/LegalEntity</
cvmap:CoreVocURI>
          <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
          <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
          <cvmap:MappingComment>A legal entity in the role of a customer.</
cvmap:MappingComment>
        </cvmap:Mapping>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="cac:Party" minOccurs="0"/>
    </xs:sequence>
```

```
  </xs:complexType>
  <xs:complexType name="DeliveryTermsType">
    <xs:sequence>
      <xs:element ref="cbc:SpecialTerms" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DeliveryType">
    <xs:sequence>
      <xs:element ref="cac:DeliveryAddress" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:DeliveryAddress</cvmap:URI>
              <cvmap:Label>Delivery Address</cvmap:Label>
              <cvmap:Definition>The address where the ordered items have to be
                delivered.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/Address</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
              <cvmap:MappingComment>Delivery Address is a specific place where the ordered items
                shall be delivered</cvmap:MappingComment>
            </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref="cac:RequestedDeliveryPeriod" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ItemIdentificationType">
    <xs:sequence>
      <xs:element ref="cbc:ID"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ItemType">
    <xs:sequence>
      <xs:element ref="cbc:Description" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="cac:SellersItemIdentification" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LineItemType">
    <xs:sequence>
      <xs:element ref="cbc:ID"/>
      <xs:element ref="cbc:Quantity" minOccurs="0"/>
      <xs:element ref="cac:Price" minOccurs="0"/>
      <xs:element ref="cac:Item"> </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="OrderLineType">
    <xs:sequence>
      <xs:element ref="cac:LineItem"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="PartyLegalEntityType">
    <xs:sequence>
```

```
      <xs:element ref=”cbc:RegistrationName” minOccurs=”0”>
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:CustomerName</cvmap:URI>
              <cvmap:Label>Customer Name</cvmap:Label>
              <cvmap:Definition>The name of the customer.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/LegalEntityLegalName</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>exact</cvmap:MappingRelation> </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element ref=”cbc:CompanyID” minOccurs=”0”>
        <xs:annotation>
          <xs:documentation>
            <cvmap:Mapping>
              <cvmap:URI>urn:x-mylibrary:dataelement:BuyerIdentifier</cvmap:URI>
              <cvmap:Label>Buyer Identifier</cvmap:Label>
              <cvmap:Definition>The identifier of the buyer.</cvmap:Definition>
              <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/LegalEntityIdentifier</
cvmap:CoreVocURI>
              <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
              <cvmap:MappingRelation>exact</cvmap:MappingRelation> </cvmap:Mapping>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name=”PartyType”>
    <xs:sequence>
      <xs:element ref=”cac:PartyLegalEntity” minOccurs=”0” maxOccurs=”unbounded”/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name=”PeriodType”>
    <xs:sequence>
      <xs:element ref=”cbc:EndDate” minOccurs=”0”/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name=”PriceType”>
    <xs:sequence>
      <xs:element ref=”cbc:PriceAmount”/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name=”SupplierPartyType”>
    <xs:sequence>
      <xs:element ref=”cac:Party” minOccurs=”0”/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

ANNEX II
Example: Create a data model with a new syntax

This annex illustrates how to create a new syntax using the Core Vocabularies. The UBL methodology, the UBL Naming and Design Rules, and the Genericode to UBL NDR tool will be used. This tool is an open-source package provided by Crane Softwrights available under the Modified BSD Licence. It allows creating XSD Schemas and OASIS CVA (context/value association) files according to the UBL Naming and Design Rules.

The example is based on a use case and more elaborate pilot proposed by e-SENS[13] WP5 'Use Case 5.4 – Registering a new activity', which describes the activity registration of a business in a foreign Member State. In this document, we use the UBL methodology to generate a new e-Document using the Core Vocabularies.

The Activity Registration example allows a business (a legal entity) to expand its activities in another EU Member State and to identify the related and equivalent regulations and administrative requirements via the point of single contact (PSC) of the destination country. The business can submit a request to register a new activity for its legal entity via the point of single contact in the Member State. The central authority (CA) in the Member State is able to more easily validate the required documents submitted by the business because the documents are now submitted electronically and are digitally signed.

The scope of the example can also be depicted in a BPMN diagram, as shown in Figure 6.

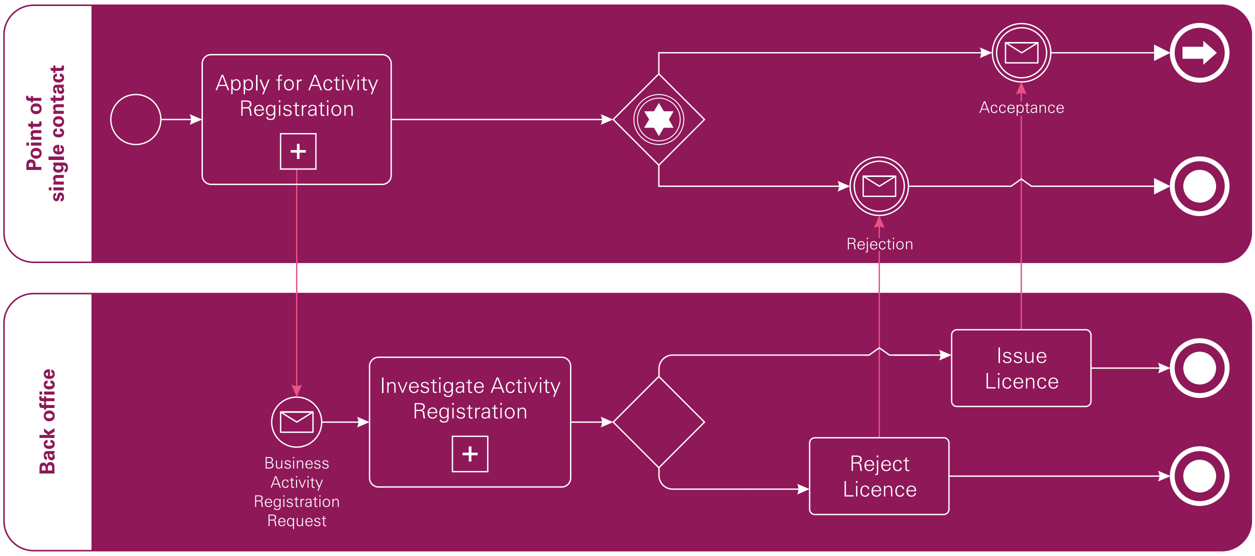

Figure 6: BPMN diagram: activity registration pilot

[13] The e-SENS project (www.esens.eu) aims at developping a digital infrastructure for improving the quality of public services in EU.

Step 1: Context and requirements

Define the objective of the business process. The requirements gathering is described according to the goals, the scope, key examples and specific requirements of the activity registration pilot. This task is derived from preliminary documents provided by the e-SENS team.

The result of this first step is a spreadsheet called “e-Document Engineering Template_ Business Activity Registration.xlsx” with the following sheets:

- ***Goals:*** Table 14 gives an overview of the identified goals in the context of this mini-pilot.

Goal ID	Goal Name	Goal Description
G1	Improve Business Process Performance	To simplify the business activity registration procedure both for the businesses and competent authorities.
G2	Improve Management Efficacy	To harmonize the business activity registration both at European level and at national level.
G3	Decrease Costs and save time	To enable competent authorities to check for validity and suitability of the information and supporting documents submitted by the businesses.
G4	Improve Security	To increase the security and reliability of the business activity registration transactions.

Table 14: Sample goals for the activity registration pilot

- ***Scope:*** The scope of the pilot is described in Table 15.

Scope statement
A business person accesses a website to retrieve information on the documents that have to be presented in a destination country (being a foreign country or their own) in order to register a business activity. The website system provides the user with information on the documents he has to upload in order to be able to submit the business activity registration request to the destination country. The process of the website system describing the documents to be submitted is out of scope. The website collects the electronic unstructured documents and metadata from the business. The website creates the e-Document with the metadata about the user, the business, the activity and the documents uploaded by the user. The website submits the e-Document instance to the destination country Point of Single Contact. The Point of Single Contact in the destination country acknowledges the business activity registration request and forwards it to the proper authority for licence issuance.

Table 15: Scope statement of the activity registration pilot

- ***Key examples:*** By means of key examples concerning activity registration given in Table 16, a real-life scenario is represented to give a description of the business process flow.

Key Example ID	Key Example Description
KE1	A business person browses the Point of Single Contact (PSC) website as a user looking for general information about his/her activity sector and legal forms required for service provision. The user selects the activity he/she needs more information about.
	The PSC offers on the home page an option to tailor the information for visitors from specific countries, and the user chooses his country of origin. Before starting the actual registration process, the user gets more detailed information regarding requirements and documents. Moreover, the user finds information on the equivalence of legal forms and supporting documents that are required from his own country.
	The user searches the proper procedures and the PSC responds with the specific procedures and requirements according to the activity that he/she has chosen and the location that he/she will offer services. Every procedure on the PSC defines the documents and requirements needed. The user is presented with a list of official documents with their equivalents in his/her home country and information from where he/she can obtain and download these documents.
	The user uses his/her identification to register on the website.
	After registering on the website, the user can save his/her list of procedures as a favourite in his/her personal area in order to begin the processing at a convenient time. In some cases an additional administrative verification process may be required.
	The user gathers all the required documents or data from his/her home country.
	The user begins the process to register the activity.
	The website responds with the types of activities that can be registered. The user selects the activity to register and the location he/she intends to offer services. The website responds with the information and documents required for the selected activity and location. The website proposes (if required) the documents that are equivalent according to the country of origin of the user. The user begins filling in the forms, creates a first draft and uploads required documents as attachments to the application. The application can be saved without being signed and submitted, which means that the user can continue with the registration process at a later time. When the application is complete, the user signs the application with his/her e-ID. The user uploads the documents to the website. The website facilitates the technical validation of the supporting documents and information e.g. digital signature and information derived from back office systems. The website creates the appropriate metadata for the uploaded documents so that they can be handled at a further stage. The website creates an envelope with all the required documents and digitally signs the documents. The website sends the validated and signed data and documents to the back office of the relevant Licensing system (e.g. Profession Association system). An electronic receipt is sent to the user either by the website or through a secure message box in his/her country. The website/or back office system provides the decision/answer of the competent authorities to the user using a secure channel.

Table 16: Key examples of activity registration (provided by e-SENS WP5)

- ***Specific requirements:*** The specific requirements related to the goals are outlined in Table 17.

Requirement identifier	Requirement name	Requirement statement	Rationale	Reference to goals
R1	Business information	The business requesting the registration of the activity has to be identified	The receiving PSC needs to know which business requests the business registration activity to be able to understand the documents it has to receive.	G1, G4
R2	Requestor	The person requesting the service on behalf of the business has to be identified	The receiving PSC has to ensure the requestor is authorized to request the service on behalf of the business.	G4
R3	Business activity	The business activity to be registered has to be identified	The receiving PSC has to know for which business activity the requester is registering for.	G1, G2
R4	Documents	The provided documents have to be identified and their purpose has to be described	The receiving PSC has to be able to identify unstructured documents to automate the registration process.	G1, G2, G3
R5	Identification	The business request has to be identified	The business request has to be uniquely identifiable, with information about its issuance.	G1, G2, G3

Table 17: High-level requirements of the activity registration pilot

Step 2: Information modelling

The semantics phase identifies and describes the information to be exchanged according to the requirements specified in Table 17.

Following the method explained in section 4.2 and in order to align with the Core Vocabularies, the information requirements shall be created as follows:

- ***Identify common concepts.*** Compare the concepts of the new data model with the concepts defined in the Core Vocabulary repository[14] in order to find matches.

In our example, the table below matches new concepts with existing concepts in the Core Vocabularies:

Requirement Identifier	Requirement name	Requirement statement	Core Vocabulary identifier	Description
R1	Business Information	The business requesting the registration of the activity has to be identified.	Legal Entity	A business that is legally registered.
R2	Requestor	The person requesting the service on behalf of the business has to be identified.	Person	A natural person.

Table 18: Matching concepts to Core Vocabularies

[14] https://joinup.ec.europa.eu/asset/core_vocabularies/description

- ***Align concepts and classes*** using the Core Vocabularies as described in Section 3.3:
  - ***Name matching.***
    - "Business Information" is an exact match to "Legal Entity"
    - "Requestor" is a narrow match to "Person": the Requestor is "a person in the role of a requestor".
  - ***Align concept descriptions*** using the Core Vocabulary descriptions.

    Definitions from the Core Vocabularies are enough to describe the concept in a context-neutral environment. In the project, these concepts can be further refined with usage explanation.
  - ***Align the data model classes*** to the Core Vocabulary classes. Use the Core Vocabulary classes as a pattern to identify the properties that have to be used in the new syntax.

    Table 19 below shows classes defined by the Core Vocabularies aligned with the requirements identified in the project. The Core Vocabulary properties can be added or omitted in the resulting data model. This is identified in the Action column.

    White rows represent properties and blue rows associations to other classes. Because of the Business Information requires the Registered Address, the Core Vocabulary Address class must be also aligned.

Requirement property	Core Vocabulary property	Action
Requestor	Person	
	PersonIdentifier	Add
	PersonFullName	Remove
	PersonGivenName	Add
	PersonFamilyName	Add
	PersonPatronymicName	Remove
	PersonAlternativeName	Remove
	PersonGender	Remove
	PersonBirthName	Remove
	PersonDateOfBirth	Remove
	PersonCountryOfBirth	Remove
	PersonCountryOfDeath	Remove
	PersonPlaceOfBirth	Remove
	PersonPlaceOfDeath	Remove
	PersonCitizenship	Remove
	PersonResidency	Remove
	PersonAddress	Remove
Business Information	Legal Entity	
	LegalEntityLegalIdentifier	Add
	LegalEntityIdentifier	Remove
	LegalEntityLegalName	Add
	LegalEntityAlternativeName	Remove
	LegalEntityCompanyType	Remove
	LegalEntityCompanyStatus	Remove
	LegalEntityCompanyActivity	Remove
	LegalEntityRegisteredAddress	Add
	LegalEntityAddress	Remove
	LegalEntityLocation	Remove

Requirement property	Core Vocabulary property	Action
	Address	
	AddressFullAddress	Add
	AddressPOBox	Remove
	AddressThoroughfare	Remove
	AddressLocatorDesignator	Remove
	AddressLocatorName	Remove
	AddressAddressArea	Remove
	AddressPostName	Remove
	AddressAdminUnitL2	Remove
	AddressAdminUnitL1	Remove
	AddressPostCode	Remove
	AddressAddressID	Remove

Table 19: Align data model to the Core Vocabularies

The information requirements shall refer to the elements inherited from the Core Vocabularies and new elements that do not exist in the Core Vocabularies.

IReqID	Type	Business Term	Definition	CoreVocID
	Doc	Activity Registration Request	An electronic document used to request an activity registration	
IR-001	Prop	Request date	The date of the request for a return authorization.	
IR-002	Prop	Request identifier	Identifier of the business activity registration request.	
IR-003	Prop	Point of Single Contact ID	Identifier of the PSC that collected and issued the business activity registration request.	
IR-004	Assoc	Requesting Person	The person requesting the registration.	Person
IR-005	Prop	PersonIdentification	A formally-issued identifier for the person.	PersonIdentification
IR-006	Prop	PersonGivenName	The denominator(s) that identify the person within a family.	PersonGivenName
IR-007	Prop	PersonFamilyName	A name that is usually shared by members of a family.	PersonFamilyName
IR-008	Assoc	Legal Entity	A business that is legally registered.	Legal Entity
IR-009	Prop	Legal Identifier	The identifier given to the legal entity by the authority with which it is registered.	LegalEntityLegalIdentifier
IR-010	Prop	Legal Name	The legal name of the business.	LegalEntityLegalName
IR-011	Assoc	Registered Address	The registered address of the business.	LegalEntityRegisteredAddress
IR-012	Prop	Full Address	The complete address with or without formatting.	AddressFullAddress
IR-013	Assoc	Document	The document provided by the requestor.	
IR-014	Prop	Document identifier	The identifier of the document.	
IR-015	Prop	Document name	The name of the document.	
IR-016	Prop	Attached document	The attached document.	

Table 20: Information data model aligned with the Core Vocabularies

Step 3: Business rules

Cardinalities of the data elements and new business rules to describe the behaviour shall be identified in this step.

IReqID	Type	Business Term	Definition	Card	CoreVocID
	Doc	Activity Registration Request	An electronic document used to request an activity registration		
IR-001	Prop	Request date	The date of the request for a return authorization.	1	
IR-002	Prop	Request identifier	Identifier of the business activity registration request.	1	
IR-003	Prop	Point of Single Contact ID	Identifier of the PSC that collected and issued the business activity registration request.	0..1	
IR-004	Assoc	Requesting Person	The person requesting the registration.	1	Person
IR-005	Prop	PersonIdentification	A formally-issued identifier for the person.	1	PersonIdentification
IR-006	Prop	PersonGivenName	The denominator(s) that identify the person within a family.	0..1	PersonGivenName
IR-007	Prop	PersonFamilyName	A name that is usually shared by members of a family.	0..1	PersonFamilyName
IR-008	Assoc	LegalEntity	A business that is legally registered.	1	LegalEntity
IR-009	Prop	LegalIdentifier	The identifier given to the legal entity by the authority with which it is registered.	1	LegalEntityLegalIdentifier
IR-010	Prop	LegalName	The legal name of the business.	1	LegalEntityLegalName
IR-011	Assoc	RegisteredAddress	The registered address of the business.	0..1	LegalEntityRegisteredAddress
IR-012		FullAddress	The complete address with or without formatting.	1	AddressFullAddress
IR-013	Assoc	Document	The documents provided by the requestor.	0..n	
IR-014	Prop	Document identifier	The identifier of the document.	1	
IR-015	Prop	Document name	The name of the document.	0..1	
IR-016	Prop	Attached document	The attached document.	1	

Table 21: Information requirements aligned with the Core Vocabularies and cardinalities

There are no additional business rules defined for this particular project.

Step 4: Syntax binding (create a new syntax)

There is no standard syntax supporting the requirements of this project, therefore, a new syntax shall be created. As described in Section 4.4 the following tasks shall be performed:

1. Choose a representation format:

In this example an e-Document will be used to represent the data model.

2. Choose a methodology:

The e-Document will be created according to the UBL methodology, and using the Genericode to UBL NDR tool.

3. Create a new library and the specific schema (validation artefacts):

In this section we describe the tasks required to create the document XSD Schema according to the UBL representation format and methodologies.

The Core Vocabularies provide a template package that allows creating a document XSD Schema using the UBL methodology. Get the CoreVoc-UBL.1.0.zip file and unzip it. It contains the following files:

- build.bat - Script to build a new set of schemas (Windows)
- build.sh- Script to build a new set of schemas (Linux)
- config-corevoc-ubl-1.0.xml – Setup file to create the new document model
- CoreVocabularies-ubl-1.0.gc – XML Genericode file with the header columns
- CoreVocabularies-ubl-1.0.ods – Spreadsheet with the information model
- CoreVocabularies-ubl-1.0.xml – XML Genericode file with the contents of the data.
- README.txt
- tool - Folder with Crane's tools to produce XSD Schemas
- xsd - Folder where that will contain XSD Schemas with annotations
- xsdrt – Folder that will contain XSD Schemas without annotations

To create an electronic document following the UBL naming and design rules, the information requirements shall follow the UBL metadata. Therefore, it is necessary to transfer the data model defined in step 2 to the template OpenOffice document provided by CoreVocabularies-ubl-1.0.ods. This template contains a sheet with the Data Library where the common classes have to be added, and a sheet with the data model of the main document model.

Each sheet in the file has the following columns:

- ***Component Name*** – The UBL Component name is derived from the Dictionary Entry Name according to the UBL Naming and Design Rules. This will be the name of the XML Tag.
- ***Dictionary Entry Name*** – Dictionary Entry Name is the unique official name of the Business Information Entity in the data dictionary. It is based on the ISO 11179.
- ***Object Class*** – Represents the logical data grouping or aggregation (in a logical data model) to which a Property belongs. Object Classes have explicit boundaries and meaning, and their Properties and behaviour follow the same rules.

  Each Object Class is an ABIE. Object classes are also referred to as Re-usable Types. In UBL, a document type is also an ABIE, and this means that the Object Class for the Business Activity Registration Request ABIE will be the same for all the properties of the document model.
- ***Property Term Qualifier*** – Property Term Qualifiers specialize or modify the Property Term. For example, when the BIE is used in another context.
- ***Property Term*** – Property Term represents the distinguishing characteristic or Property of the Object Class and "shall occur naturally in the definition." It is also known as an attribute. The combination of Object Class and its Property Term should give the basic semantic meaning of the item.
- ***Representation Term*** – Is an element of the name that describes the form in which the property is represented.

- ***Data Type*** – The data type distinguishes the lexical constraints on an item's value, plus any supplemental pieces of distinguishing information. Unqualified data types in UBL are based on UN/CEFACT ebXML CCTS core component types.
- ***Associated Object Class*** – This is the object class at the other end of the association. It is an ABIE in this model.
- ***Alternative Business Terms*** – Business Terms (optional) consists of one or more synonyms by which the Business Information Entity is commonly known and used in a specific context. A Business Information Entity may have several Business Terms or synonyms. These may be used to map BIEs to a controlled vocabulary, to other vocabularies, or to labels for forms presentation.
- ***Component Type*** – Following the CCTS there are three BIE Types:
  - Basic BIE (BBIE),
  - Associate BIE (ASBIE; "an association"), and
  - Aggregate BIE (ABIE; "an aggregate").
- ***Definition*** – This is the unique semantic business meaning of the Business Information Entity. We use the definitions described in the previous phase of the project.
- ***Cardinality*** – The cardinality of the element, defined as indicated in the information requirements model.

In order to modify the CoreVocabularies-ubl.1.0.ods file to create new information requirements the following tasks have to be done:

1. Open the CoreVocabularies-ubl.1.0.ods and rename the worksheet of the document template from "DummyDocument" to the name of the new data model. In this case, the name will be "BusinessActivityRegistrationRequest".
2. Edit the document model adding the information from the information requirements. Follow the steps below to edit the file:
   - Change the Object Class name to the new data model.
   - Maintain the UBL Version ID, Customization ID, Profile ID and Profile Execution ID in the data model to ensure the resulting XSD follows the UBL NDR.
   - Add simple properties using new white rows: copy the formulas from an existing white row and edit the following columns:
     i. Property Term Qualifier, Property Term Possessive Noun and Property Term Noun to create the new Property Term.
     ii. Representation Term from the list of possible UBL data types
     iii. Cardinality (1..1, 0..1, 1..n, 0..n)
     iv. Definition.
   - Add new associations to classes using a green row. Reuse classes from the Data library sheet just referring to the class name editing the following columns:
     i. Property Term Qualifier if any.
     ii. Associated Object Class. The name of the object class as defined in the Data Library sheet.

Component Name	Cardinality	Component Type	Definition
BusinessActivityRegistrationRequest		ABIE	A dummy document, only needed for schema generation.
UBLVersionID	0..1	BBIE	The earliest version of the UBL 2 schema for this document type that defines all of the
CustomizationID	0..1	BBIE	Identifies a user-defined customization of UBL for a specific use.
ProfileID	0..1	BBIE	Identifies a user-defined profile of the customization of UBL being used.
ProfileExecutionID	0..1	BBIE	Identifies a user-defined profile execution of the customization of UBL being used.
RequestDate	1	BBIE	The date of the request for a return authorization
RequestID	1	BBIE	Identifier of the business activity registration request
PointSingleContactID	0..1	BBIE	Identifier of the PSC that collected and issued the business activity registration request
RequestingPerson	1	ASBIE	The party requesting the registration
LegalEntity	1	ASBIE	The legal entity requested for registration
ProvidedDocumentReference	0..n	ASBIE	The documents provided by the requestor.
		END	

Figure 7: OpenOffice main document model sheet

- Add new classes into the Data Library sheet. The contents of the new class starts with a pink row and can contain properties (white rows) and associations (green rows). For the object class you have to specify:
    i. Object class name
    ii. Definition
    iii. Properties and associations following the same process as explained for the document model above.

Component Name	Cardinality	Component Type	Definition
Address		ABIE	An address representing a location.
AddressFullAddress	1	BBIE	The complete address with or without formatting.
LegalEntity		ABIE	A business that is legally registered.
LegalEntityLegalID	1	BBIE	The identifier given to the legal entity by the authority with which it is registered.
LegalEntityLegalName	1	BBIE	The legal name of the business.
RegisteredAddress	0..1	ASBIE	The registered address of the business.
Person		ABIE	A natural person.
PersonID	1	BBIE	A formally-issued identifier for the person.
PersonFamilyName	0..1	BBIE	The denominator(s) that identify the person within a family.
PersonGivenName	0..1	BBIE	A name that is usually shared by members of a family.
Document		ABIE	A reference to a document
DocumentID	1	BBIE	The identifier of the document.
DocumentName	0..1	BBIE	The name of the document.
AttachedDocumentBinaryObject	1	BBIE	The attached document.
		END	

Figure 8: OpenOffice data library document model sheet

Setup the export filter in OpenOffice

The CoreVocabularies-ubl-1.0.ods file has to be exported to a Genericode file.

Install the open-source OpenOffice spreadsheet export to Genericode subset[15] export filter that serializes the contents of the spreadsheet as a set of Genericode rows.

To install this export filter, refer to the Readme file provided with the package.

Export a Genericode Subset file from OpenOffice

Use the OpenOffice spreadsheet export to genericode subset filter from Crane Softwrights to produce a Genericode file from the CoreVocabularies-ubl-1.0.ods document. Use the “File / Export ...” function to open the export dialogue.

It is recommended to use the “.xml” extension for the exported file.

In our pilot, we have created the CoreVocabularies-ubl-1.0.xml Genericode file.

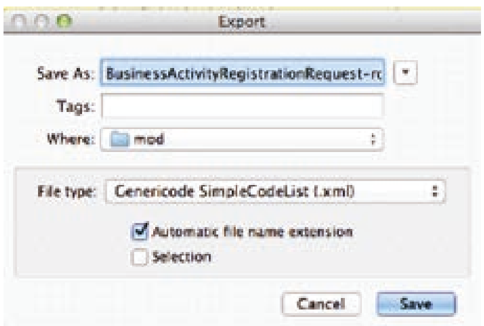

Figure 9: Export to Genericode file

15 http://www.cranesoftwrights.com/resources/ubl/Crane-gcExportSubset-20111111-2000z.zip

Setup the configuration file

There is a launch file and a configuration file.

- build.bat / build.sh – Files to launch the generation process.
- config-corevoc-ubl-1.0.xml– Setup data to generate the XSD schemas.

Launching file

The build batch uses the Java saxon9he.jar engine to build the XSD schema file using the CoreVocabulary-ubl-1.0 Genericode file and the config-corevoc-ubl-1.0.xml configuration file as inputs.

The Crane-ublndrChecker.xsl file is the XSLT file that checks that the UBL naming and design rules are properly applied into the Genericode file.

The Crane-gc2ublndr.xsl file is the XSLT file that converts the Genericode file to a XSD Schema following the UBL naming and design rules.

```
echo ISA Programme additional documents...
echo ...checking...
java -jar tool/saxon9he.jar -s:CoreVocabularies-ubl-1.0.gc -xsl:tool/Crane-ublndrChecker.xsl
-o:junk.out  configuration-uri=config-corevoc-ubl-1.0.xml common-config-uri=tool/config-ubl-2.1.xml
common-gc-uri=tool/UBL-Entities-2.1.gc
echo ...building...
java -jar tool/saxon9he.jar -s:CoreVocabularies-ubl-1.0.gc -xsl:tool/Crane-gc2ublndr.xsl
-o:junk.out  qdt-as-cva=yes configuration-uri=config-corevoc-ubl-1.0.xml common-config-uri=tool/
config-ubl-2.1.xml common-gc-uri=tool/UBL-Entities-2.1.gc aabie-prefix=dummy
```

The launching file starts checking the Genericode file provided as input and then creates the schema following the configuration. Be aware that the default alias for the document model is the one called aabie-prefix and it is set up as “dummy”. It is necessary to modify the namespace prefix. In our project, we have used the “barr” namespace prefix.

Configuration file

The configuration file is called config-corevoc-ubl-1.0.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               version="1.0">
  <!--
    This is the configuration of the schema for Core Vocabularies to UBL 1.0  -->
  <abbreviations>
    <abbreviation>CV2</abbreviation>
    <abbreviation>ID</abbreviation>
    <abbreviation>URI</abbreviation>
    <abbreviation>UNDG</abbreviation>
    <abbreviation>UBL</abbreviation>
    <abbreviation>UUID</abbreviation>
    <abbreviation>XPath</abbreviation>
  </abbreviations>
  <types>
    <type>Amount</type>
    <type>Binary Object</type>
    <type>Code</type>
    <type>Date Time</type>
    <type>Date</type>
    <type>Graphic</type>
    <type>Identifier</type>
    <type>Indicator</type>
    <type>Measure</type>
    <type>Name</type>
    <type>Numeric</type>
    <type>Percent</type>
    <type>Picture</type>
    <type>Quantity</type>
    <type>Rate</type>
    <type>Sound</type>
    <type>Text</type>
    <type>Time</type>
    <type>Value</type>
    <type>Video</type>
  </types>
  <comment>
  Library:          ISA Core Vocabularies to UBL 1.0 BETA
                    https://joinup.ec.europa.eu/community/core_vocabularies
  Module:           %f
  Generated on:     %z
</comment>
  <copyright position="end">
  Copyright (c) European Union, 2014
  Licensed under the ISA Open Metadata Licence
```

```
  ISA Open Metadata Licence v1.1

  1. Copyright

  The Work (as defined below) is provided under the terms of this ISA
  Open Metadata Licence (this Licence, or later versions of this Licence
  published by the European Commission). The work is protected by
  copyright and/or other applicable law. Any use of the work other than
  as authorised under this Licence or copyright law is prohibited.

  By exercising any rights to the work provided here, you accept and
  agree to be bound by the terms of this Licence. The Owner (as defined
  below) grants You the rights contained here in consideration of your
  acceptance of such terms and conditions.
5. Governing Law

  This licence is governed by the laws of the jurisdiction in which the
  first Contributor listed in the Copyright Notice has its principal
  place of business, unless otherwise specified by this Contributor.
</copyright>
  <type-documentation>
    <ccts:Component xmlns:ccts=”urn:un:unece:uncefact:documentation:2”>
      <ccts:ComponentType>ComponentType</ccts:ComponentType>
      <ccts:DictionaryEntryName>DictionaryEntryName</ccts:DictionaryEntryName>
      <ccts:Version>Version</ccts:Version>
      <ccts:Definition>Definition</ccts:Definition>
      <ccts:Cardinality>Cardinality</ccts:Cardinality>
      <ccts:ObjectClassQualifier>ObjectClassQualifier</ccts:ObjectClassQualifier>
      <ccts:ObjectClass>ObjectClass</ccts:ObjectClass>
      <ccts:PropertyTermQualifier>PropertyTermQualifier</ccts:PropertyTermQualifier>
      <ccts:PropertyTerm>PropertyTerm</ccts:PropertyTerm>
      <ccts:AssociatedObjectClass>AssociatedObjectClass</ccts:AssociatedObjectClass>
      <ccts:RepresentationTerm>RepresentationTerm</ccts:RepresentationTerm>
      <ccts:DataTypeQualifier>DataTypeQualifier</ccts:DataTypeQualifier>
      <ccts:DataType>DataType</ccts:DataType>
      <ccts:AlternativeBusinessTerms>AlternativeBusinessTerms</ccts:AlternativeBusinessTerms>
      <ccts:Examples>Examples</ccts:Examples>
    </ccts:Component>
  </type-documentation>
  <dir name=”xsd” runtime-name=”xsdrt”>
    <file type=”AABIE” name=”My-Document.xsd”
          abie=”MyDocument”
          prefix=”dummy” sabie-prefix=”cva” sbbie-prefix=”cvb”
          namespace=”http://example.com/”/>
    <file type=”SABIE” name=”AggregateComponents-1.0.xsd”
          prefix=”cva” namespace=”http://www.my-company.org/ns/AggregateComponents”/>
    <file type=”SBBIE” name=”BasicComponents-1.0.xsd”
          prefix=”cvb” namespace=”http://www.my-company.org/ns/BasicComponents”/>
  </dir>
</configuration>
```

The configuration file has the following sections:

1. A ***copyright*** section where the copyright statement to be added in the XSD files can be defined.
2. A ***documentation*** section with a structure following the CCTS UN/CEFACT documentation structure to create documented schemas.
3. A ***dir*** section to describe the directories, where the generated files have to be located.
4. A ***file*** section repeated per each file that has to be generated. Each file instruction has the name of the file, its type, its namespace and the namespace prefix used in the Schema. Select a new name, prefix and namespace for your new library.

This section has to be edited to define the proper file names and namespaces.

Run the Genericode-to-UBL-NDR script

The last step consists on generating the XSD Schema itself.

The script has to be launched using the launch script.

```
oriolbausa@Silver BusinessActivityRegistrationRequest$ ./build.sh
...checking...
...building...
Creating xsd/BusinessActivityRegistrationRequest.xsd ...
Creating xsd/MyLibrary-AggregateComponents-1.0.xsd ...
Creating xsd/MyLibrary-BasicComponents-1.0.xsd ...
Creating xsdrt/BusinessActivityRegistrationRequest.xsd ...
Creating xsdrt/MyLibrary-AggregateComponents-1.0.xsd ...
Creating xsdrt/MyLibrary-BasicComponents-1.0.xsd ...
```

Figure 10: Create the XSD Schema

When there are no errors in the checking phase, the script generates the XSD Schemas. In the example, the following schemas are created:

- xsd/BusinessActivityRegistrationRequest.xsd – Main document XSD Schema with annotations following the CCTS.
- xsd/MyLibrary-AggregateComponents.xsd – Library containing the project classes with comments following the CCTS.
- xsd/MyLibrary-BasicComponents.xsd – Library containing the project attributes with comments following the CCTS.
- xsdrt/BusinessActivityRegistrationRequest.xsd - Main document XSD Schema without annotations.
- xsdrt/MyLibrary-AggregateComponents.xsd - Library containing the project classes without comments following the CCTS.
- xsdrt/MyLibrary-BasicComponents.xsd - Library containing the project attributes without comments following the CCTS.

The XSD Schemas follow the UBL naming and design rules.

Step 5: Syntax documentation and mappings

The mappings from the information requirements to the Core Vocabularies library are identified in step 2. These mappings have to be transferred to the validation artefacts, in our case to the XSD Schema using the mapping annotations described in Step 4.5.

Schema annotations

The UBL XSD Schemas have the usual CCTS annotations to describe the data types and for the elements that are derived from the Core Vocabularies, there is an additional annotation to describe the mapping.

The main document XSD Schema for the Business Activity Registration Request has the annotations marked in bold below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Library:           ISA Core Vocabularies UBL 1.0 BETA
                     https://joinup.ec.europa.eu/community/core_vocabularies
  Module:            xsd/BusinessActivityRegistrationRequest.xsd
  Generated on:      2014-10-31 18:01z
-->
<xsd:schema xmlns="http://example.com/"
            xmlns:cva="http://www.myorg.org/ns/AggregateComponents"
            xmlns:cvb="http://www.myorg.org/ns/BasicComponents"          xmlns:ext="urn:oasis:n
ames:specification:ubl:schema:xsd:CommonExtensionComponents-2"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ccts="urn:un:unece:uncefact:documentation:2"
            xmlns:cvmap="http://data.europa.eu/core-vocabularies/"
            targetNamespace="http://example.com/"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            version="1.0">
   <!-- ===== Imports ===== -->
   <xsd:import namespace="http://www.myorg.org/ns/AggregateComponents"
               schemaLocation="MyLibrary-AggregateComponents-1.0.xsd"/>
   <xsd:import namespace="http://www.myorg.org/ns/BasicComponents"
               schemaLocation="MyLibrary-BasicComponents-1.0.xsd"/>
   <xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
               schemaLocation="../xsd/common/UBL-CommonBasicComponents-2.1.xsd"/>
   <xsd:import namespace="urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponen
ts-2"
               schemaLocation="../xsd/common/UBL-CommonExtensionComponents-2.1.xsd"/>
   <!-- ===== Element Declarations ===== -->
   <xsd:element name="BusinessActivityRegistrationRequest"
                type="BusinessActivityRegistrationRequestType"/>
   <!-- ===== Type Definitions ===== -->
   <!-- ===== Aggregate Business Information Entity Type Definitions ===== -->
   <xsd:complexType name="BusinessActivityRegistrationRequestType">
      <xsd:annotation>
         <xsd:documentation>
            <ccts:Component>
               <ccts:ComponentType>ABIE</ccts:ComponentType>
               <ccts:DictionaryEntryName>Business Activity Registration Request. Details</
ccts:DictionaryEntryName>
               <ccts:Definition>A dummy document, only needed for schema generation.</
ccts:Definition>
               <ccts:ObjectClass>Business Activity Registration Request</ccts:ObjectClass>
            </ccts:Component>
```

```
         </xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
           <xsd:element ref="ext:UBLExtensions" minOccurs="0" maxOccurs="1">
              <xsd:annotation>
                 <xsd:documentation>A container for all extensions present in the document.</
xsd:documentation>
              </xsd:annotation>
           </xsd:element>
          <xsd:element ref="cbc:UBLVersionID" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>BBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. UBL Ver-
sion Identifier. Identifier</ccts:DictionaryEntryName>
                     <ccts:Definition>The earliest version of the UBL 2 schema for this document
type that defines all of the elements that might be encountered in the current instance.</
ccts:Definition>
                     <ccts:Cardinality>0..1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>UBL Version Identifier</ccts:PropertyTerm>
                     <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                     <ccts:DataType>Identifier. Type</ccts:DataType>
                     <ccts:Examples>2.0.5</ccts:Examples>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cbc:CustomizationID" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>BBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Custom-
ization Identifier. Identifier</ccts:DictionaryEntryName>
                     <ccts:Definition>Identifies a user-defined customization of UBL for a specific
use.</ccts:Definition>
                     <ccts:Cardinality>0..1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>Customization Identifier</ccts:PropertyTerm>
                     <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                     <ccts:DataType>Identifier. Type</ccts:DataType>
                     <ccts:Examples>NES</ccts:Examples>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cbc:ProfileID" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>BBIE</ccts:ComponentType>
```

```
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Profile
Identifier. Identifier</ccts:DictionaryEntryName>
                     <ccts:Definition>Identifies a user-defined profile of the customization of UBL
being used.</ccts:Definition>
                     <ccts:Cardinality>0..1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>Profile Identifier</ccts:PropertyTerm>
                     <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                     <ccts:DataType>Identifier. Type</ccts:DataType>
                     <ccts:Examples>BasicProcurementProcess</ccts:Examples>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cbc:ProfileExecutionID" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>BBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Profile
Execution Identifier. Identifier</ccts:DictionaryEntryName>
                     <ccts:Definition>Identifies a user-defined profile execution of the customiza-
tion of UBL being used.</ccts:Definition>
                     <ccts:Cardinality>0..1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>Profile Execution Identifier</ccts:PropertyTerm>
                     <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                     <ccts:DataType>Identifier. Type</ccts:DataType>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cvb:RequestDate" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>BBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Request
Date. Date</ccts:DictionaryEntryName>
                     <ccts:Definition>The date of the request for a return authorization</
ccts:Definition>
                     <ccts:Cardinality>1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>Request Date</ccts:PropertyTerm>
                     <ccts:RepresentationTerm>Date</ccts:RepresentationTerm>
                     <ccts:DataType>Date. Type</ccts:DataType>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cvb:RequestID" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
```

```
                <xsd:documentation>
                    <ccts:Component>
                        <ccts:ComponentType>BBIE</ccts:ComponentType>
                        <ccts:DictionaryEntryName>Business Activity Registration Request. Request
Identifier. Identifier</ccts:DictionaryEntryName>
                        <ccts:Definition>Identifier of the business activity registration request</
ccts:Definition>
                        <ccts:Cardinality>1</ccts:Cardinality>
                        <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                        <ccts:PropertyTerm>Request Identifier</ccts:PropertyTerm>
                        <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                        <ccts:DataType>Identifier. Type</ccts:DataType>
                    </ccts:Component>
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element ref="cvb:PointSingleContactID" minOccurs="0" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    <ccts:Component>
                        <ccts:ComponentType>BBIE</ccts:ComponentType>
                        <ccts:DictionaryEntryName>Business Activity Registration Request. Point
Single Contact. Identifier</ccts:DictionaryEntryName>
                        <ccts:Definition>Identifier of the PSC that collected and issued the busi-
ness activity registration request</ccts:Definition>
                        <ccts:Cardinality>0..1</ccts:Cardinality>
                        <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                        <ccts:PropertyTerm>Point Single Contact</ccts:PropertyTerm>
                        <ccts:RepresentationTerm>Identifier</ccts:RepresentationTerm>
                        <ccts:DataType>Identifier. Type</ccts:DataType>
                    </ccts:Component>
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element ref="cva:RequestingPerson" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
                <xsd:documentation>
                    <ccts:Component>
                        <ccts:ComponentType>ASBIE</ccts:ComponentType>
                        <ccts:DictionaryEntryName>Business Activity Registration Request. Request-
ing_ Person. Person</ccts:DictionaryEntryName>
                        <ccts:Definition>The party requesting the registration</ccts:Definition>
                        <ccts:Cardinality>1</ccts:Cardinality>
                        <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                        <ccts:PropertyTermQualifier>Requesting</ccts:PropertyTermQualifier>
                        <ccts:PropertyTerm>Person</ccts:PropertyTerm>
                        <ccts:AssociatedObjectClass>Person</ccts:AssociatedObjectClass>
                        <ccts:RepresentationTerm>Person</ccts:RepresentationTerm>
                    </ccts:Component>
                    <cvmap:Mapping>
                        <cvmap:URI>urn:x-mylibrary:dataelement:RequestingPerson</cvmap:URI>
                        <cvmap:Label>Requesting Person</cvmap:Label>
```

```
                     <cvmap:Definition>The person requesting the service on behalf of the busi-
ness.</cvmap:Definition>
                     <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/Person</
cvmap:CoreVocURI>
                     <cvmap:CoreVocVersion>1.0</cvmap:CoreVocVersion>
                     <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
                     <cvmap:MappingComment>A person in the role of a requestor</
cvmap:MappingComment>
                  </cvmap:Mapping>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cva:LegalEntity" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>ASBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Legal
Entity</ccts:DictionaryEntryName>
                     <ccts:Definition>The legal entity requested for registration</
ccts:Definition>
                     <ccts:Cardinality>1</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTerm>Legal Entity</ccts:PropertyTerm>
                     <ccts:AssociatedObjectClass>Legal Entity</ccts:AssociatedObjectClass>
                     <ccts:RepresentationTerm>Legal Entity</ccts:RepresentationTerm>
                  </ccts:Component>
                  <cvmap:Mapping>
                     <cvmap:URI>urn:x-mylibrary:dataelement:LegalEntity</cvmap:URI>
                     <cvmap:Label>Requestor Name</cvmap:Label>
                     <cvmap:Definition>The name of the requestor</cvmap:Definition>
                     <cvmap:CoreVocURI>http://data.europa.eu/core-vocabularies/Person/Person-
Name</cvmap:CoreVocURI>
                     <cvmap:CoreVocVersion>1.91</cvmap:CoreVocVersion>
                     <cvmap:MappingRelation>narrow</cvmap:MappingRelation>
                     <cvmap:MappingComment>Given name of therequestor</cvmap:MappingComment>
                  </cvmap:Mapping>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element ref="cva:ProvidedDocumentReference"
                      minOccurs="0"
                      maxOccurs="unbounded">
            <xsd:annotation>
               <xsd:documentation>
                  <ccts:Component>
                     <ccts:ComponentType>ASBIE</ccts:ComponentType>
                     <ccts:DictionaryEntryName>Business Activity Registration Request. Provid-
ed_ Document Reference. Document Reference</ccts:DictionaryEntryName>
                     <ccts:Definition>The documents provided by the requestor.</ccts:Definition>
                     <ccts:Cardinality>0..n</ccts:Cardinality>
                     <ccts:ObjectClass>Business Activity Registration Request</
ccts:ObjectClass>
                     <ccts:PropertyTermQualifier>Provided</ccts:PropertyTermQualifier>
```

```
                     <ccts:PropertyTerm>Document Reference</ccts:PropertyTerm>
                     <ccts:AssociatedObjectClass>Document Reference</
ccts:AssociatedObjectClass>
                     <ccts:RepresentationTerm>Document Reference</ccts:RepresentationTerm>
                  </ccts:Component>
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

ANNEX III
Glossary

Term / Acronym	Description
Core data model	A context-neutral data model that captures the fundamental characteristics of an entity. The Core Vocabularies are at this level.
Core Vocabularies	Simplified, re-usable, and extensible data models that capture the fundamental characteristics of a data entity in a context-neutral fashion [1].
Domain model	A data model of a particular domain (e.g. the justice domain, the healthcare domain) that identifies the entities involved and their relationships.
Information exchange specification	A data model that defines and describes the structure and content of information that is exchanged in a specific information exchange context.
Interoperability	According to the ISA Decision, interoperability means the ability of disparate and diverse organisations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organisations, through the business processes they support, by means of the exchange of data between their respective ICT systems.
Interoperability solution	Interoperability solutions means common frameworks, common services and generic tools facilitating cooperation between disparate and diverse organisations, either autonomously funded and developed by the ISA/ISA2 Programme or developed in cooperation with other Union initiatives, based on identified requirements of European public administrations[16]: • A framework (strategies, specifications, methodologies, guidelines and similar approaches and documents, according to DECISION No 922/2009/EC); • A service (operational applications and infrastructures of a generic nature, which meet common user requirements across policy areas, according to DECISION No 922/2009/EC); or • A Generic tool (means reference platforms, shared and collaborative platforms, common components and similar building blocks which meet common user requirements across policy areas according to DECISION No 922/2009/EC).
ISA Programme	The Interoperability Solutions for European Public Administrations Programme of the European Union.
RDF	Resource Description Framework
Semantic interoperability	Semantic interoperability enables organisations to process information from external sources in a meaningful manner. It ensures that the precise meaning of exchanged information is understood and preserved throughout exchanges between parties [9].
UML	Unified Modelling Language [10]
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XML Schema	An XML schema is a generic term used to identify the family of grammar based XML document structure validation languages to include the more formal W3C XML Schema Definition Language, ISO 8601 Document Type Definition, or Schematron [4].

[16] ISA2 decision: http://ec.europa.eu/isa/documents/isa_2_proposal_en.pdf

LIST OF TABLES

LIST OF FIGURES

HOW TO OBTAIN EU PUBLICATIONS

Free publications:

- one copy:
  via EU Bookshop (http://bookshop.europa.eu);
- more than one copy or posters/maps:
  from the European Union's representations (http://ec.europa.eu/represent_en.htm);
  from the delegations in non-EU countries (http://eeas.europa.eu/delegations/index_en.htm);
  by contacting the Europe Direct service (http://europa.eu/europedirect/index_en.htm) or
  calling 00 800 6 7 8 9 10 11 (freephone number from anywhere in the EU) (*).
  (*) The information given is free, as are most calls (though some operators, phone boxes or hotels may charge you).

Priced publications:

- via EU Bookshop (http://bookshop.europa.eu).

Priced subscriptions:

- via one of the sales agents of the Public ations Office of the European Union
  (http://publications.europa.eu/others/agents/index_en.htm).

NO-04-15-537-EN-N

You can download our publications or subscribe for free at
http://ec.europa.eu/social/publications

If you would like to receive regular updates about the Directorate-General for Employment, Social Affairs and Inclusion, sign up to receive the free Social Europe e-newsletter at
http://ec.europa.eu/social/e-newsletter

https://www.facebook.com/socialeurope

https://twitter.com/EU_Social

Publications Office