Guidelines for public administrations on e-Document engineering methods

SC17DI06692 - D1.2 Report on Semantics for e-Documents based on the Core Vocabularies

Date: 10/06/2014

Document Metadata

Property	Value
Date	2014-05-27
Status	Accepted
Version	1.00
Authors	Oriol Bausà Peris – Invinet Natacha Dewyngaert – PwC EU Services Stijn Goedertier – PwC EU Services Vianney le Clément de Saint Marcq – PwC EU Services
Reviewed by	Pieter Breyne – PwC EU Services
Approved by	Susanne Wigard – European Commission Vassilios Peristeras – European Commission

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.

EXECUTIVE SUMMARY

This report presents the result of a study and pilot on e-Document engineering methods. It identifies generic functional requirements for e-Document engineering methods, provides an assessment of three existing standard e-Document engineering methods and gives recommendations on e-Document engineering. The report also presents the lessons learned from a mini-pilot on e-Document engineering that was carried out in collaboration with the e-SENS WP 6.2 Competence Cluster on Semantics, processes and Documents. The study is commissioned by the Interoperability Solutions for European Public Administrations Programme (ISA programme)[1] of the European Commission, in the context of its Action 2.15 on e-Documents and e-Files.

For the purpose of this study, we consider the following definitions:

An **electronic document (e-Document)** is any document in electronic format containing structured data (and possibly also unstructured data) used in the context of an administrative process.
An **e-Document engineering method** is *a discipline for specifying, designing and implementing the documents that serve as the interfaces to business processes.* [Glushko & McGrath, 2005]

Public administrations typically specify and use e-Documents in the context of *message-based application integration*. In this study, we will focus on the use of the structured part of e-Document formats used in message-based information exchanges.

The report consists of six chapters. In the first chapter, we introduce the study and describe our approach. In Chapter 2 we outline generic functional requirements that e-Document engineering methods should fulfil. In Chapter 3, three commonly used standard e-Document engineering methods are described and assessed according to several relevant criteria based on the Common Assessment Method of Standards and Specifications (CAMSS) method. The criteria allow illustrating to which extent the e-Document engineering methods could be used by public administrations. A detailed overview of the CAMSS assessment per e-Document engineering method is available in Annex III. In Chapter 4, we conduct a mini-pilot in collaboration with e-SENS WP 6.2 Competence Cluster on Semantics, processes and Documents. Resulting from the assessments and the mini-pilot, we formulate a number of recommendations for public administrations on e-Document engineering in Chapter 5. In the following paragraphs, we provide a summary for each chapter.

Chapter 1 introduces the report. We provide context to the definitions for e-Document and e-Document engineering. We also motivate that there is a need for guidance on e-Document engineering among pubic administrations. A recent survey[2] [ISA Programme, 2014] revealed that to date, public administrations spend

[1] ISA Programme, http://ec.europa.eu/isa/

[2] ISA Programme (2014) Analysis of structured e-Document formats used in Trans-European Systems.

considerable effort to assess, select, and apply existing e-Document engineering methods or to develop and apply their own method.

In Chapter 2 we list high-level and **generic functional requirements for e-Document engineering methods**. These generic requirements have been validated by e-Document engineering experts and discussed with the e-SENS Competence Cluster on Semantics, processes and Documents. These requirements are classified into seven categories:

1. Context and requirements gathering;
2. Information modelling;
3. Business rules definition;
4. Syntax binding;
5. Schema production;
6. Governance; and
7. Conformance testing.

In Chapter 3, we assess **three commonly used standard e-Document engineering methods**, i.e. UN/CEFACT, OASIS UBL TC and the CEN BII method. The selection of these three engineering methods for our study is motivated by the fact that the former two have inspired the development of the Core Vocabularies, whereas the latter method demonstrates how existing e-Document formats can be reused in context specific application *profiles*. The assessment is carried out according to the **Common Assessment Method of Standards and Specifications** (CAMSS) in the field of ICT. CAMSS consists among others of an assessment process and a set of neutral and unbiased assessment criteria. The CAMSS assessment criteria cover the following domains: applicability, maturity, openness, intellectual property rights, market support, potential and coherence. A full overview of the CAMSS assessment per e-Document engineering method is provided in Annex III Assessment e-Document engineering methods based on the CAMSS criteria, illustrating to which extent the methods could be used by public administrations.

In Chapter 4, using the requirements gathering methodology from the CEN BII method and the information modelling and XML naming and design rules from the OASIS UBL TC method, we carried out a **mini-pilot on e-Document engineering** in collaboration with e-SENS WP 6.2 Competence Cluster on Semantics, processes and Documents. The objectives of the mini-pilot were:

1. To demonstrate the use of a standard e-Document engineering method to create e-Document formats; and
2. To demonstrate how a metadata registry, a *common library of data elements* and *mappings,* can help e-Document engineering.

The mini-pilot is based on a use case and more elaborate pilot proposed by e-SENS WP 5 *'Use Case 5.4 – Registering a new business activity'*, which describes the activity registration of a business in a foreign Member State.

Chapter 5 concludes the report by formulating a number of **recommendations** based on the experience gained from the mini-pilot:

1. **Select a *standard* e-Document engineering method**, speeding up the process, enhancing documentation, lowering risk and cost, and facilitating maintenance and governance;
2. **Use *standard* libraries**, increasing interoperability and ease of development and deployment, and facilitating mash-up from different e-Documents and data models;
3. **Make e-Document formats available for reuse**, increasing cross-sector and cross-border interoperability, and gaining additional feedback from peers;
4. **Follow good practices for metadata governance and management**, ensuring the stability and durability of the e-Document formats;
5. **Explore the feasibility of operating a federated metadata registry**, increasing discoverability and reuse of data elements from standard libraries and other e-Document formats, describing mappings between libraries, and enhancing documentation and traceability of e-Document formats through links between data elements and requirements; and
6. **Use existing tools**, reducing cost and risk for errors, and increasing interoperability through uniformity and compliance with standards.

Table of Contents

List of Tables

List of Figures

1. INTRODUCTION

This report presents the result of a study and pilot on e-Document engineering methods. It provides an assessment to which extent standard e-Document engineering methods can be used by public administrations and elicits generic guidelines on e-Document engineering. The report also presents the lessons learned from a mini-pilot on e-Document engineering that was carried out in collaboration with the e-SENS WP6.2 Competence Cluster on Semantics, processes and Documents. The study is commissioned by the Interoperability Solutions for European Public Administrations Programme (ISA programme)[3] of the European Commission, in the context of its Action 2.15 on e-Documents and e-Files.

1.1. Context: e-Document engineering methods

In a previous EU study on electronic documents in the EU Member States [Graux, 2009][4] focusing specifically on the implementation of Article 8 of the Services Directive[5], an electronic document (e-Document) is defined as follows: *any document in an electronic form regardless of the specific formats or solutions used […] when completing procedures and formalities [..].* In this study, we adopt this definition, but additional require that an e-Document contains at least a portion of structured data that can be easily processed by a machine.

An **electronic document (e-Document)** is any document in electronic format containing structured data (and possibly also unstructured data) used in the context of an administrative process.

The report 'Analysis of structured e-Document formats used in Trans-European Systems' [ISA Programme, 2014][6] includes a detailed analysis of fourteen e-Document formats. Examples of *e-Documents* are requests for criminal records in the European Criminal Record Information System (ECRIS), the implementation of the European Payment Order (EPO) procedure by the e-CODEX Large-Scale Pilot, or the exchange of electronic orders and invoices following the guidelines from PEPPOL and CEN WS/BII on the use of the OASIS UBL XML Schema standard.

This study aims at formulating a number of guidelines for the ***engineering*** of e-Document formats. It focuses on the ***structured*** part of e-Document formats used in a ***message-based information exchange***. The concept '*e-Document engineering method*' can be defined as follows:

An **e-Document engineering method** is *a discipline for specifying, designing and implementing the documents that serve as the interfaces to business processes* [Glushko & McGrath, 2005].

[3] ISA Programme, http://ec.europa.eu/isa/

[4] IDABC Programme (2009). Study on electronic documents and electronic delivery. http://ec.europa.eu/idabc/en/document/7667/5644.html

[5] Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market.

[6] ISA Programme (2014). Analysis of structured e-Document formats used in Trans-European Systems. https://joinup.ec.europa.eu/node/78169

There are two different approaches to e-Document engineering[7]:

1. **Syntax binding – create a guideline to reuse an existing e-Document format**: when a standard syntax exists and is applicable, project teams analyse it to verify whether it fits their requirements. The focus is on explaining how to use the e-Document rather than on creating a new e-Document format.
2. **Document format creation – create a new e-Document format**: when no standard syntax exists it is necessary to develop a new e-Document format. In this case, the project team identifies the requirements for information and creates the new format based on these requirements.

In both cases it is necessary to identify the requirements that the e-Document has to fulfil, and to establish the exact semantics of the data being exchanged and the business rules that are to be applied. The final outcome of the process is different depending on the approach. In the first case the result is usually an explanatory document or guideline on how to use the standard syntax while in the second case there are often technical artefacts such as XML Schemas to describe the new e-Document format. When possible, it is recommended applying the syntax binding approach to on-going standardization projects, rather than creating new formats to promote interoperability.

There are several methods for creating new e-Document formats or guidelines on existing ones. Standards Development Organizations (SDOs) have their own formal e-Document engineering methods, but other organisations and project teams do not have formal processes and have difficulties implementing or reusing the methods defined in SDOs.

The components used to define an e-Document format are:

- **Methodology**: A procedure and a set of conventions for capturing requirements and formalising the process model, data model, and business rules.
- **Library of data elements**: A lexicon of business terms, data types and attributes that define the elements (based on a controlled vocabulary) that can be used in the e-Document.
- **Naming and design rules**: A grammar that governs the composition of e-Documents, covering both the syntax and the semantics.

The e-Document engineering method establishes the steps to formalize these three components.

To build them, e-Document engineering uses several tools:

- Tools to design new e-Document formats;
- Tools to create new e-Document schemas; and
- Tools to ensure conformance testing.

[7] Whilst the overall focus of this guideline is for the production of e-documents using the XML syntax these concepts apply to any e-Document format syntaxes such as EDI (EDIFACT, ANSI X12), ASN.1, JSON, etc.

1.2. The need for guidance

To date, public administrations spend considerable effort to assess, select, and apply existing e-Document engineering methods or to develop and apply their own method. A recent survey[8] [ISA Programme, 2014] that we conducted on e-Document formats used in Trans-European Systems revealed that in half of the analysed cases public administrations develop their own:

- Projects were not generally using standard e-Document engineering methods. Ad-hoc methods are used to define requirements, leading to poorly-documented requirements with little or no traceability, i.e., it is difficult to link data elements back to the underlying requirements.
- Projects were not generally using standard libraries, but create their own data elements. Hence, there is little interoperability between e-Document formats created in different projects.
- Projects tended to develop their own tools to create the e-Document schemas, incurring an increase in development cost. Different tools also result in different outputs, again decreasing interoperability.

This situation leads to a fragmentation of e-Document engineering methods, a growing divergence in available e-Document formats, and increasing semantic interoperability conflicts. Besides, every project requires a huge effort to define the way the e-Document formats will be created or customized.

What is interoperability for e-Documents?

Interoperability, within the context of European public service delivery, is 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[9]. To achieve interoperability, systems must not only use the same e-Documents formats but also use them in the same way. It is common to find non-interoperable implementations of the same e-Document standard formats due to different interpretations of their semantics. For heterogeneous applications to interoperate through e-Documents, the following components have to be agreed on:

- **Choreography**: Choreography in this context is the sequence of e-Documents being exchanged between applications.
- **Syntax**: The structure of the e-Documents being exchanged.
- **Semantics:** The meaning of the elements in the e-Documents.
- **Business rules**: The behaviour of the elements within the e-Documents, their relationships and calculations.

[8] ISA Programme (2014). Analysis of structured e-Document formats used in Trans-European Systems. https://joinup.ec.europa.eu/node/78169

[9] European Interoperability Framework, http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf

There is a need among public administrations for guidance on assessing, selecting and applying e-Document engineering methods. In the short term, the guidelines are expected to contribute to aligning e-Documents engineered by public administrations in areas where there is no legacy of existing e-Document formats. In the long run, the guidelines can help align e-Documents engineered throughout the entire European public sector.

1.3. Approach to this study

To address the above-mentioned fragmentation of e-Document engineering methods we have carried out the following activities:

1. **Elicit requirements** by public administrations for e-Document engineering methods;
2. **Assess existing e-Document engineering methods** from international standardization bodies using a relevant subset of the criteria of the Common Assessment Method of Standards and Specifications (CAMSS) [IDABC Programme, 2012];
3. **Execute a mini-pilot**: apply the guidelines to document libraries of data elements and create an e-Document format; and
4. **Formulate guidelines on e-Document engineering methods**.

1.4. Glossary

This section provides a number of common definitions used throughout the report.

Table 1 – Glossary

Term / Acronym	Description
ASiC	Associated Signature Containers
CAMSS	Common Assessment Method of Standards and Specifications
CCTS	Core Components Technical Specification
Core Components	Core components are defined as context-free semantic building block for creating clear and meaningful data models, vocabularies, and information exchange packages [UN/CEFACT, 2009].
Core Vocabularies	Simplified, re-usable, and extensible data models that capture the fundamental characteristics of a data entity in a context-neutral fashion [Interoperability solutions for European public administrations (ISA), 2011].
e-Document	Any document in electronic format containing structured data (and possibly also unstructured data) used in the context of an administrative process.
e-Document format	An e-Document format is a specification that lays down the syntax (structure) and semantics of a particular type of e-Documents.

ISO 14662	ISO/IEC 14662:2010 specifies the framework for co-ordinating the integration of existing International Standards and the development of future International Standards for the inter-working of Open-EDI Parties via Open-EDI and provides a reference for those International Standards.
ISO 20022	A standardisation approach (methodology, process, repository) to be used by financial standards initiatives.
ISO 9735	Standard for Electronic Data Interchange for Administration, Commerce and Transport.
Interoperability	According 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.
NDR	Naming and Design Rules
Public data	All the information that public bodies in the European Union produce, collect or pay for. Examples are geographical information, statistics, weather data, data from publicly funded research projects, and digitised books from libraries [European Commission, 2011].
Open data	"*A piece of data or content is open if anyone is free to use, reuse, and redistribute it — subject only, at most, to the requirement to attribute and/or share-alike*" [Open Knowledge Definition].
OASIS	OASIS (Organization for the Advancement of Structured Information Standards) is a non-profit consortium that drives the development, convergence and adoption of open standards for the global information society.
OWL	The Web Ontology Language
RDF	Resource Description Framework
SBDH	Standard Business Document Header
SKOS	Simple Knowledge Organization System – RDF Vocabulary for the representation of key reference data such as code lists, and taxonomies.
Trans European Systems	Trans-European ICT solutions contribute to the realisation of a Digital Single Market in Europe and the free movement of people, information and goods across the Member States. They are set up to support an EU policy, often – but not necessarily – as a direct consequence of new EU legislation. Examples of Trans-European ICT Solutions are the VAT Information Exchange System (VIES), the European Criminal Record Information System (ECRIS), the Emissions Trading System (ETS), the Visa Information System, the Internal Market Information System (IMI) and the Electronic Exchange of Social Security Information (ESSI).
UMM	UN/CEFACT Modelling Methodology

UML	Unified Modelling Language [Object Management Group, Inc., 2012].
UN/CEFACT	United Nations Centre for Trade Facilitation and Electronic Business
URI	Uniform Resource Identifier
URI set	A collection of reference data published using URIs, about a single concept, governed from a single source.
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 [UN/CEFACT, 2009].
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformations
W3C	World Wide Web Consortium

2. Generic functional requirements for e-Document engineering methods

This chapter lists high-level and generic functional requirements that public administrations (and other organisations) may have for e-Document engineering methods. These generic requirements have been validated by e-Document engineering experts. They have also been discussed with the e-SENS Competence Cluster on Semantics, processes and Documents. They are described according the blueprint for e-Document engineering methods that is depicted in Figure 1.

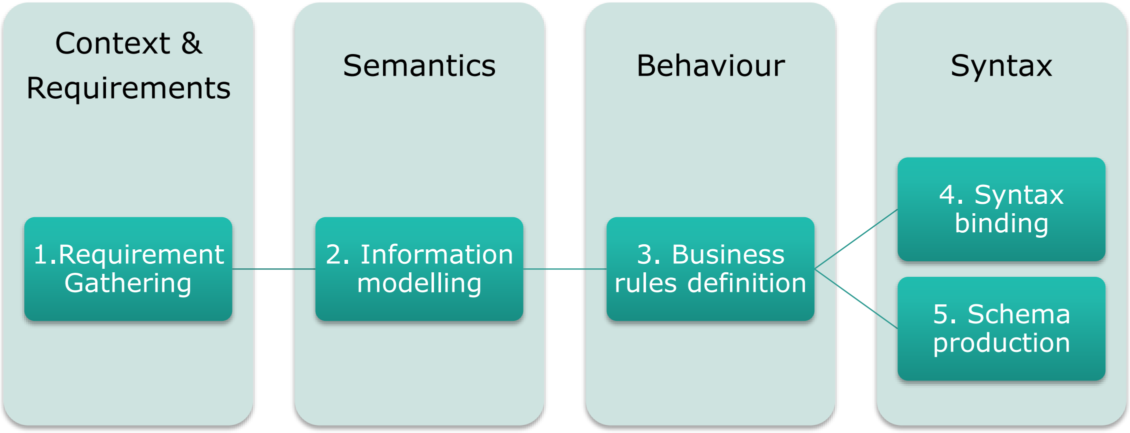


Figure 1 – Blueprint for e-Document engineering

2.1. Context and requirements gathering

2.1.1.Description

The first step of the e-Document engineering method is gathering the requirements that the e-Documents must fulfil.

The definition of e-Documents is usually done within the framework of larger and more complex projects that involve inter-organizational business processes. These will engage business process engineering activities outside the scope of these e-Document engineering guidelines. The outcome of the business process engineering activities is used to identify specific goals and requirements for these e-Documents. These goals and requirements must be explicitly stated in the context of the overall project.

The goals that have to be achieved with the exchange of e-Documents must be formulated as well as the high level requirements that help achieving them.

Identifying and documenting these goals is a key objective because they have to be the driver for the e-Document design.

A good practice to derive requirements from goals is to use examples. Describing key examples as real-life scenarios with real data helps identifying the requirements. It also forces the team members to detail new goals and requirements using a deep and pragmatic approach, rather than using an abstract and theoretical vision.

Using these goals, key examples and requirements, inter-organisational business processes can be described to establish the context of use of the e-Documents.

2.1.2.Rationale

Keeping track of the goals from the beginning of the project down to the schema production phase ensures the end results will fulfil the intended goals.

It is common finding projects where e-Documents contain data elements that do not have clear semantics because they have been added into the data model in later stages of the process, without a clear rationale, definition or linkage to the goals and requirements. This fact opens up the models to different interpretations, and it is the main cause for lack of interoperability in different implementations of the same e-Document format.

Companies and implementers not involved in the definition project are potential users of the e-Document formats. They should be able to implement the e-Document format specifications in an interoperable way, following the data models of the e-Documents, and the related documentation that has to be clear and precise.

This is a phase where the knowledge on the topic, the identification of the concepts, the scope of the project and the final outcomes must be built and shared among the project team.

2.1.3.High level requirements

The high level requirements for the requirement-gathering phase are:

Table 2 – High level requirements for gathering requirements

HLR-1	Describe specific goals to be achieved with the exchange of electronic documents.
HLR-2	Define the requirements. This can be done using techniques such as: - **Assertions**: high-level statements about the business transaction. - **Use case scenarios** using real-life examples to exemplify the business process flow; and - **Business process choreography**: a formal description of the interactions and exchange of information that takes place.
HLR-3	**Traceability**: goals and requirements should be individually identifiable and traceable. This means that requirements should be linked to goals.

2.1.4.Outcomes

The outcomes of this phase should be:

- A list of comprehensive goals described in text
- Scope documented based on the previous goals
- A set of real-life examples covering all the cases that have to be addressed using the exchange of e-Documents

- A list of textual requirements linked to the goals
- Graphical and textual description of the business process

2.2. Information modelling

2.2.1.Description

This phase must identify and describe the information to be exchanged in e-Documents according to the key examples and requirements specified in the previous step. Every data element to be exchanged has to have a unique name and a unique meaning.

This task is huge if done from scratch, and it may diverge from good practices or controlled vocabularies. These are the main reasons why it is recommended to use existing core vocabularies when possible.

In this regard, there are some standard general-purpose libraries such as the Core Component Library from UN/CEFACT, the common library of Business Information Entities from UBL or the ISA Core Vocabularies that can be used to describe the semantics of the information model that has to be built. For special industries and contexts, there are also other special libraries that can be used, such as in the health sector HL7 CDA or in the financial sector the ISO 20022 and XBRL taxonomies.

Reusing the concepts, structures and definitions of standard vocabularies promote interoperability of the e-Document formats when created.

2.2.2.Rationale

This phase is used to identify all the data elements that have to be exchanged in the e-Documents with their precise semantics. The exercise allows linking the information needs to the requirements established in the first phase. It has to be led by business experts, with deep knowledge on the scope of the project.

This task can be done from scratch, identifying each element and providing its description. Nevertheless, it is recommended to make this effort reusing existing libraries of common vocabularies that:

- Shorten the required effort
- Promote interoperability of final e-Documents

This means that before starting the definition task, there is an activity to identify possible standard syntaxes to be used or standardized data models that could fit in the information model.

Finding similar and standardized data models should help creating the e-Document information model. Using existing patterns helps not deviating from what standardization groups have done and this ends up easing the syntax binding or the syntax creation processes in the final steps of the process.

2.2.3.High level requirements

High-level requirements for this phase are:

Table 3 – High level requirements for information models

HLR-4	Capture business terms in an information model describing the explicit semantics of every data element, its attributes, cardinalities and relationships.
HLR-5	Describe relationships between information components and requirements.
HLR-6	Depict the information model requirements using conceptual modelling languages such as UML, SBVR and repositories ISO11179 (Metadata Registry).
HLR-7	Identify and reuse semantics and concepts from standardized vocabularies.

2.2.4.Outcomes

The outcomes of this phase should be:

- A set of concepts with unique semantic definition covering all the elements in the information model. Each concept or business term in the model must be uniquely identified so it will be possible to map it to existing syntaxes or to track them from the data elements up to the requirements.
- Set of objects or classes with properties and relationships between them, describing cardinalities.
- Optionally graphical representation of the information models.

2.3. Business rule definition

2.3.1.Description

The next step is the description of the business rules that affect the e-Document. A business rule is a statement that defines or constrains some aspect of the e-Document. It is used to assert the document structure or to control or influence its behaviour.

According to the SBVR[10] there are different types of business rules:

- **Business terms**, also known as the controlled vocabulary, describe how people think and talk about things. The business terms are described in the previous step.
- **Facts** relate one term with another. The facts can be documented as natural language sentences known as structural assertions or as relationships in a graphical model.
- **Action assertions** or **constraints** are statements that concern dynamic aspects of the business. It specifies constraints on the results that actions

[10] Semantics of Business Vocabulary and Business Rules: http://www.omg.org/spec/SBVR/1.0/

can produce. Constraints are described using declarative assertions, and in terms of the other atomic business rules.

  - **Integrity constraints** are assertions that always must be true.
  - **Conditions** are assertions that when true, another business rule should apply.

- **Derivations**, from certain business rules, other new business rules can be inferred or calculated. A derivation may be then either a mathematical calculation or an inference

The business terms and facts have already been described in the conceptual model. However, there are still action assertions, constraints and derivations that have to be defined for the e-Document. These business rules have to be described according to the goals and requirements.

2.3.2. Rationale

Stating the semantic of data elements (their definitions, structure, relationships and cardinalities) is not enough to describe how the e-Document format should be used. There is a need to further specify business rules that constrain the contents (the values) of data elements and their behaviour.

For example, defining the set of allowed values for a coded element, establishing mathematical operations that have to be validated for a numerical data or indicating the required value in a data element based on the contents of another (co-occurrence constraints) are things that cannot be described just with the semantics of the data elements.

2.3.3. High level requirements

High-level requirements for this phase are:

Table 4 – High level requirements for business rules

HLR-8	Identify the integrity constraints on the information model and describe them as business rules.
HLR-9	Define inferences and mathematical calculations that the e-Document elements must fulfil.
HLR-10	Define conditional business rules and co-occurrence constrains that the e-Document elements must fulfil.
HLR-11	Define sets of allowed values for coded data elements.

2.3.4. Outcomes

The outcomes of this phase should be a:

- List of business rules in plain language; and a
- List of allowed code values for specific coded fields.

2.4. Syntax binding (reuse)

2.4.1.Description

Syntax binding is one of the options to produce physical artefacts to help developers when implementing the e-Documents to ensure they will follow the e-Document format rules.

Syntax binding means mapping the required information model to an existing syntax model (generally from a commonly used standard). Instead of creating a new e-Document format, the project uses an existing one, and specifies the usage guidelines.

The actual syntax selected for the mapping should be as close as possible to the information model defined in previous stages.

Sometimes there will not be an exact equivalent for the required information in the syntax model, therefore the syntax should:

- Have extension capabilities, to allow adding new elements

Sometimes there may be additional constraints in the syntax model; therefore the syntax should:

- Avoid including unnecessary constraints in the schemas such as facets or sets of values or code lists bound to coded elements

In other cases the semantics of the data elements of the syntax are refined and overwritten by the semantics of the information model. This may sometimes be in violation to the semantics of the standardized syntax. A conformance statement can be used to enumerate and report such deviations. This is important to ensure interoperability over time and between different implementation projects reusing the same formats.

Once the mapping is done, a textual guideline has to be produced to help implementers. It is also possible to create special validation artefacts further constraining the original syntax. These artefacts can be:

- Rule oriented validation artefacts, such as **Schematron artefacts**, validating the business rules and code list restrictions; and
- **Restricted XSD Schemas** that can further restrict the original syntax.

2.4.2.Rationale

There are several advantages to re-using existing e-Document formats even though it may initially seem easier to create new e-Document formats fitting the exact need for a particular project.

Reusing standard e-Document formats (especially those that are international and open) leverages the interoperability, tools, experiences and ongoing governance of a standard, as the same format is likely to be used in other similar projects.

For example, if two different projects use the same e-Document standards as their basis, they are more likely to converge and to be interoperable.

The syntax binding process provides valuable tools, such as a guideline on how to use the standard, or extra validation tools for the additional constraints the project requires. These new tools enrich the standard; ease its implementation to the users and potentially provides feedback to the standardization community on how users understand and use their work.

2.4.3.High level requirements

High-level requirements for this phase are:

Table 5 – High level requirements for syntax binding

HLR-12	Map the information model to a standard format where this format fulfils most of the goals and requirements of the project.
HLR-13	Create a usage guideline on the syntax for implementers.
HLR-14	Create validation artefacts for business rules and code lists.
HLR-15	List minor gaps and/or requirements that cannot be fulfilled using the selected syntax.

2.4.4.Outcomes

The outcomes of this phase should be:

- Mapping from the information model to a suitable format.
- Validation artefacts including business rules and code lists.
- Guidelines on the usage of the e-Document format.

2.5. Schema production (partial reuse)

2.5.1.Description

The second option is to produce a new e-Document format. This option should be followed when there are no recognized international or open standards for the industry and business process the project is targeting.

In this case, the effort should attempt the partial reuse of existing standards libraries or vocabularies:

- The **Core Vocabularies** from the ISA Programme define the Core Person, Core Business and Core Location.
- The **OASIS UBL common library** contains all the basic and aggregated business information entities used in the UBL e-Documents, and they can be reused as they are or restricted if necessary.
- The **UN/CEFACT Core Components Library** contains a set of core component elements from where business information entities can be derived.

These libraries can be used as the building blocks to create the new e-Document format.

For the new elements identified and defined in the previous phases, a new library must be defined.

The last step for schema production is to create the schemas themselves. This step should be achieved using standard naming and design rules.

Naming and Design Rules (NDR) are the formal rules that define how to create an e-Document format, specifying how the data elements are named and structured. The most relevant NDR are:

- UBL NDR[11]
- UN/CEFACT NDR[12]
- NIEM NDR[13]

2.5.2.Rationale

Creating a new e-Document format can be done in many different ways. When there is the need to create such a new e-Document format, it is a good practice to produce it following standard patterns described in formal Naming and Design Rules. This allows common structures and naming of elements and data types in e-Document formats, which leads to harmonized treatment of e-Documents, easing the learning curve for implementers.

Besides the use of a common method to build the schemas, the use of common vocabularies has also several benefits:

- Alignment with standard vocabularies
- Reuse of descriptions and concepts
- Eventual submission of the new e-Document format to the standardization body for adoption
- Ease implementations due to reuse

2.5.3.High level requirements

High-level requirements for this phase are:

Table 6 – High level requirements for syntax binding

HLR-16	Map common information model components to available Common Vocabulary schemas.
HLR-17	Create new syntax e-Document formats using a standard set of Naming and Design Rules (such as for example UBL or UN/CEFACT) to automate the schema production.
HLR-18	Create validation artefacts for business rules and code lists.

[11] UBL NDR: http://docs.oasis-open.org/ubl/cs01-UBL-2.0-NDR/cs01-UBL-2.0-NDR.pdf
[12] UN/CEFACT NDR: http://www.unece.org/fileadmin/DAM/cefact/xml/UNCEFACT+XML+NDR+V3p0.pdf
[13] NIEM NDR: https://www.niem.gov/documentsdb/Documents/Technical/NIEM-NDR-1-3.pdf

2.5.4. Outcomes

The outcomes of this phase should be:

- XML schema for the new e-Document format.
- Validation artefacts including business rules and code lists.
- Guidelines on the usage of the format.

2.6. Governance

2.6.1. Description

Usually, creating an XML schema and a usage guideline is enough for a user community to start adopting an e-Document format.

However, when the e-Document format is to be used in a broad community, there is a need for governing the evolution and maintenance of the e-Document format. In effect the format becomes a standard.

New requirements and special use cases may arise or supporting standards can evolve, and this may be the trigger to make changes in the e-Document format.

The Governance of an e-Document format has to take among others into account:

- Which organization will take responsibility for the e-Document format;
- What process will be defined to deal with new requirements and requests; from users
- How the e-Document formats, including validation artefacts will be versioned;
- Ownership, IPR and access rights; and
- Conformance criteria.

Deliverable 'D4.2 Methodology and tools for Metadata Governance and Management for EU Institutions and Member States'[14] contains more elaborate requirements and specifications for the governance and management of structural metadata, including e-Document formats.

2.6.2. Rationale

The implementation of a e-Document format standard is costly. Implementers need to know their investment will be supported by a proper governance mechanism to do the effort.

Setting up a governance model and formalizing procedures when developing the e-Document format provides for long-term security of users' investments and increases confidence for the potential users.

[14] ISA Programme (2014). D4.2 Methodology and tools for Metadata Governance and Management for EU Institutions and Member States.

2.6.3. High level requirements

High-level requirements for this phase are:

Table 7 – High level requirements for governance

HLR-19	Set up a governance model for long-term sustainability of the e-Document format(s).
HLR-20	Define the maintenance and versioning mechanisms for the e-Document format.
HLR-21	Establish policies and procedures for the use and maintenance of the e-Document format.

2.6.4. Outcomes

The outcomes of this phase should be:

- Governance model.
- Policies and procedures for maintenance and use.

2.7. Conformance testing

2.7.1. Description

***"Interoperability,** within the context of European public service delivery, is 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"* [ISA, 2011].

To achieve semantic interoperability between inter-organizational business processes, they must:

- **Use the same information models**; and
- **Use business rules in the same way.**

They have to share the same understanding on the concepts and comply with the same rules in the components.

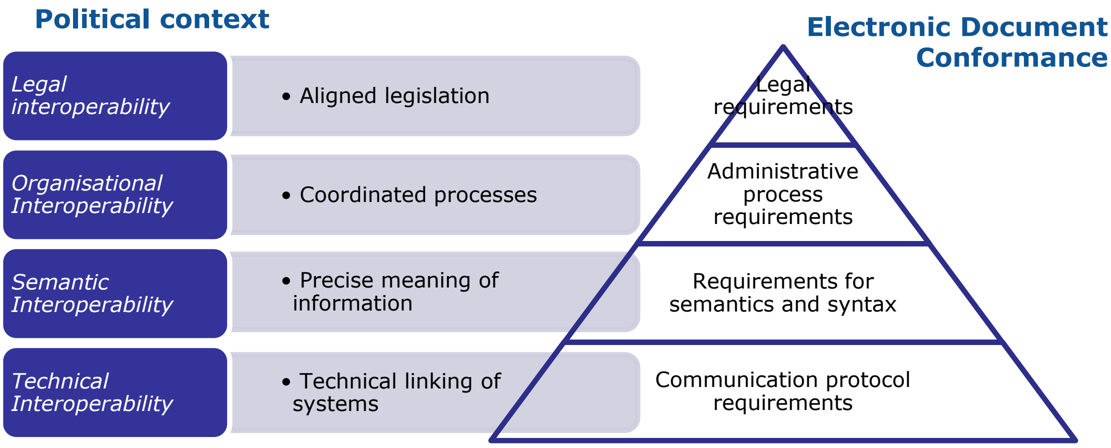

Political context
Electronic Document Conformance
Legal interoperability
• Aligned legislation
Organisational Interoperability
• Coordinated processes
Semantic Interoperability
• Precise meaning of information
Technical Interoperability
• Technical linking of systems
Legal requirements
Administrative process requirements
Requirements for semantics and syntax
Communication protocol requirements

Figure 2 - European Interoperability Framework

If we only focus on the formats, the same documents can be used in different projects, however, this does not guarantee interoperability. We need to provide an additional layer to the format to guarantee that the e-Documents are used in the same way. This additional layer includes **business rules** and **code lists**.

This means that besides technology such as XSD Schema, other tools should be provided in order to ensure that the e-Document instances follow these business rules and code lists.

2.7.2. Rationale

With EDI technology Message Implementation Guidelines have been used to describe how the e-Document formats should be implemented. The experience has demonstrated that, even with such textual guidelines there are deviations due to different interpretations of the same text.

Providing additional software tools in a test-bed environment can help implementers to ensure the instances they are producing are fully compliant with the specification of the e-Document format and any usage guidelines.

Establishing a testing framework is a good complement to the usage guidelines that can provide additional support for the end users.

2.7.3.High level requirements

High-level requirements for this phase are:

Table 8 – High level requirements for conformance testing

HLR-22	Define a testing framework for the e-Document instances.

2.7.4.Outcomes

The outcomes of this phase should be:

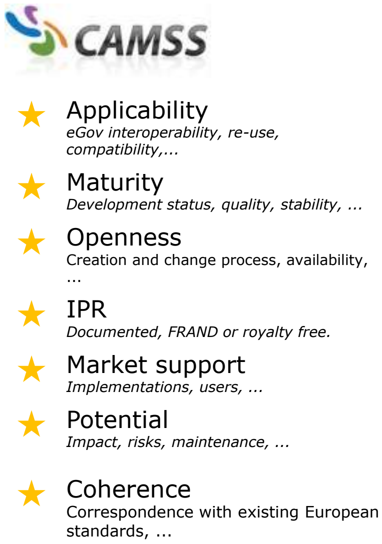
- Conformance testing procedure.

3. Assessment of standard e-Document engineering methods

In this chapter we give a summary of the assessment of three commonly used, standard e-Document engineering methods:

1. UN/CEFACT e-Document engineering method;
2. OASIS UBL method; and
3. CEN BII e-Document engineering method, a method to bind information requirements to existing syntaxes.

The selection of these three engineering methods for our study is motivated by the fact that the former two have inspired the development of Core Vocabularies[15], whereas the latter method demonstrates how existing e-Document formats can be reused in context specific application *profiles*. Many more exist, such as those of eXtensible Business Reporting Language (XBRL), Health Level 7 (HL7), and ISO 20022.



★ Applicability
eGov interoperability, re-use, compatibility,...

★ Maturity
Development status, quality, stability, ...

★ Openness
Creation and change process, availability, ...

★ IPR
Documented, FRAND or royalty free.

★ Market support
Implementations, users, ...

★ Potential
Impact, risks, maintenance, ...

★ Coherence
Correspondence with existing European standards, ...

Figure 3 – Categories of assessment criteria in CAMSS

We assess these methods according to the criteria of the **Common Assessment Method of Standards and Specifications** (CAMSS). CAMSS is a method for the assessment of technical specifications and standards in the field of ICT. It consists among others of an assessment process and a set of neutral and unbiased assessment criteria [IDABC Programme, 2012][16]. Figure 3 gives an overview of the seven categories of assessment criteria in the CAMSS framework. Annex II gives a detailed listing of all 51 assessment criteria. In Annex III, an overview of the CAMSS assessments is presented for each e-Document engineering method.

[15] The XML Schema of the Core Vocabularies follows the UBL XML Naming and Design Rules. The data elements of the Core Vocabularies are inspired by the UN/CEFACT Core Component library. https://joinup.ec.europa.eu/community/semic/og_page/studies#core-vocabularies

[16] Further information on CAMSS can be found in the dedicated CAMSS community on Joinup: https://joinup.ec.europa.eu/community/camss/description

Once all applicable criteria are assessed, an overall score of the CAMSS assessment strength is calculated for the e-Document engineering method. Each category of criteria is evaluated against an automated score. The automated score (%) per category is generated based on the level of positive answers meeting the applicable criteria for the e-Document engineering method. The CAMSS assessment strength per category is calculated based on the applicability of the criteria for the e-Document methods.

Table 9 gives an overview of the three standard e-Document engineering methods and their respective overall automated score derived from the CAMSS assessment. The overall automated scores show that the CEN BII method scores lower than the methods of UN/CEFACT and OASIS UBL. However, this can be explained by the fact that UN/CEFACT and OASIS UBL are both methods that have been established for a long time and that originate from standardization organisations. CEN BII, on the other hand, is still a CEN Workshop Agreement (CWA) in the process of becoming a CEN EN standard.

Note that it is not meaningful to give a ranking to e-Document engineering based on their assessment scores only. Assessments are based on available information. Missing information has a negative impact on the score. Only an analysis of the detailed evidence for each assessment criterion allows making an informed comparison. The complete assessments are included in Annex III.

Table 9 – Overall CAMSS automated assessment score

Category	UN/CEFACT method	UBL method	CEN BII method	Assessment Strength
Applicability	100%	100%	100%	**88%**
Maturity	86%	86%	67%	**100%**
Openness	100%	100%	89%	**100%**
Intellectual property rights	100%	100%	100%	**100%**
Market support	75%	100%	50%	**80%**
Potential	100%	100%	50%	**62%**
Coherence	50%	33%	25%	**67%**
Overall score:	87%	88%	69%	**85%**

The assessment scores are calculated as follows:

- **Automated Score =** [#Yes/(#Yes + #No)], and
- **Assessment Strength =** [(#Yes + #No)/#Criteria per category].

The assessment strength is an indicator of the number of criteria for which an assessment is performed. The overall assessment strength of 80% indicates that 80% of the assessment criteria have been found relevant and have been filled in.

3.1. UN/CEFACT e-Document engineering method

This section contains a description and assessment of the e-Document engineering method of UN/CEFACT according to the assessment categories in CAMSS.

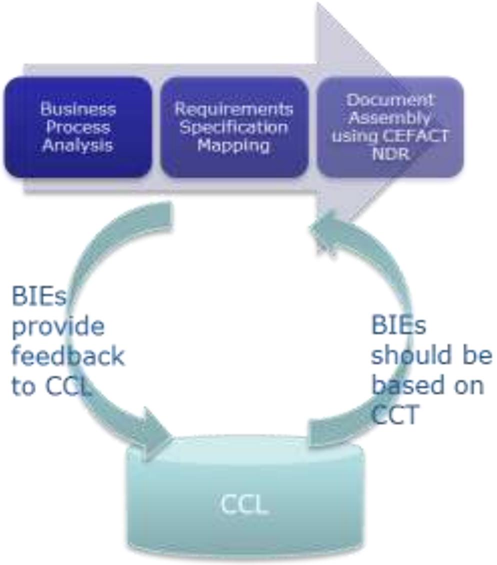

Business Process Analysis
Requirements Specification Mapping
Document Assembly using CEFACT NDR
BIEs provide feedback to CCL
BIEs should be based on CCT
CCL

Figure 4 – UN/CEFACT Methodology

3.1.1.Description

Originally approved in 1996 and part of the United Nations Economic Commission for Europe (UNECE), the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) aims to facilitate the exchange of recommendations and electronic business standards. Revisions concerning the organization were last approved in 2006.

UN/CEFACT's e-Document engineering method is not a single standard, but consists of an ensemble of standards:

- **Methodology**: UN/CEFACT has four integrated standards:
    - ○ **UN/CEFACT's Modelling Methodology (UMM)** describes the business processes required for the collaboration and implementation of services in a service-oriented architecture. The objective of the UML modelling approach is to create a global choreography of inter-organizational business processes along with their information exchanges. The resulting UMM models are based on the UML syntax and are platform independent [UN/CEFACT, 2011]. The UN/CEFACT Modelling Methodology[17] (UMM) is based on OMG UML for modelling the exchange of information between businesses [UN/CEFACT, 2009].

[17] The UN/CEFACT Modelling Methodology (UMM): http://www.unece.org/cefact/umm/umm_index.htm

  - **The UML Profile for Core Components (UPCC)** allows the conceptual models of CCTS and UMM to operate as a UML meta-model [UN/CEFACT, 2008].
  - **The Core Component Technical Specification (CCTS)** contains meta-models and rules concerning the structure and contents of data models and **information** exchange models. Its purpose is to enhance information interoperability by maximizing the reuse of business information. CCTS is composed of two main concepts: Core Components (CC) and Business Information Entities (BIE). Core Components are building blocks covering data modelling, information modelling and information exchange. Core Components are conceptual models that define the Business Information Entities. The Business Information Entities create logical data models, interoperable business process models, business documents and information exchanges [UN/CEFACT, 2009].
  - **The Core Component Business Document Assembly (CCBDA) technical specification** outlines the structure of a business document created from Core Components that is based on a Business Requirements Specification (BRS) and a Requirements Specification Mapping (RSM). CCBDA is constructed on UMM [UN/CEFACT, 2012].

- **Library of data elements**: The **Core Component Library (CCL)** contains Core Components (CC), reusable building blocks for the modelling and exchange of data and information. Business Information Entities (BIEs) are derived from Core Components and are applicable in a specific context.
- **e-Document engineering tools**: the GEFEG.FX tool is a software used to develop XML Schemas.
- **Representation techniques**: e-Documents formats from UN/CEFACT are represented using XSD Schema.
- **XML Naming and Design Rules**: The UN/CEFACT XML Naming and Design Rules[18] describe the architecture and the rules to outline XML Schemas [UN/CEFACT, 2009].

In the remaining subsections, we will describe the fit of the UN/CEFACT method with the high-level requirements that are elicited in Chapter 2. These requirements are structured according to the following steps:

1. Requirement gathering;
2. Information modelling;
3. Business rule definition;
4. Syntax binding;
5. Schema production;
6. Governance; and

[18] The UN/CEFACT XML Naming and Design Rules: http://www.unece.org/cefact/xml/xml_index.html

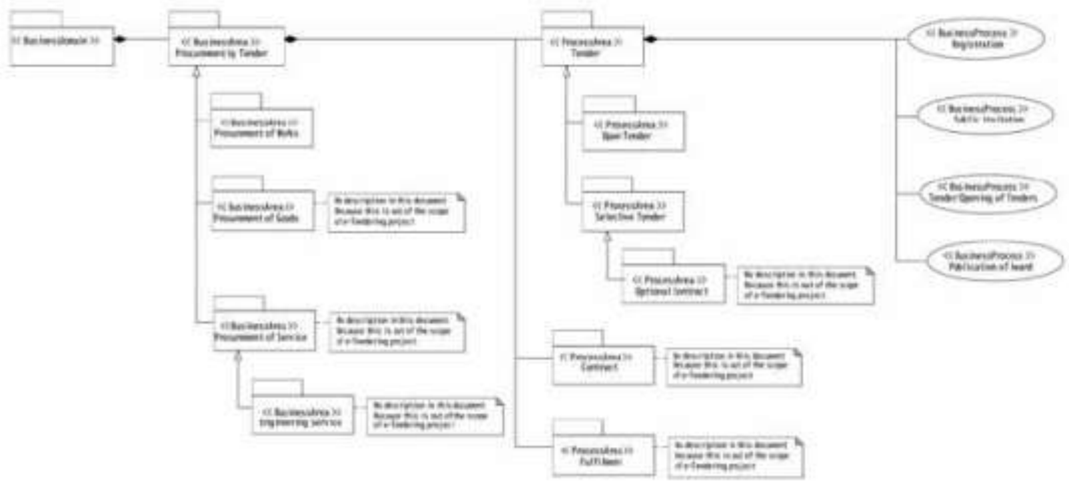
7. Conformance testing.

3.1.1.1. Requirement gathering

UN/CEFACT uses the Business Requirement Specifications (BRS) to describe the business processes. The BRS have a formal layout to describe the way the business processes have to be performed. Each business process is captured using a top-down sequence of UML diagrams.

The BRS uses the following sections:

- Business Operation Map – high level view of the business processes covered by the BRS



<< BusinessDomain >>
<< BusinessArea >> Procurement by Tender
<< BusinessArea >> Procurement of Works
<< BusinessArea >> Procurement of Goods
No description in this document because this is out of the scope of e-Tendering project
<< BusinessArea >> Procurement of Service
No description in this document because this is out of the scope of e-Tendering project
<< BusinessArea >> Engineering Service
No description in this document because this is out of the scope of e-Tendering project
<< ProcessArea >> Tender
<< ProcessArea >> Open Tender
<< ProcessArea >> Selective Tender
<< ProcessArea >> Optional Contract
No description in this document because this is out of the scope of e-Tendering project
<< ProcessArea >> Contract
No description in this document because this is out of the scope of e-Tendering project
<< ProcessArea >> Fulfilment
No description in this document because this is out of the scope of e-Tendering project
<< BusinessProcess >> Registration
<< BusinessProcess >> Public Invitation
<< BusinessProcess >> Tender Opening of Tenders
<< BusinessProcess >> Publication of Award

Figure 5 – UN/CEFACT Business Operation Map

- Business Domain View – description of the process within the domain it has to be developed
- Business Area Model – description of the first breakdown of the model in business areas
- Process Area Model – description of the second breakdown of the areas in process areas
- Context Classification Scheme – identification of the context as described by the UMM
- Business Process elaboration – description of the business process using
  - UML business operation activity diagram
  - Business process use case and description
- Business Process Collaboration – description of the collaboration using
  - Business collaboration use case and description
  - Business process collaboration activity diagram

The business process definition in UN/CEFACT requires the creation of many diagrams and views of the business processes breaking them down into the structure of business area, process area and business process. This means that the e-Documents created at later stages have to follow a strict pattern on the business process.

The following table compares the fulfilment of the high level requirements when using the UN/CEFACT:

Table 10 – High level requirements using UN/CEFACT methods for Requirements gathering

ID	Description	UN/CEFACT
HLR-1	Describe specific goals to be achieved with the exchange of electronic documents.	No
HLR-2	Define the requirements. This can be done using techniques such as: - Assertions: high-level statements about the business transaction. - Use case scenarios using real-life examples to exemplify the business process flow; and - Business process choreography: a formal description of the interactions and exchange of information that takes place.	Yes
HLR-3	Traceability: goals and requirements should be individually identifiable and traceable. This means that requirements should be linked to goals.	No

UN/CEFACT engineering methods use the following tools to produce the expected outcomes:

Outcome	UN/CEFACT
A list of comprehensive goals described in text	Not applicable
A set of real-life examples covering all the cases that have to be addressed using the exchange of e-Documents	Not applicable
A list of textual requirements linked to the goals	Requirements are described in each BRS document in PDF format. The requirements are not formally stated nor linked to goals
Graphical and textual description of the business process	Each BRS document in PDF format has multiple UML diagrams with comments depicting the business process and the textual description

3.1.1.2. Information modelling

UN/CEFACT uses the Requirements Specification Mapping (RSM) to describe the Conceptual Model required by the BRS.

A Requirement Specification Mapping consists of:

- A Conceptual Model Class
- A Logical Message Diagram
- Attributes of Business Entities
- References to the Core Components Library – There is a need to harmonize any new term that appears in a new electronic document, starting an update cycle that has to be harmonized within the UN/CEFACT library maintenance team.
- Message Assembly in a Class Diagram and a spread sheet

The next figure depicts an example of the message assembly model for a UN/CEFACT e-Document using a UML Class diagram.

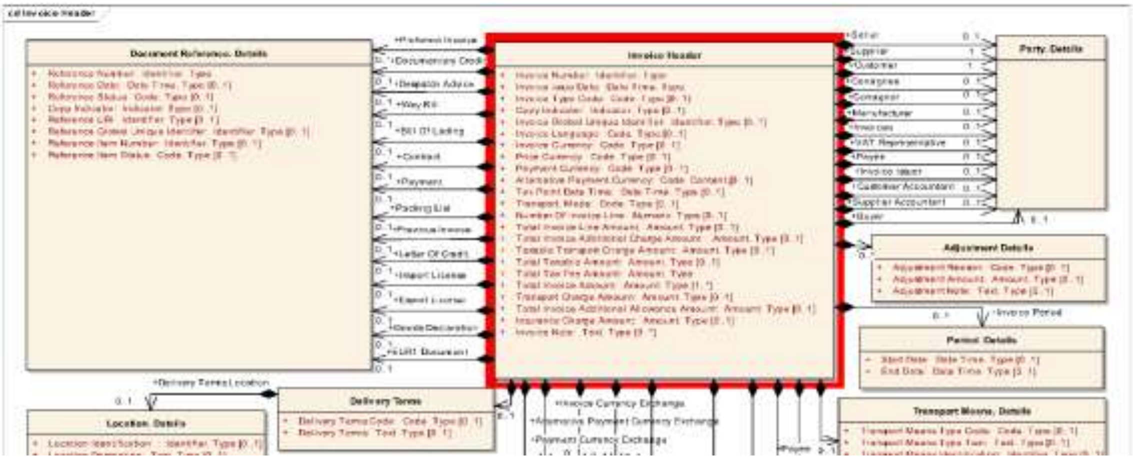


Figure 6 – UML Class diagram UN/CEFACT

A canonical model follows this conceptual model. The canonical model is basically identical to the conceptual model but using standard core components.

After the models, there is a section in the RSM listing the entities and attributes, the Basic Information Entities (BIE)[19], which identify:

- **BBIE Business term** – The name of the Basic Business Information Entity (BBIE). Each name is composed of its class name and the attribute name as identified in the class.
- **Core Component Type** – The core component type that has been used to identify its data type.

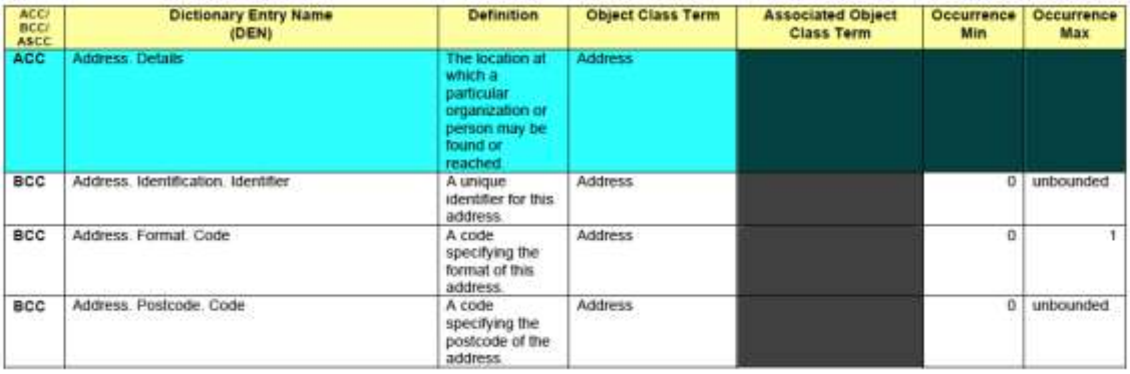
[19] The terminology is from the ebXML Core Component Technical Specification (ISO 15000-5).

- **Core Component Dictionary Entry Name** – If the BBIE is based on an existing basic core component, the name of the basic core component is indicated.
- **Restrictions** – The restrictions that have to be applied to the generic data type. These restrictions may be, for example, size, format or particular pattern matching.
- **Code list extension / Restriction** – Any codelist restrictions or extensions that have to be applied to the BBIE. Code list extensions may be required in a context where a standardized code list does not exist.

Then the aggregated business information entities are specified:

- **ABIE Business term** - The ABIE class name that identified the class in the UML model.
- **Aggregate Core Component Dictionary Entry Name** - The identification of the ACC that is the origin of the ABIE.
- **Restrictions (BCC to be omitted)** - The identification of each BCC that is not to be employed in the ABIE.
- **Extensions** - The identification of ABIE that is used as an additional attribute to extend the original ACC.

This process requires the project team to analyse the CCL and identify the properties of the class to be excluded from every Business Information Entity. The RSM establishes the reference to the Core Component Library.



ACC/ BCC/ ASCC	Dictionary Entry Name (DEN)	Definition	Object Class Term	Associated Object Class Term	Occurrence Min	Occurrence Max
ACC	Address. Details	The location at which a particular organization or person may be found or reached	Address			
BCC	Address. Identification. Identifier	A unique identifier for this address.	Address		0	unbounded
BCC	Address. Format. Code	A code specifying the format of this address.	Address		0	1
BCC	Address. Postcode. Code	A code specifying the postcode of the address.	Address		0	unbounded

Figure 7 – Core Component Reference

Finally, the list of Core Components is also added in the RSM:

- **Dictionary Entry Name** following the CCTS standard.
- **Definition** of the Core Component.
- **Object Class** name.
- **Property Term** identifying the name of the property.
- **Datatype Qualifier** to qualify the Core Component Type.
- **Representation Term** identifying the representation of the element.
- **CC Type** identifying the Core Component Type.

- **Associated Object Class Term** describing the object class associated as a property.
- **Occurrence Min** describing the minimum occurrences of the element.
- **Occurrence Max** describing the maximum occurrences of the element.

It has to be noted that the UN/CEFACT RSM does not provide contextual semantics of the Business Information Entities. It points to the definitions of the Core Components as defined in the Core Component Library.

Every new business term needs formal approval from the UN/CEFACT library maintenance group to be included in the Core Component Library. This is a standardization step that has to be performed if there are new business terms required in a new e-Document. This means that the e-Document format cannot be created without the participation of the UN/CEFACT library maintenance team unless all its terms are already found in the CCL.

The ebXML Core Components Technical Specification promotes extension-by-restriction. This means that it is assumed that all the terms exist in the CCL and the only action is restricting them to only those required terms. This represents two challenges:

1) The process to extend the model implies the work of the UN/CEFACT library maintenance team is complex
2) The "based on" relationship between Core Components and Information Entities is loose and promotes terminology inheritance but can prevent interoperability due to different derivation of the same Core Component by different groups.

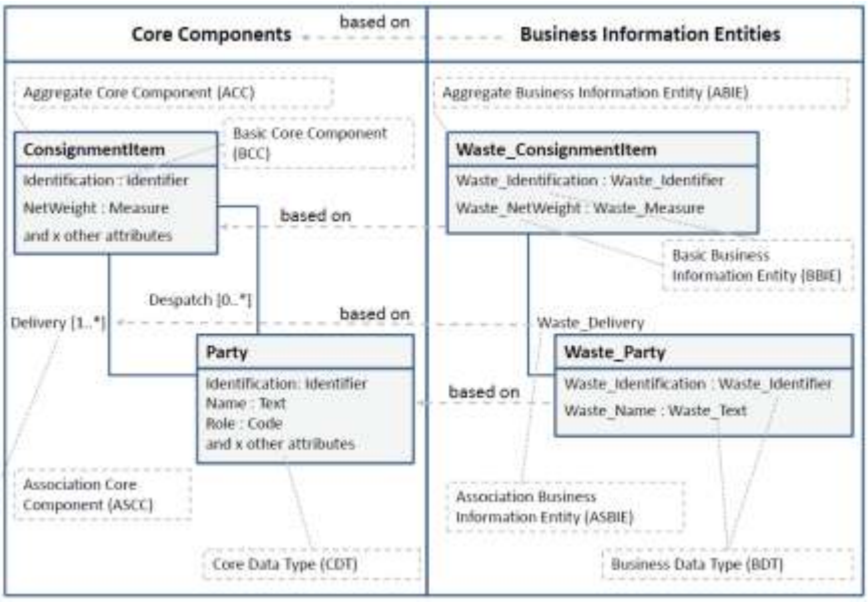

Core Components
based on
Business Information Entities
Aggregate Core Component (ACC)
Aggregate Business Information Entity (ABIE)
Basic Core Component (BCC)
ConsignmentItem
Identification : Identifier
NetWeight : Measure
and x other attributes
Waste_ConsignmentItem
Waste_Identification : Waste_Identifier
Waste_NetWeight : Waste_Measure
based on
Basic Business Information Entity (BBIE)
Despatch [0..*]
Delivery [1..*]
based on
Waste_Delivery
Party
Identification: Identifier
Name : Text
Role : Code
and x other attributes
Waste_Party
Waste_Identification : Waste_Identifier
Waste_Name : Waste_Text
based on
Association Core Component (ASCC)
Association Business Information Entity (ASBIE)
Core Data Type (CDT)
Business Data Type (BDT)

Figure 8 – UN/CEFACT CCTS: information modelling

The following table compares the fulfilment of the high level requirements when using the UN/CEFACT methodology:

Table 11 – High level requirements using UN/CEFACT for information modelling

ID	Description	UN/CEFACT
HLR-4	Capture business terms in an information model describing the explicit semantics of every data element, its attributes, cardinalities and relationships.	The semantics of the business concepts are in the CCL Library, the RSM captures and restricts the data elements but not the semantic concepts
HLR-5	Describe relationships between information components and requirements.	The relationships between components are constrained to the ones in the CCL
HLR-6	Depict the information model requirements using conceptual modelling languages such as UML, SBVR and repositories ISO11179 (Metadata Registry).	The UMM uses UML class diagrams. Dictionary Entry Names use CCTS (partially a derivation from ISO 11179).
HLR-7	Identify and reuse semantics and concepts from standardized vocabularies.	Business Information Entities can only be derived from the CCL

UN/CEFACT e-Document engineering methods use the following tools to produce the expected outcomes:

Outcome	UN/CEFACT
A set of concepts with unique semantic definition covering all the elements in the information model	No explicit tool, UN/CEFACT reuses common definitions from the CCL. Representations are MS-Excel spreadsheets and GEFEG.FX models.
A set of objects or classes with properties and relationship between them, describing cardinalities	Uses GEFEG.FX to depict the UML diagrams.
Optionally graphical representation of the information models	Uses GEFEG.FX to depict the UML diagrams

3.1.1.3. Business rule definition

UN/CEFACT does not differentiate a phase to define additional business rules. When creating the RSM model, some of the data elements are constrained with patterns or code lists.

The following table compares the fulfilment of the high level requirements when using the UN/CEFACT:

Table 12 – High level requirements using UN/CEFACT methods for business rule definition

ID	Description	UN/CEFACT
HLR-8	Identify the integrity constraints on the information model and describe them as business rules.	Only cardinalities are established in the data model.
HLR-9	Define inferences and mathematical calculations that the e-Document elements must fulfil.	Not supported
HLR-10	Define conditional business rules and co-occurrence constrains that the e-Document elements must fulfil.	Not supported
HLR-11	Define sets of allowed values for coded data elements.	Embedded as constrains in the data model

UN/CEFACT engineering methods use the following tools to produce the expected outcomes:

Outcome	UN/CEFACT
List of business rules in plain language	Not applicable
List of allowed code values for specific coded fields	Not applicable

3.1.1.4. Syntax binding

UN/CEFACT RSMs are bound to XML Schema using the UN/CEFACT Naming and Design Rules. There is no binding to any other syntax.

3.1.1.5. Schema production

UN/CEFACT uses its own naming and design rules to produce schemas. There are currently two versions of these, NDR 2.0[20] to support RSMs based on the CCL developed under CCTS 2.01 and NDR 3.0[21] to support RSMs based on the CCTS 3.0.

These NDRs define:

- The overall XML Schema structure.
- The relationship to CCTS defining the way of representation of ABIEs and BBIEs and data types
- The way the business messages use assembly components.

[20] http://www.unece.org/fileadmin/DAM/cefact/xml/XML-Naming-and-Design-Rules-V2.0.pdf

[21] http://www.unece.org/fileadmin/DAM/cefact/xml/UNCEFACT+XML+NDR+V3p0.pdf

- Naming and modelling constraints for elements and attributes, following the CCTS Dictionary Entry Names.
- The reusability scheme as a hybrid approach with some types declared global and some declared local.
- The namespace scheme for the components in the message
- The names of the XML Schema files and the imports and includes they have to use.
- The way to provide schema location.
- The versioning scheme for UN/CEFACT XSD schemas.

There are some tools implementing the UN/CEFACT NDR. From these tools the actual XSD Schemas can be produced using the data model as the input.

Table 13 – High level requirements using UN/CEFACT for schema production

ID	Description	UN/CEFACT
HLR-16	Map common information model components to available Common Vocabulary schemas.	This is done in the information model phase, mapping the Business Information Entities to the Core Components.
HLR-17	Create new syntax e-Document formats using a standard Naming and Design Rules (such as for example UBL or UN/CEFACT) to automate the schema production.	Yes, NDR 2.0 and NDR 3.0 are used to produce XSD Schemas and the production is automated using different tools.
HLR-18	Create validation artefacts for business rules and code lists.	No additional artefacts are created.

UN/CEFACT e-Document engineering methods use the following tools to produce the expected outcomes:

Outcome	UN/CEFACT
XSD Schema for the new e-Document format	GEFEG.FX is used to produce the XSD Schemas based on NDR 2.0 Cloud Data Technologies is used to produce the XSD Schemas based on NDR 3.0
Validation artefacts including business rules and code lists	Not applicable
Guidelines on the usage of the syntax	GEFEG.FX is used to produce the XSD Schema Guidelines for NDR 2.0

3.1.1.6. Governance

UN/CEFACT is an intergovernmental committee of the UNECE (United Nations Economic Commission for Europe) and provides two sets of libraries (and associated e-Document formats) per year. Each year has an A and a B set of standards. These versions are not backward compatible. This is the method used by UN/CEFACT to improve their standards adding more requirements twice a year based on the input from the team members.

Table 14 – HLR using UN/CEFACT for Governance

ID	Description	UN/CEFACT
HLR-19	Set up a governance organization for long-term sustainability of the e-Document format.	The UN/CEFACT Organization governs the standards they produce
HLR-20	Define the maintenance and versioning mechanisms for the e-Document format.	Library maintenance and associated NDR describes how to version the e-Document formats.
HLR-21	Establish policies and procedures for the use and maintenance of the e-Document format.	The UN/CEFACT Organization governs the standards they produce

UN/CEFACT e-Document engineering methods use the following tools to produce the expected outcomes:

Outcome	UN/CEFACT
Maintenance and versioning procedure	Based on internal procedures and the Open Development Process[22]
Governance policy	Covered by the UN/CEFACT structure, mandate, terms of reference and procedures [23] and supporting documents.

3.1.1.7. Conformance testing

UN/CEFACT does not provide conformance testing artefacts or test cases.

3.1.2.Summary of CAMSS assessment

The assessment of the UN/CEFACT e-Document engineering method in the CAMSS spreadsheet templates allows for the calculation of an automated score. This score is summarised in the below table. In Annex III.1.1, an overview of the CAMSS assessments is presented for the UN/CEFACT e-Document engineering method.

[22] http://www.unece.org/fileadmin/DAM/cefact/cf_plenary/plenary12/ECE_TRADE_C_CEFACT_2010_24_Rev.2E_ODP_revised.pdf

[23] http://www.unece.org/fileadmin/DAM/cefact/documents/ECE_TRADE_C_CEFACT_2010_15_Rev_5_final.pdf

Table 15 – CAMSS Assessment Score UN/CEFACT e-Document engineering method

Category	Automated Score [#Yes/(#Yes + #No)]	CAMSS Assessment Strength [(#Yes + #No)/#Criteria per category]
Applicability	100%	88%
Maturity	86%	100%
Openness	100%	100%
Intellectual property rights	100%	100%
Market support	75%	80%
Potential	100%	62%
Coherence	50%	67%
Overall score:	87%	85%

Figure 9 refers to the CAMSS assessment score of the UN/CEFACT e-Document engineering method. It represents the automated score (%) for the e-Document engineering method, as well as the amount of "yes" and "no" answers per category of CAMSS criteria.

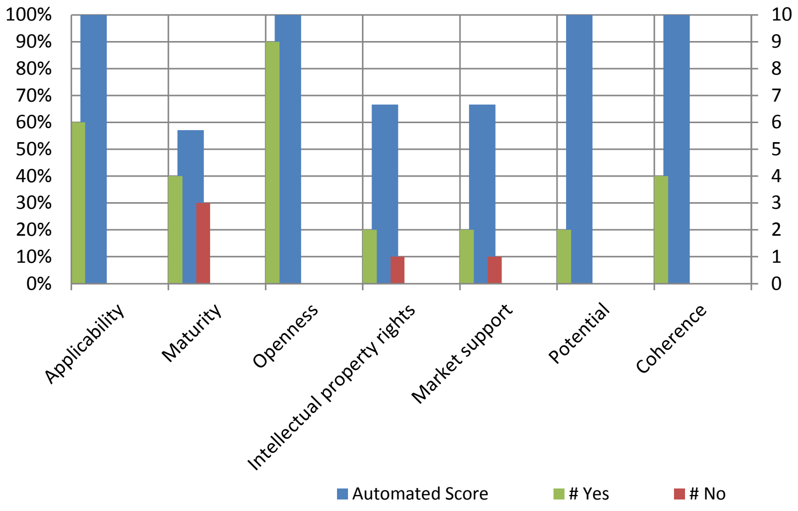


Figure 9 – Summary graph with the CAMSS assessment scores of the UN/CEFACT e-Document engineering method

3.2. OASIS UBL method

This section contains a description and assessment of the e-Document engineering method of the OASIS Universal Business Language (UBL) according to the assessment categories in CAMSS.

3.2.1.Description

The OASIS Universal Business Language (UBL) Technical Committee (TC) has applied a pragmatic e-Document engineering method for the creation and maintenance of the UBL standard, in conformance with UN/CEFACT Core Components Technical Specification (CCTS) Version 2.01 – ISO TS15000-5:2005. The OASIS UBL TC method is primarily defined in the Guidelines for Customization[24]

- **The UBL 2.0 Guidelines for Customization**: When modifying UBL schemas according to particular requirements, the UBL Guidelines for Customization specification provides recommendations on how to develop custom schemas in order to be UBL conformant or compatible. These guidelines aim to "*maintain a common understanding of the meaning of information being exchanged between specific implementations*" [OASIS, 2009]. UBL conformance implies that a "*UBL conformant document instance is an instance that validates against a UBL standard schema. A UBL conformant schema is a schema that will only validate UBL conformant instances*" [OASIS, 2009].

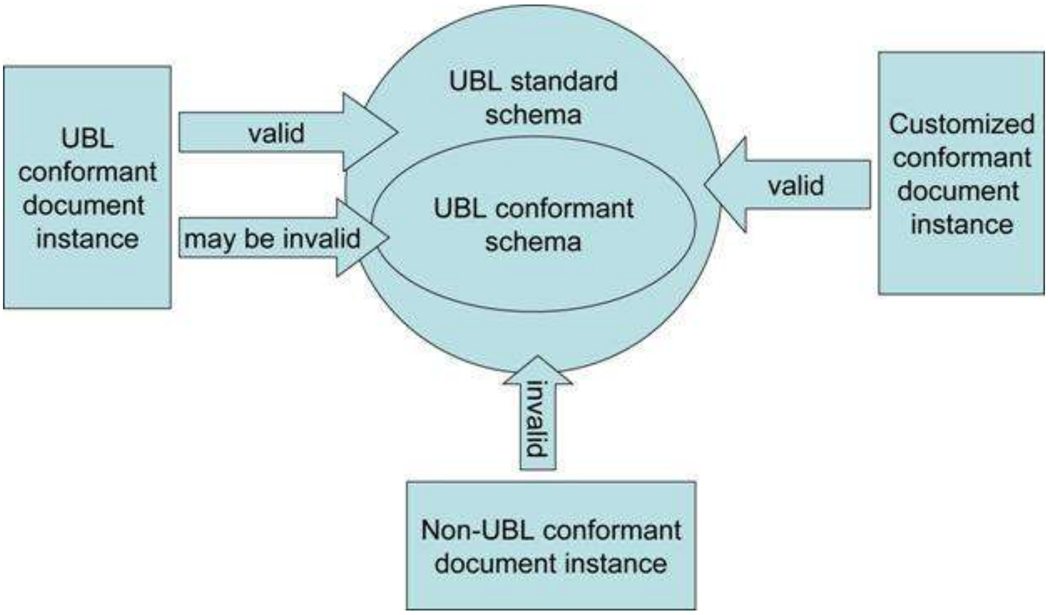

UBL conformant document instance
valid
may be invalid
UBL standard schema
UBL conformant schema
valid
Customized conformant document instance
invalid
Non-UBL conformant document instance

Figure 10 – UBL Conformant schemas and document instances [OASIS, 2009]

UBL compatibility implies the consistency with the UBL principles, including the ebXML CCTS and the UBL Naming and Design Rules, and the information entities.

[24] UBL 2 Guidelines for Customization, First Edition: http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.pdf

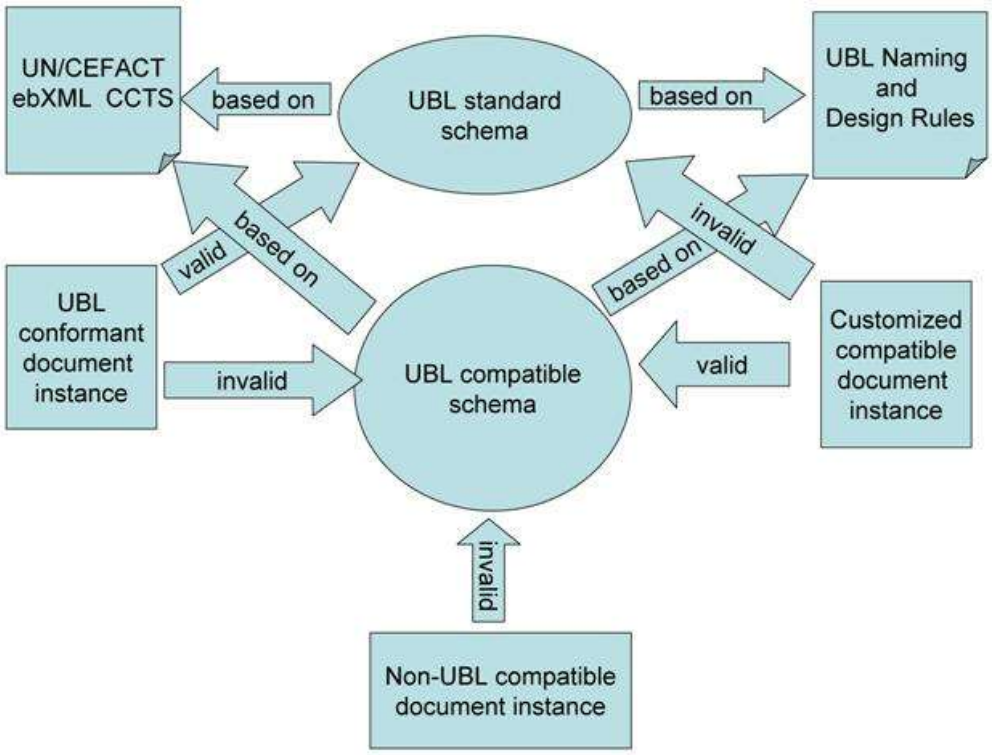

UN/CEFACT ebXML CCTS
based on
UBL standard schema
based on
UBL Naming and Design Rules
valid
based on
invalid
based on
UBL conformant document instance
invalid
UBL compatible schema
valid
Customized compatible document instance
invalid
Non-UBL compatible document instance

Figure 11 – Compatible schemas and document instances [OASIS, 2009]

This is supported by a set of standards and committee specifications consisting of:

- **The UBL 2.1 Standard**[25]: The UBL 2.1 Standard contains the 65 business document schemas. These XSD schemas represent the UBL 2.1 document types and library components. It also contains the business concepts and the multiple requirements.
- **The UBL 2.0 Naming and Design Rules**[26] **(NDR)**: For the development of extensible and reusable UBL schemas, a set of XML schema design rules and naming conventions is described in the UBL 2.0 NDR specification. These rules and guidelines are agreed by the OASIS UBL TC. In order to develop consistent UBL schemas, general XML constructs are described. These include the overall structure of a schema, naming and modelling constraints, schemes related to the reusability, extension, namespace and versioning, a modularity strategy, and additional documentation requirements. Three types of naming rules exist: general naming rules, type naming rules and element naming rules. Other definitions and XSD rules are also included in the specification.
- **The UBL 2.0 International Data Dictionary** (IDD): The reference language for the business terms defined in UBL is English. However, an

[25] *Universal Business Language Version 2.1* 04 November 2013. OASIS Standard:
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.zip

[26] *Universal Business Language Version 2.0* Naming and Design Rules:
http://docs.oasis-open.org/ubl/cs01-UBL-2.0-NDR/cs01-UBL-2.0-NDR.html

International Data Dictionary has been standardized to provide translations of these terms and definitions of UBL 2.0 in Japanese, Chinese, Italian and Spanish in IDD Volume 1[27]. A draft version of the German translation is available in IDD Volume 2[28]. It must be noted that with the release of UBL 2.1, some of the business terms and definitions have been revised, which also affects their translation. However, these replacements are not completed yet.

- **Two-phase validation mechanism**: A two-phase model of data verification allows for a flexible application of code lists within the standard structure. In the first validation phase, a generic schema validator checks the structure and vocabulary of the UBL instance against a standard UBL schema. The second phase validates the code list values in the UBL instance against values from configuration files of external code lists.

OASIS Universal Business Language 2.1 Standard

The Universal Business Language (UBL) is developed by the OASIS Technical Committee in a public and transparent process. The model of open standards describes generic XML business documents that can be exchanged electronically. UBL offers a standards-based infrastructure by providing a universal syntax for business documents and by operating within a standard business framework. Its main goal is to achieve data interoperability in the global electronic business environment. UBL is designed to be directly implemented in several domains, covering business, legal, auditing, and records management practices. Primarily, it serves as the foundation for European public procurement frameworks.

Initially, UBL was created as part of the UN/CEFACT – OASIS ebXML partnership, aiming to standardize XML data formats for electronic business. Even though the use of UBL has now expanded beyond ebXML and is independent of any framework, it still remains a component of ebXML. UBL is also applied as a reference format in the CEN Workshop Agreement (CWA) 16667 and other standards [OASIS, 2014].

UBL was first released as an OASIS Standard in 2004 **[OASIS, 2013]**. Since then, two version upgrades have been carried out. UBL is considered as one of the most mature and implemented OASIS Standards.

UBL 1.0 contained eight document types (*Order, Order Response, Order Response Simple, Order Change, Order Cancellation, Despatch Advice, Receipt Advice*, and *Invoice*), which are still included in the current version. The assumed process context for these UBL document types was an "Order-to-Invoice" business process.

As the amount of business documents expanded, new business content went beyond the basic order-to-invoice documents of UBL 1.0. A major revision led to **UBL 2.0**, increasing the number of business documents to 31. The use of UBL 1.0

[27] *Universal Business Language Version 2.0* IDD Volume 1: http://docs.oasis-open.org/ubl/idd/cs-UBL-2.0-idd01/cs-UBL-2.0-idd01.zip

[28] *Universal Business Language Version 2.0* IDD Volume 2: http://lists.oasis-open.org/archives/ubl/200906/msg00008.html

is deprecated, as no backward-compatibility is available for versions of UBL starting from 2.0 with UBL 1.0. This is mainly due to a change in the use of the XML schema methodology in UBL 2.0, featuring the adoption of global scoping for all element types. The original business documents of UBL 1.0 were revised according to the applied changes. The remaining 23 business documents define generic supply chain and procurement processes:

- Sourcing : *Catalogue, Catalogue Deletion, Catalogue Item Specification Update, Catalogue Pricing Update, Catalogue Request, Quotation, Request for Quotation*
- Fulfilment: *Bill of Lading, Certificate of Origin, Forwarding Instructions, Packing List, Transportation Status, Waybill*
- Billing: *Credit Note, Debit Note, Freight Invoice, Reminder, Self-Billed Credit Note, Self-Billed Invoice*
- Payment: *Remittance Advice, Statement*
- Supplementary: *Application Response, Attached Document*

The OASIS Standardization for UBL 2.0 was achieved in 2006, and the updates and corrections were implemented in 2008.

An additional technical revision was performed, resulting in a minor upgrade to **UBL 2.1**. The changes included financial information enhancements, revised data definitions and other modifications of library elements. 34 new document types were included in the upgrade:

- eTendering*: Awarded Notification, Call for Tenders, Contract Award Notice, Contract Notice, Guarantee Certificate, Tender, Tender Receipt, Tenderer Qualification, Tenderer Qualification Response, Unawarded Notification*
- Collaborative planning, forecasting, and replenishment: *Exception Criteria, Exception Notification, Forecast, Forecast Revision, Item Information Request, Prior Information Notice, Trade Item Location Profile*
- Vendor Managed Inventory: *Instruction for Returns, Inventory Report, Product Activity, Retail Event, Stock Availability Report*
- Intermodal Freight Management: *Goods Item Itinerary, Transport Execution Plan, Transport Execution Plan Request, Transport Progress Status, Transport Progress Status Request, Transport Service Description, Transport Service Description Request, Transportation Status, Transportation Status Request*
- Utility billing: *Utility Statement*
- Supplementary: *Document Status, Document Status Request*

Backward compatibility is available for all UBL versions since 2006.

In the remaining subsections, we will describe the fit of the UBL method with the high-level requirements that are elicited in Chapter 2. These requirements are structured according to the following steps:

1. Requirement gathering;

2. Information modelling;
3. Business rule definition;
4. Syntax binding;
5. Schema production;
6. Governance; and
7. Conformance testing.

Figure 12 depicts the OASIS UBL TC method. It consists of the following:

- **Business process, actors and roles**
- **Document engineering:** The conceptual models are created using online tools (with the data models published as spreadsheets and UML models). UBL schemas are considered to be modular, reusable and extensible in order to meet any particular and industry-specific requirements. The UBL library of reusable data components complies with the UN/CEFACT ebXML CCTS version 2.01 and is provided as Business Information Entities (BIEs). The royalty-free library provides common XML components, which form the XML business documents. These e-Document models are, for instance, purchase orders and invoices that can be used in a generic procurement and transportation context.

  External code lists are available for coded data types. If needed, new BIEs can be created using the UBL Extension methodology.
- **Document assembly:** The business documents are transformed into W3C XSD schema syntax, according to the UBL Naming and Design Rules.

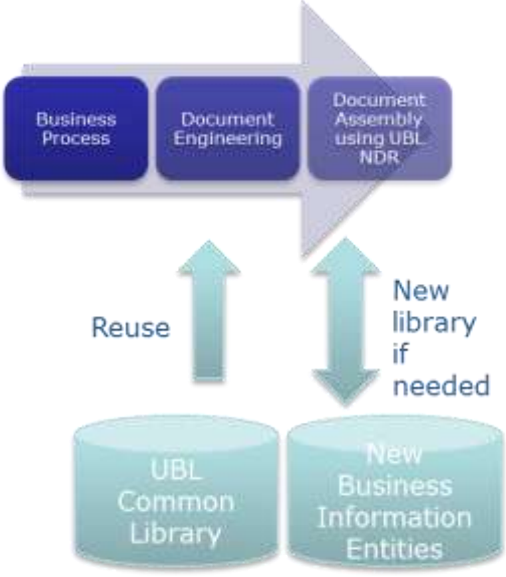


Figure 12 – Applicability of the OASIS UBL TC method

3.2.1.1. Requirement gathering

UBL describes the goals and requirements textually in the UBL main document. The documentation uses UML Activity Diagrams to depict the business process choreography.

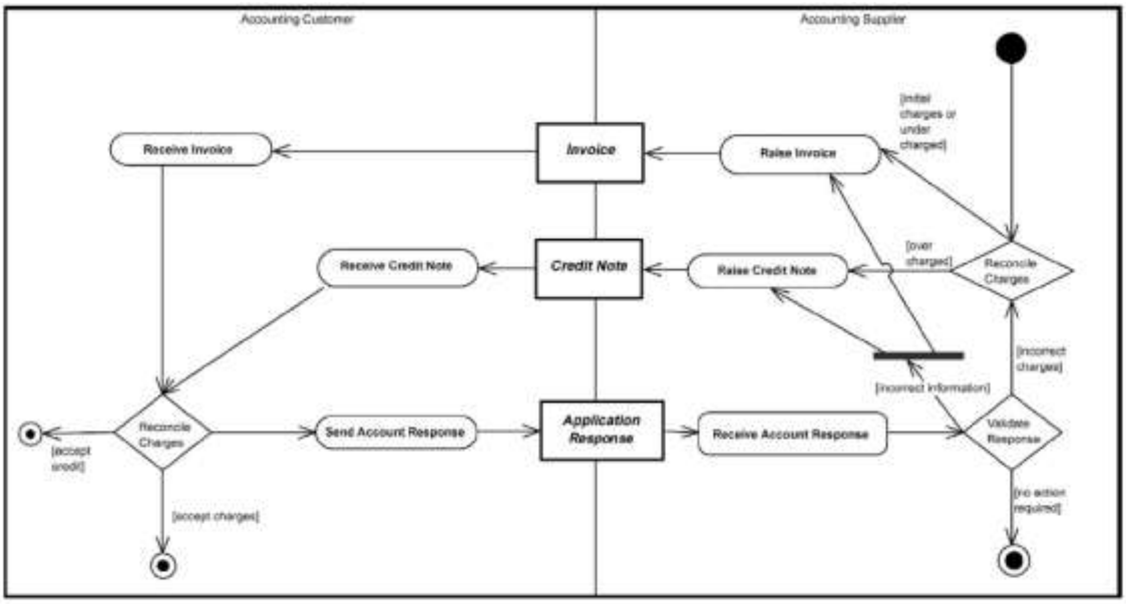


Figure 13 – UBL billing UML activity diagram

The business process description in UBL establishes a set of business rules and a generic flow for the process. It focuses more on explaining the context of use of the transactions that trying to standardize the process itself.

The focus for UBL is on the analysis of the e-Documents rather than on the standardization of the business process that produces or consumes them. This allows the UBL e-Documents to be used in different scenarios.

The following table compares the fulfilment of the high level requirements when using the UBL engineering methods.

Table 16 – Comparing assessed methods when gathering requirements

ID	Description	UBL
HLR-1	Describe specific goals to be achieved with the exchange of electronic documents	No
HLR-2	Define the requirements. This can be done using techniques such as: - Assertions: high-level statements about the business transaction. - Use case scenarios using real-life examples to exemplify the business process flow; and - Business process choreography: a formal description of the interactions and exchange of information that takes place	Yes

HLR-3	Traceability: goals and requirements should be individually identifiable and traceable. This means that requirements should be linked to goals.	No

UBL has the following tools to produce the expected outcomes:

Outcome	UBL
A list of comprehensive goals described in text	Not applicable
A set of real-life examples covering all the cases that have to be addressed using the exchange of e-Documents	Not applicable
A list of textual requirements linked to the goals	Requirements are provided in a single UBL main document in HTML, XML and PDF format. They are not formally stated nor linked to goals
Graphical and textual description of the business process	The UBL main document contains UML Activity Diagrams and textual descriptions depicting the business processes covered in the specification.

3.2.1.2. Information modelling

The Universal Business Language (UBL) standard is based on a common semantic data model described in the UBL main document, and depicted below

:

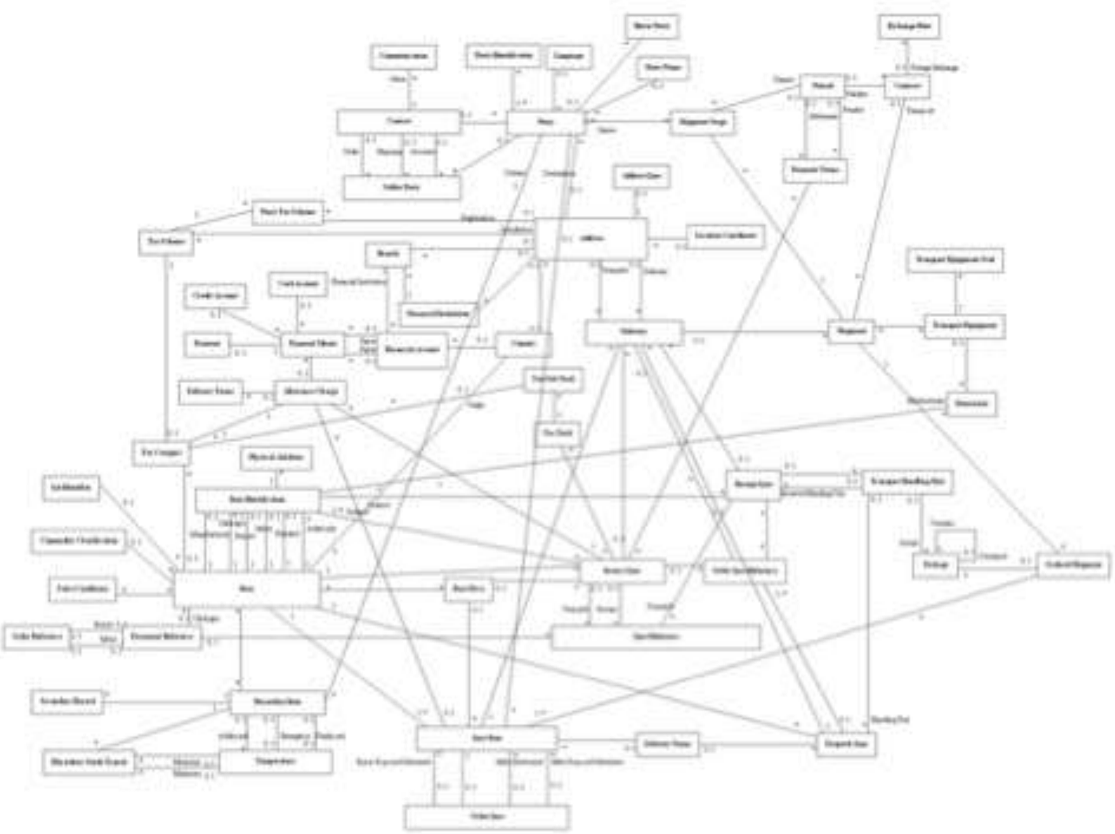
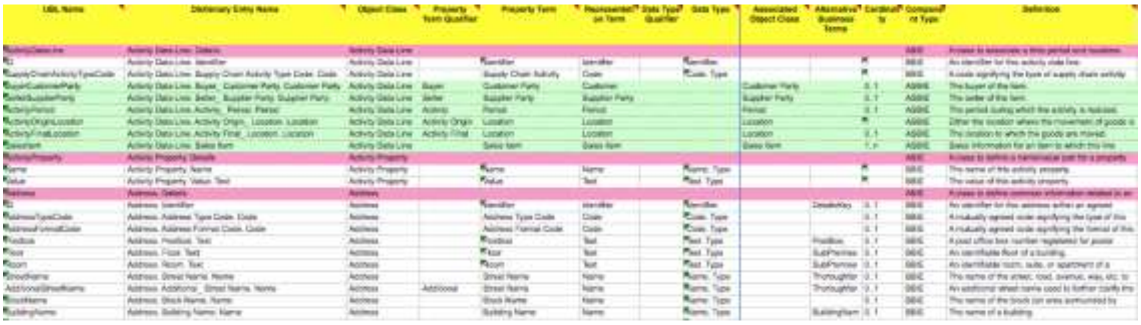


Figure 14 – UBL Data Model Foundation

Based on that semantic common library, the UBL TC captures business terms and definitions following CCTS specifications in an online tool. Spreadsheets like the one below can be then created from the tool.



UBL Name	Dictionary Entry Name	Object Class	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Data Type	Associated Object Class	Alternative Business Terms	Cardinality	Component Type	Definition
ActivityDataLine	Activity Data Line. Details	Activity Data Line									ABIE	A class to associate a time period and locations
ID	Activity Data Line. Identifier	Activity Data Line		Identifier	Identifier		Identifier. Type			1	BBIE	An identifier for this activity data line.
SupplyChainActivityTypeCode	Activity Data Line. Supply Chain Activity Type Code. Code	Activity Data Line		Supply Chain Activity	Code		Code. Type			1	BBIE	A code signifying the type of supply chain activity
BuyerCustomerParty	Activity Data Line. Buyer_ Customer Party. Customer Party	Activity Data Line	Buyer	Customer Party	Customer			Customer Party		0..1	ASBIE	The buyer of the item.
SellerSupplierParty	Activity Data Line. Seller_ Supplier Party. Supplier Party	Activity Data Line	Seller	Supplier Party	Supplier Party			Supplier Party		0..1	ASBIE	The seller of the item.
ActivityPeriod	Activity Data Line. Activity_ Period. Period	Activity Data Line	Activity	Period	Period			Period		0..1	ASBIE	The period during which the activity is realized.
ActivityOriginLocation	Activity Data Line. Activity Origin_ Location. Location	Activity Data Line	Activity Origin	Location	Location			Location		1	ASBIE	Either the location where the movement of goods is
ActivityFinalLocation	Activity Data Line. Activity Final_ Location. Location	Activity Data Line	Activity Final	Location	Location			Location		0..1	ASBIE	The location to which the goods are moved.
SalesItem	Activity Data Line. Sales Item	Activity Data Line		Sales Item	Sales Item			Sales Item		1..n	ASBIE	Sales information for an item to which this line
ActivityProperty	Activity Property. Details	Activity Property									ABIE	A class to define a name/value pair for a property
Name	Activity Property. Name	Activity Property		Name	Name		Name. Type			1	BBIE	The name of this activity property.
Value	Activity Property. Value. Text	Activity Property		Value	Text		Text. Type			1	BBIE	The value of this activity property.
Address	Address. Details	Address									ABIE	A class to define common information related to an
ID	Address. Identifier	Address		Identifier	Identifier		Identifier. Type		DetailsKey	0..1	BBIE	An identifier for this address within an agreed
AddressTypeCode	Address. Address Type Code. Code	Address		Address Type Code	Code		Code. Type			0..1	BBIE	A mutually agreed code signifying the type of this
AddressFormatCode	Address. Address Format Code. Code	Address		Address Format Code	Code		Code. Type			0..1	BBIE	A mutually agreed code signifying the format of this
Postbox	Address. Postbox. Text	Address		Postbox	Text		Text. Type		PostBox	0..1	BBIE	A post office box number registered for postal
Floor	Address. Floor. Text	Address		Floor	Text		Text. Type		SubPremise	0..1	BBIE	An identifiable floor of a building.
Room	Address. Room. Text	Address		Room	Text		Text. Type		SubPremise	0..1	BBIE	An identifiable room, suite, or apartment of a
StreetName	Address. Street Name. Name	Address		Street Name	Name		Name. Type		Thoroughfare	0..1	BBIE	The name of the street, road, avenue, way, etc. to
AdditionalStreetName	Address. Additional_ Street Name. Name	Address	Additional	Street Name	Name		Name. Type		Thoroughfare	0..1	BBIE	An additional street name used to further clarify the
BlockName	Address. Block Name. Name	Address		Block Name	Name		Name. Type			0..1	BBIE	The name of the block (an area surrounded by
BuildingName	Address. Building Name. Name	Address		Building Name	Name		Name. Type		BuildingName	0..1	BBIE	The name of a building.

Figure 15 – UBL Common Library (spreadsheet view)

The elements captured in UBL are based upon the ebXML Core Component Technical Specification Version 2.01 on the basis of ISO 11179 naming rules:

- **UBL Name** – The UBL name is derived from the Dictionary Entry Name according to the UBL Naming and Design Rules.
- **Dictionary Entry Name** – Dictionary Entry Name is the unique official name of the Business Information Entity in the data dictionary.
- **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. They are called Classes in UML and Tables/Entities in database contexts.

- **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 (to database designers). 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.

The UBL package also contains the UML class diagram representation for all the UBL documents and classes.

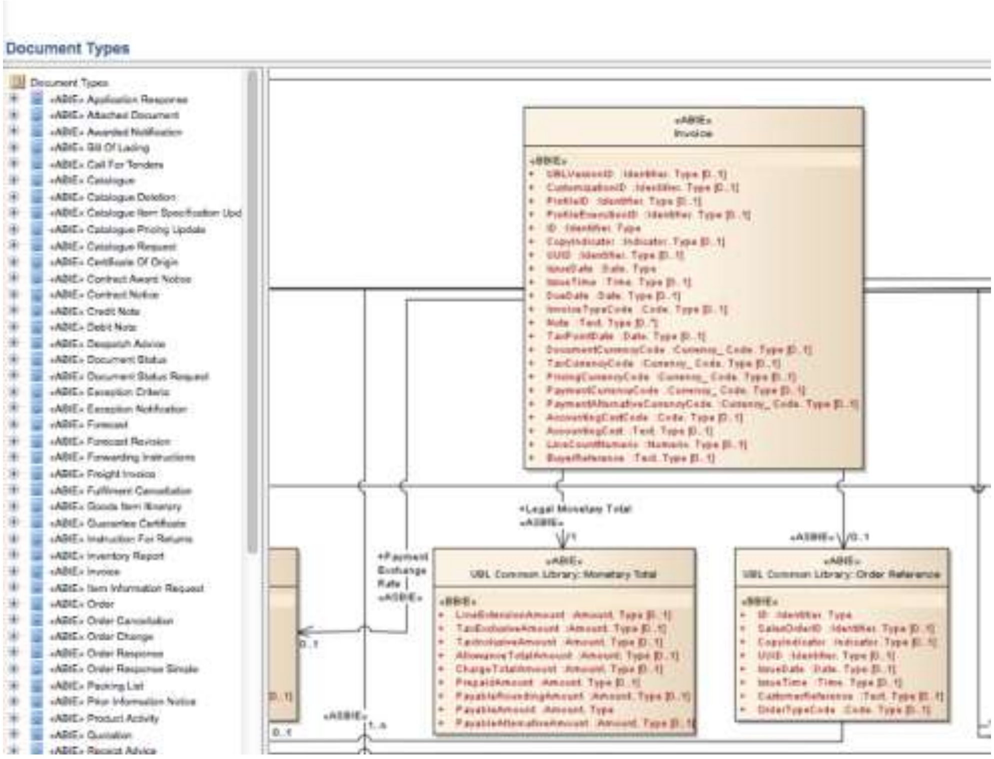

Document Types
Document Types
«ABIE» Application Response
«ABIE» Attached Document
«ABIE» Awarded Notification
«ABIE» Bill Of Lading
«ABIE» Call For Tenders
«ABIE» Catalogue
«ABIE» Catalogue Deletion
«ABIE» Catalogue Item Specification Upd
«ABIE» Catalogue Pricing Update
«ABIE» Catalogue Request
«ABIE» Certificate Of Origin
«ABIE» Contract Award Notice
«ABIE» Contract Notice
«ABIE» Credit Note
«ABIE» Debit Note
«ABIE» Despatch Advice
«ABIE» Document Status
«ABIE» Document Status Request
«ABIE» Exception Criteria
«ABIE» Exception Notification
«ABIE» Forecast
«ABIE» Forecast Revision
«ABIE» Forwarding Instructions
«ABIE» Freight Invoice
«ABIE» Fulfilment Cancellation
«ABIE» Goods Item Itinerary
«ABIE» Guarantee Certificate
«ABIE» Instruction For Returns
«ABIE» Inventory Report
«ABIE» Invoice
«ABIE» Item Information Request
«ABIE» Order
«ABIE» Order Cancellation
«ABIE» Order Change
«ABIE» Order Response
«ABIE» Order Response Simple
«ABIE» Packing List
«ABIE» Prior Information Notice
«ABIE» Product Activity
«ABIE» Quotation
«ABIE» Receipt Advice
«ABIE»
Invoice
«BBIE»
+ UBLVersionID :Identifier. Type [0..1]
+ CustomizationID :Identifier. Type [0..1]
+ ProfileID :Identifier. Type [0..1]
+ ProfileExecutionID :Identifier. Type [0..1]
+ ID :Identifier. Type
+ CopyIndicator :Indicator. Type [0..1]
+ UUID :Identifier. Type [0..1]
+ IssueDate :Date. Type
+ IssueTime :Time. Type [0..1]
+ DueDate :Date. Type [0..1]
+ InvoiceTypeCode :Code. Type [0..1]
+ Note :Text. Type [0..*]
+ TaxPointDate :Date. Type [0..1]
+ DocumentCurrencyCode :Currency_ Code. Type [0..1]
+ TaxCurrencyCode :Currency_ Code. Type [0..1]
+ PricingCurrencyCode :Currency_ Code. Type [0..1]
+ PaymentCurrencyCode :Currency_ Code. Type [0..1]
+ PaymentAlternativeCurrencyCode :Currency_ Code. Type [0..1]
+ AccountingCostCode :Code. Type [0..1]
+ AccountingCost :Text. Type [0..1]
+ LineCountNumeric :Numeric. Type [0..1]
+ BuyerReference :Text. Type [0..1]
+Legal Monetary Total
«ASBIE»
1
«ASBIE»
0..1
+Payment Exchange Rate
«ASBIE»
0..1
«ABIE»
UBL Common Library: Monetary Total
«BBIE»
+ LineExtensionAmount :Amount. Type [0..1]
+ TaxExclusiveAmount :Amount. Type [0..1]
+ TaxInclusiveAmount :Amount. Type [0..1]
+ AllowanceTotalAmount :Amount. Type [0..1]
+ ChargeTotalAmount :Amount. Type [0..1]
+ PrepaidAmount :Amount. Type [0..1]
+ PayableRoundingAmount :Amount. Type [0..1]
+ PayableAmount :Amount. Type
+ PayableAlternativeAmount :Amount. Type [0..1]
«ABIE»
UBL Common Library: Order Reference
«BBIE»
+ ID :Identifier. Type
+ SalesOrderID :Identifier. Type [0..1]
+ CopyIndicator :Indicator. Type [0..1]
+ UUID :Identifier. Type [0..1]
+ IssueDate :Date. Type [0..1]
+ IssueTime :Time. Type [0..1]
+ CustomerReference :Text. Type [0..1]
+ OrderTypeCode :Code. Type [0..1]
[0..1]
«ASBIE»
1..n
0..1

Figure 16 – UBL UML class diagram

When new information elements are not included in the UBL common library, it is possible to create another library using another namespace to add new aggregated or basic business information entities.

This new library can be used to create the new document models following the UBL Guideline on Customizations.

Table 17 – HLR using UBL for information modelling

ID	Description	UBL
HLR-4	Capture business terms in an information model describing the explicit semantics of every data element, its attributes, cardinalities and relationships.	The common library of UBL is published in spreadsheet, HTML, UML and XML formats. The XML format is normative. For additional business terms, you can capture the exact semantics of the information model using the customization methodology.
HLR-5	Describe relationships between information components and requirements.	The common library can express the relationships between information components using ABIEs, BBIEs and ASBIEs.
HLR-6	Depict the information model requirements using conceptual modelling languages such as UML, SBVR and repositories ISO11179 (Metadata Registry).	The UBL information model is represented in UML class diagrams.

HLR-7	Identify and reuse semantics and concepts from standardized vocabularies.	The semantics of the business terms not in the UBL common library can be reused from other standard vocabularies.

UBL engineering methods use the following tools to produce the expected outcomes:

Outcome	UBL
A set of concepts with unique semantic definition covering all the elements in the information model	A spreadsheet is used to capture all the business terms with their unique semantic definition based on the semantic common library.
A set of objects or classes with properties and relationship between them, describing cardinalities	The CCTS is used to describe the relationship between classes and the cardinalities of the elements.
Optionally graphical representation of the information models	A UML class diagram can be created with any modelling tool such as Enterprise Architect or MS Visio. Enterprise Architect format is published.

3.2.1.3. Business rule definition

UBL describes the code lists as separate elements. Code values are not coupled with the e-Document format (XSD Schemas).

Code lists, structures and cardinalities are the only business rules defined within the UBL TC. There are no conditional rules or mathematical calculation models defined as standard by the UBL TC.

The following table compares the fulfilment of the high level requirements when using UBL:

Table 18 – HLR using UBL methods for business rule definition

ID	Description	UBL
HLR-8	Identify the integrity constraints on the information model and describe them as business rules.	Not defined as business rules, but identified cardinalities and constraints for qualified information items[29].
HLR-9	Define inferences and mathematical calculations that the e-Document elements must fulfil.	Not formally defined in the specs. There are non-Normative Invoice Calculation Model Rules for UBL.

[29] http://docs.oasis-open.org/ubl/prd2-UBL-2.1/cva/UBL-DefaultDTQ-2.1.html

HLR-10	Define conditional business rules and co-occurrence constrains that the e-Document elements must fulfil.	Not defined
HLR-11	Define sets of allowed values for coded data elements.	Defined the coded values as Genericode files bound to a coded data.

UBL engineering methods use the following tools to produce the expected outcomes:

Outcome	UBL
List of business rules in plain language	Not applicable
List of allowed code values for specific coded fields	There are listed in the UBL-DefaultDTQ-2.1.html and Genericode files are provided for codelists.

3.2.1.4. Syntax binding

UBL does not bind the data model to any other XML vocabulary, however, the UBL e-Documents are also published as non-normative ASN.1 syntax and RelaxNG schemas.

3.2.1.5. Schema production

UBL produces XSD Schemas following their Naming and Design Rules.

The main differences with the UN/CEFACT NDR are in the structure of the schemas and the reusability of their elements. The pattern used in UBL allows for reusability of elements and types while the pattern in UN/CEFACT is a hybrid pattern, allowing the reusability of only certain elements and types. Another important difference is the possibility to extend the schemas in UBL, which is not possible in UN/CEFACT.

The current UBL NDR[30] defines:

- The overall schema structure defining the physical layout of the UBL schema documents and the requirement for all global elements in the document to be declared either in the Common Aggregated or Common Basic Components libraries.
- The naming and modelling constrains, identifying the spreadsheets as the main UBL library documentation and establishing that the XML element name is in the form of a Dictionary Entry Name (ISO 15000-5).
- That all elements are declared global and therefore reusable.

[30] http://docs.oasis-open.org/ubl/cs01-UBL-2.0-NDR/cs01-UBL-2.0-NDR.pdf

- A unique extension mechanism that allows sending additional information not covered in the UBL structure.
- The UBL namespace scheme.
- The versioning scheme, with the differentiation between major and minor versions. The latter does not break backwards compatibility.
- The modularity strategy that establishes how schemas are organized to produce a valid UBL XSD Schema.
- How to annotate and add documentation in the XSD Schema.
- The rules for naming elements and types.
- The rules for naming declarations and definitions
- The code lists as externally maintained components.
- Miscellaneous XSD rules to describe what elements are not to be used within UBL, such as substitutionGroups or choice constructs.

There are different tools that produce XSD Schemas based on the UBL NDR directly from the UBL models: eDoCreator (cloud-based solution), Crane UBL Generator (open-source application) or GEFEG.FX (licensed tool) are three of them.

The UBL TC has used eDoCreator and the Crane UBL Generator to produce the UBL 2.1 schemas to ensure consistency of the generated Schema.

Table 19 – HLR using UBL for schema production

ID	Description	UBL
HLR-16	Map common information model components to available Common Vocabulary schemas.	The mapping is done using the online model, where the name of the XML element is inferred from the Dictionary Entry Name. The vocabulary used is the UBL Common Library. It can be extended with new ABIES and BBIES.
HLR-17	Create new syntax e-Document formats using a standard Naming and Design Rules (such as for example UBL or UN/CEFACT) to automate the schema production.	UBL has used Crane UBL Generator and eDoCreator to produce UBL 2.1 XSD Schemas.
HLR-18	Create validation artefacts for business rules and code lists.	UBL has created a set of XSLT and validation scripts to validate code lists in a two-phase validation.

UBL e-Document engineering methods use the following tools to produce the expected outcomes:

Outcome	UBL

XSD Schema for the new e-Document format	eDoCreator and Crane UBL Generator to produce UBL 2.1 XSD Schemas.
Validation artefacts including business rules and code lists	UBL-DefaultDTQ-2.1 validates test samples in the two-phase validation methodology.
Guidelines on the usage of the syntax	There are no usage guidelines other than the main document.

3.2.1.6. Governance

UBL has been developed by a Technical Committee of OASIS (the Organization for the Advancement of Structured Information Standards). OASIS is a non-profit consortium that drives the development, convergence and adoption of open standards for the global information society. OASIS has transparent governance and operating procedures. The OASIS Members themselves set the OASIS technical agenda, using a lightweight process expressly designed to promote industry consensus and unite disparate efforts. Completed work is ratified by open ballot among members in the Technical Committees.

Table 20 – HLR using UBL for Governance

ID	Description	UBL
HLR-19	Set up a governance organization for long-term sustainability of the e-Document format.	OASIS is the organization governing the UBL TC. OASIS also plans to submit UBL to the ISO-IEC Joint Technical Committee 1 for recognition as an ISO-IEC standard.
HLR-20	Define the maintenance and versioning mechanisms for the e-Document format.	The UBL Technical Committee is responsible for the maintenance of UBL following OASIS procedures. UBL 2.1 is the third release since 2004 (1.0 in 2004, 2.0 in 2007 and 2.1 in 2013). The Technical Committee plans to maintain backward compatibility with any future versions to the 2.0 standard. The UBL NDR establishes the mechanism for maintaining the versions of the UBL deliverables.
HLR-21	Establish policies and procedures for the use and maintenance of the e-Document format.	OASIS is the organization governing the UBL TC.

UBL e-Document engineering methods use the following tools to produce the expected outcomes:

Outcome	UBL
Maintenance and versioning procedure	OASIS procedures an UBL NDR
Governance policy	OASIS procedures. In future these will be complemented by ISO-IEC JTC 1 procedures.

3.2.1.7. Conformance testing

UBL defines the Two-Phase Validation methodology to enable the validation of UBL conformant XML instances. The two-phase validation ensures the structure of the e-Document instance follows the UBL XSD schema and, as a second step, validates the values for the code lists defined in the standards.

To be conformant to UBL requires that document instances are valid instances of the normative XML Schema for that document type.

There are no business rules to test other than those defining the coded elements.

Table 21 – HLR using UBL for Conformance Testing

ID	Description	UBL
HLR-22	Define a testing framework for the e-Document instances.	The normative XML Schema defines valid UBL document instances. The UBL Two-Phase validation method allows for an extended testing framework for UBL document instances.

Tools to be used for conformance testing with UBL are:

Outcome	UBL
Conformance testing procedure	Numerous XML Schema validation tools, such as XML Spy from Altova, etc. For two-phase validation the UBL specification contains a *val* folder with scripts and *xslt* transformations that can be used to further validate UBL document instances for additional business rules and coded values.

3.2.2.Summary of CAMSS assessment

The CAMSS spreadsheet template allows for the calculation of an automated score for the OASIS UBL method. This score is summarised in the below table. In Annex III.1.2, an overview of the CAMSS assessments is presented for the OASIS UBL method.

Table 22 – CAMSS Assessment Score OASIS UBL method

Category	Automated Score	CAMSS Assessment Strength

	[#Yes/(#Yes + #No)]	[(#Yes + #No)/#Criteria per category]
Applicability	100%	88%
Maturity	86%	100%
Openness	100%	100%
Intellectual property rights	100%	100%
Market support	100%	80%
Potential	100%	62%
Coherence	33%	67%
Overall score:	88%	85%

Figure 17 refers to the CAMSS assessment score of the OASIS UBL method. It represents the automated score (%) for the e-Document engineering method, as well as the amount of "yes" and "no" answers per category of CAMSS criteria.

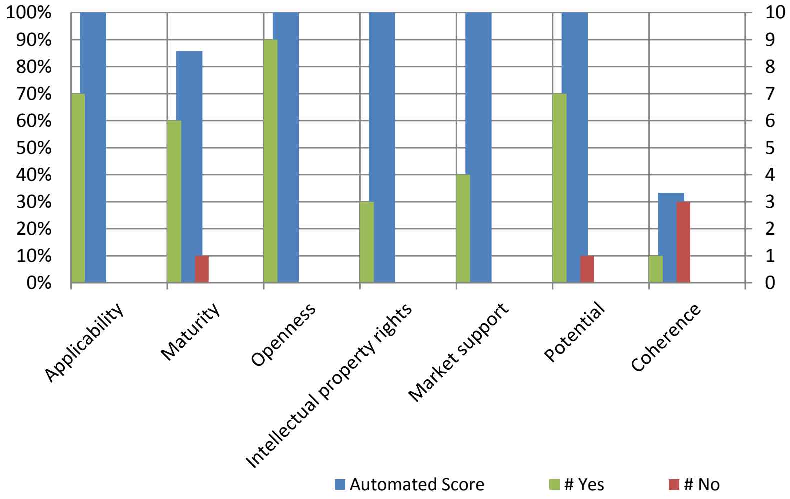


Figure 17 – Summary graph with the CAMSS assessment scores of the UBL TC e-Document engineering method

3.3. CEN BII method

This section contains a description and assessment of the e-Document engineering method of the CEN Business Interoperability Interfaces (BII) initiative according to the assessment categories in CAMSS.

3.3.1. Description

The Business Interoperability Interfaces (BII) initiative, established in May 2007 as a workshop under CEN, has used an e-Document engineering method that reuses and complements the methods of UN/CEFACT and the OASIS UBL Technical Committee. The mission of the BII workshop is to spread and facilitate the use of e-procurement standards by suppliers and buyers, and especially public administrations.

The first phase of the workshop resulted in CEN Workshop Agreement (CWA) 16073 that served as a basis for the implementations of initiatives such as PEPPOL, ePRIOR, and Open ePRIOR. Following on the successful adoption of the CWA, the second phase of the workshop was launched in 2010, resulting in 5 CWAs published in February 2013, capturing new and refining existing business requirements. A third phase, launched in March 2013 is ongoing at the time of writing[31]. The focus of BII is to collect European business requirements and to provide guidance for consistent implementation of these requirements utilizing existing international developments, i.e., from UN/CEFACT and OASIS UBL. Hence, the BII specifications are targeted at achieving organizational and semantic interoperability. They are neutral of any technical implementation and syntax. In order to provide implementable specifications however, the workshop has published additional specifications of how the models can be mapped to defined syntaxes.

The CEN BII e-Document engineering method is described in CWA 16558[32], with an overview in Annex A "**Guideline Profile Architecture**"[33]. The guideline describes how to define a *profile*, i.e., a set of *business processes* (usually only one) achieving a *business goal*. A business process, e.g., "Ordering", implies one or more *business transactions*, e.g., "Order submission", "Order rejection", "Order acceptance", etc. A business transaction is the exchange of an e-Document between the business partners.

CEN BII method consists of:

- A **methodology** defined in several guidelines within CWA 16558 Architecture. This CWA contains the following guidelines:
  - Guideline on Capturing of Business Requirements
  - Guideline on the Business Rules Description Mechanism
  - Guideline on Code List Management

[31] http://www.cenbii.eu/about/

[32] ftp://ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA16558_2013.pdf

[33] ftp://ftp.cen.eu/public/CWAs/BII2/CWA16558/CWA16558-Annex-A-BII-Guideline-ProfileArchitecture-V2_0_0.pdf

  - Guideline on Data Formats

- A **library of concepts** contained in each profile document. The business terms are described on a Profile basis, even if their semantic meaning is shared among the profiles.
- **Technical references** to produce validation artefacts, including:
  - Guideline on Syntax Binding Methodology
  - Guideline on Implementation and use of Validation Artefacts
  - Guideline on Conformance and Customizations

The figure below depicts the steps that are followed to create the CEN BII Profiles.

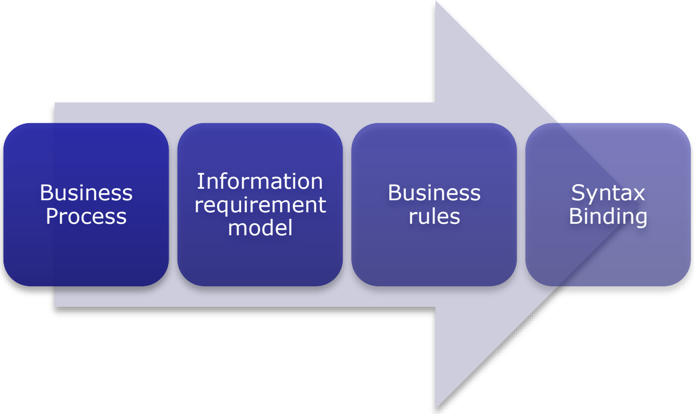


Figure 18 – CEN BII method

In the subsequent subsections, we describe the fit of the UBL method with the high-level requirements that are elicited in Chapter 2. These requirements are structured according to the following steps:

1. Requirement gathering;
2. Information modelling;
3. Business rule definition;
4. Syntax binding;
5. Schema production;
6. Governance; and
7. Conformance testing.

3.3.1.1. Requirement gathering

CEN BII defines a methodology to capture goals and gather requirements in the profiles. The following picture depicts how goals relate to requirements in the context of the CEN BII work:

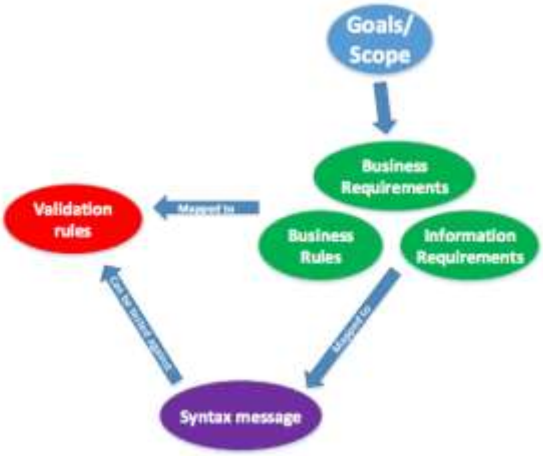


Figure 19 – Illustration of how requirements relate to goals, syntax and validation rules[34]

The list of goals is captured and agreed as a set of objectives on the profiles. Each objective has an identifier and an explanation. Based on these goals, the profile identifies the business requirements analysing the different business activities that have to be supported for this profile. The business requirements are identified and numbered with a unique identifier. They have the following metadata:

- Requirement identifier
- Group
- Requirement name
- Description of use
- Rationale
- Supporting questions (who needs it, what is needed, when is needed)
- Is this a core requirement
- Requirement source/driver to help categorizing it.

Using the list of goals and requirements, the business processes choreography is depicted using an UML activity diagram. A business process describes the behaviour of two business partners in order to achieve a common business goal.

The following table compares the fulfilment of the high level requirements when using the CEN BII engineering methods. After the assessment,

Table 23 – Comparing assessed methods when gathering requirements

ID	Description	CEN BII
HLR-1	Describe specific goals to be achieved with the exchange of electronic documents	Yes
HLR-2	Define the requirements. This can be done using techniques such as:	Yes, described using

[34] CWA 16558 – Annex B BII Guideline Capturing of Business Requirements

	- Assertions: high-level statements about the business transaction. - Use case scenarios using real-life examples to exemplify the business process flow; and - Business process choreography: a formal description of the interactions and exchange of information that takes place	assertions and key examples.
HLR-3	Traceability: goals and requirements should be individually identifiable and traceable. This means that requirements should be linked to goals.	Yes

The tools in CEN BII are focused on capturing the actual knowledge from the business participants. Therefore, they are non-technical to encourage participation of non-technical stakeholders.

CEN BII method has the following tools to produce the expected outcomes:

Outcome	CEN BII
A list of comprehensive goals described in text	Identified in each PDF document profile as a table with an identifier, the list of goals is maintained in a spreadsheet.
A set of real-life examples covering all the cases that have to be addressed using the exchange of e-Documents	Identified in each PDF document profile as a textual explanation
A list of textual requirements linked to the goals	Each PDF document profile has a specific list of requirements linked to goals, the list of requirements is maintained in a spreadsheet.
Graphical and textual description of the business process	Each PDF document profile has a UML Activity Diagram and textual description depicting the business process covered by the profile

3.3.1.2. Information modelling

CEN BII does not have any *formal* specification for information modelling. It implicitly requires a common conceptual vocabulary to be documented and linked to the requirements. Each concept in the vocabulary is linked to a business transaction.

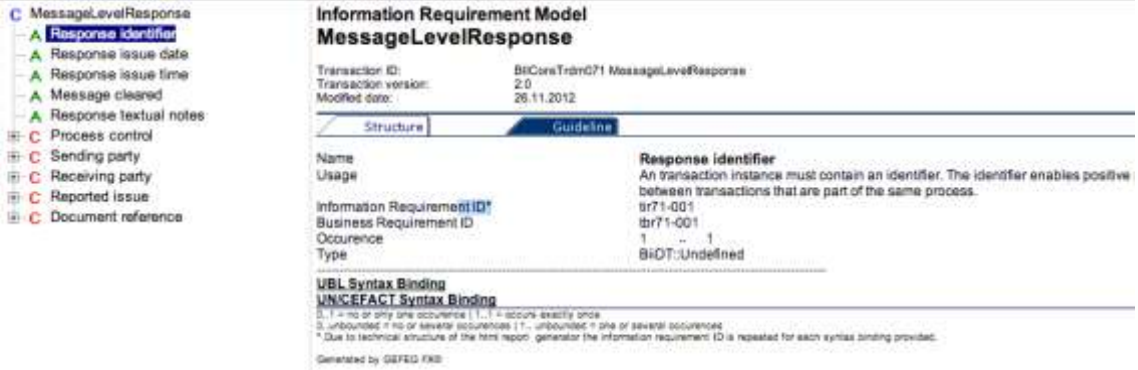
An information requirement model is built by defining the structure and relationship between several logical information elements, each of which is linked to a business requirement. An information requirement model can be seen as a synonym for a semantic data model.

The information requirement model should contain the minimum set of information elements such that each business transaction, i.e., exchange of e-Document, is useful and understandable in all business scenarios.

The CEN BII defines the information requirement models on a per transaction basis. Every transaction defines all the information requirements, establishing an identifier and a link to the business requirement identified in the previous step. The metadata used to define each business concept is as follows:

- Transaction identifier
- Transaction version
- Information requirement identifier
- Business requirement identifier it fulfils
- Date of modification
- Name of the information requirement
- Description of the information requirement
- Occurrence or cardinality
- Description of the data type according to the data types described in CEN BII.

The figure below depicts an html version of the information requirement model for the Message Level Response transaction from CEN BII.



Information Requirement Model

MessageLevelResponse

Transaction ID:	BiiCoreTrdm071 MessageLevelResponse
Transaction version:	2.0
Modified date:	26.11.2012

Name	Response identifier
Usage	An transaction instance must contain an identifier. The identifier enables positive between transactions that are part of the same process.
Information Requirement ID*	tir71-001
Business Requirement ID	tbr71-001
Occurence	1 .. 1
Type	BiiDT::Undefined

UBL Syntax Binding

UN/CEFACT Syntax Binding

0..1 = no or only one occurence | 1..1 = occurs exactly once

0..unbounded = no or several occurences | 1..unbounded = one or several occurences

* Due to technical structure of the html report generator the information requirement ID is repeated for each syntax binding provided.

Generated by GEFEG.FX®

Figure 20 – Information requirement model of CEN BII

CEN BII provides the conceptual information model, the semantics per each information element and the structure of the classes.

Table 24 – HLR using CEN BII for information modelling

ID	Description	CEN BII
HLR-4	Capture business terms in an information model describing the explicit semantics of every data element, its attributes, cardinalities and relationships.	The library of information requirements is captured and maintained with the definition in the vocabulary.
HLR-5	Describe relationships between information components and	Some information requirements are structured into classes or

	requirements.	objects with properties.
HLR-6	Depict the information model requirements using conceptual modelling languages such as UML, SBVR and repositories ISO11179 (Metadata Registry).	CEN BII does not require the information requirement model to be depicted in a conceptual modelling language such as UML, etc.
HLR-7	Identify and reuse semantics and concepts from standardized vocabularies.	The concepts and semantics were taken in CEN BII1 from the UBL Library. These concepts and definitions were improved in CEN BII2 according to EU legislation and European common practices.

CEN BII uses the following tools to produce the expected outcomes:

Outcome	CEN BII
A set of concepts with unique semantic definition covering all the elements in the information model	GEFEG.FX is used to capture the business terms with their unique semantic definition.
A set of objects or classes with properties and relationship between them, describing cardinalities	GEFEG.FX is used to inform about structures and cardinalities of the business concepts.
Optionally graphical representation of the information models	GEFEG.FX creates a class diagram per each object class.

3.3.1.3. Business rule definition

Business rules are used in CEN BII to define or constrain some aspects of the business. CEN BII provides a guideline to use the Semantics for Business Vocabulary and Rules (SBVR) standard.

BII distinguishes between two types of business rules: process rules and information constraints.

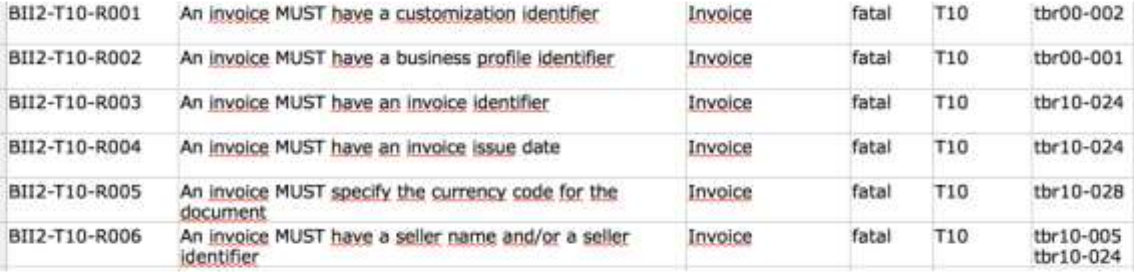
- **Process rules** control or influence the behaviour of the business partners. They may or may not affect the workflow in the process. Process rules typically express how the e-Document must be processed.
- **Information constraints** express what may appear in the e-Document. Typical rules include information element cardinalities, data element interactions (formulas, dependencies, relationships, etc.), and data element value restrictions. A validation artefact can be created in order to enforce the information constraints in the information requirement model.

SBVR, the "Semantics of Business Vocabulary and Rules" is an OMG standard, used to create a Business Semantics Glossary [Collibra, 2010]. It is the standard used by CEN BII to capture the abstract business rules for their information requirement models. In the syntax binding process, the abstract business rules are bound to a particular syntax.

The rules are captured in spreadsheets with the following metadata:

- A **rule identifier** used to identify each particular rule.
- The **rule statement**, a plain English statement of the rule, as mandated by SBVR, and using the vocabulary of terms as identified in the previous phase.
- A **rule context** describing where the rule applies
- The **severity** of the rule, being severity fatal or warning
- The **transaction identifier** for the business rule
- The **business requirements** from which the business rule is derived.

The figure below depicts an excerpt of the Invoice Transaction business rule set.



BII2-T10-R001	An invoice MUST have a customization identifier	Invoice	fatal	T10	tbr00-002
BII2-T10-R002	An invoice MUST have a business profile identifier	Invoice	fatal	T10	tbr00-001
BII2-T10-R003	An invoice MUST have an invoice identifier	Invoice	fatal	T10	tbr10-024
BII2-T10-R004	An invoice MUST have an invoice issue date	Invoice	fatal	T10	tbr10-024
BII2-T10-R005	An invoice MUST specify the currency code for the document	Invoice	fatal	T10	tbr10-028
BII2-T10-R006	An invoice MUST have a seller name and/or a seller identifier	Invoice	fatal	T10	tbr10-005 tbr10-024

Figure 21 – CEN BII set of abstract business rules

CEN BII methods to fulfil the high level requirements for capturing and defining business rules:

Table 25 – HLR using CEN BII methods for business rule definition

ID	Description	CEN BII
HLR-8	Identify the integrity constraints on the information model and describe them as business rules.	Captured in the information requirement model.
HLR-9	Define inferences and mathematical calculations that the e-Document elements must fulfil.	Defined as abstract business rules in spreadsheets.
HLR-10	Define conditional business rules and co-occurrence constrains that the e-Document elements must fulfil.	Defined as abstract business rules in spreadsheets.
HLR-11	Define sets of allowed values for coded data elements.	Defined coded values and code lists in spreadsheets.

CEN BII uses the following tools to produce the expected outcomes:

Outcome	CEN BII

List of business rules in plain language	Spreadsheet file containing the abstract business rules.
List of allowed code values for specific coded fields	Spreadsheet file containing the codes and the business rules associating codes to coded elements.

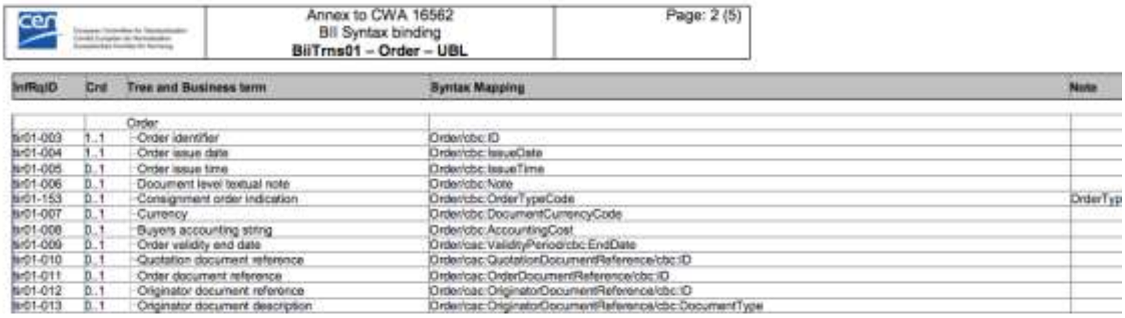
3.3.1.4. Syntax binding

Instead of creating a new e-Document format, the information requirement model is mapped to existing syntaxes.

The information requirement models are syntax neutral, implying that they could be bound to any existing syntax solution. The syntax mapping is often complex due to different granularity on the elements and slightly different semantics between the information requirement and the meaning of the component of the syntax. As a convention, CEN BII establishes that the semantics of the information requirement model overrides the semantics of the syntax.

CEN BII provides the relevant syntax mappings to UN/CEFACT XML and OASIS UBL. CEN BII does not provide any XML schemas, but relies on the XML Schemas published by UN/CEFACT and UBL.

CEN BII uses the GEFEG.FX tool to create and maintain the syntax bindings, mappings between the information requirement models and the syntaxes.



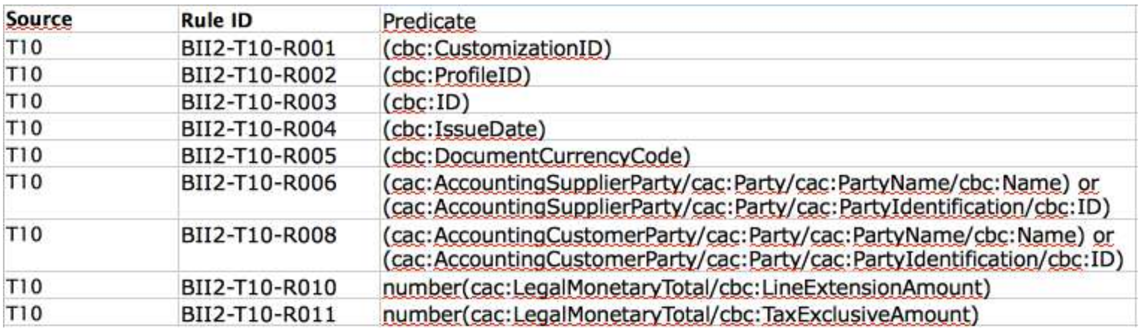
Annex to CWA 16562
BII Syntax binding
BiiTrns01 – Order – UBL

Page: 2 (5)

InfRqID	Crd	Tree and Business term	Syntax Mapping	Note
		Order		
tir01-003	1..1	Order identifier	Order/cbc:ID	
tir01-004	1..1	Order issue date	Order/cbc:IssueDate	
tir01-005	0..1	Order issue time	Order/cbc:IssueTime	
tir01-006	0..1	Document level textual note	Order/cbc:Note	
tir01-153	0..1	Consignment order indication	Order/cbc:OrderTypeCode	OrderTyp
tir01-007	0..1	Currency	Order/cbc:DocumentCurrencyCode	
tir01-008	0..1	Buyers accounting string	Order/cbc:AccountingCost	
tir01-009	0..1	Order validity end date	Order/cac:ValidityPeriod/cbc:EndDate	
tir01-010	0..1	Quotation document reference	Order/cac:QuotationDocumentReference/cbc:ID	
tir01-011	0..1	Order document reference	Order/cac:OrderDocumentReference/cbc:ID	
tir01-012	0..1	Originator document reference	Order/cac:OriginatorDocumentReference/cbc:ID	
tir01-013	0..1	Originator document description	Order/cac:OriginatorDocumentReference/cbc:DocumentType	

Figure 22 – CEN BII syntax binding the Order Information Requirement Model to UBL

Apart from the information model, the abstract business rules described in the previous sections have to be bound also to the syntax. Spreadsheets are used to capture the XPATH expressions required to create the Schematron validation artefacts binding business rules and code lists to the syntaxes.



Source	Rule ID	Predicate
T10	BII2-T10-R001	(cbc:CustomizationID)
T10	BII2-T10-R002	(cbc:ProfileID)
T10	BII2-T10-R003	(cbc:ID)
T10	BII2-T10-R004	(cbc:IssueDate)
T10	BII2-T10-R005	(cbc:DocumentCurrencyCode)
T10	BII2-T10-R006	(cac:AccountingSupplierParty/cac:Party/cac:PartyName/cbc:Name) or (cac:AccountingSupplierParty/cac:Party/cac:PartyIdentification/cbc:ID)
T10	BII2-T10-R008	(cac:AccountingCustomerParty/cac:Party/cac:PartyName/cbc:Name) or (cac:AccountingCustomerParty/cac:Party/cac:PartyIdentification/cbc:ID)
T10	BII2-T10-R010	number(cac:LegalMonetaryTotal/cbc:LineExtensionAmount)
T10	BII2-T10-R011	number(cac:LegalMonetaryTotal/cbc:TaxExclusiveAmount)

Figure 23 – CEN BII XPATH expressions for abstract business rules

Table 26 – HLR using CEN BII for syntax binding

ID	Description	CEN BII
HLR-12	Map the information model to a standard format where this format fulfils most of the goals and requirements of the project.	Yes
HLR-13	Create a usage guideline on the syntax for implementers.	Yes
HLR-14	Create validation artefacts for business rules and code lists.	Yes
HLR-15	List minor gaps and/or requirements that cannot be fulfilled using the selected syntax.	Yes

The outcomes of this phase are created using the following tools:

Outcome	CEN BII
Mapping from the information model to the syntax	GEFEG.FX is used to create the mappings. From that tool RTF files containing the mappings can be produced.
Validation artefacts including business rules and code lists	An open-source tool[35] can be used to produce the Schematron artefacts.
Guidelines on the usage of the syntax	GEFEG.FX is used to produce usage guidelines.

3.3.1.5. Schema production

CEN BII does not create schemas.

[35] https://github.com/oriol/b2btoolkit

3.3.1.6. Governance

Table 27 – HLR using CEN BII for Governance

ID	Description	CEN BII
HLR-19	Set up a governance organization for long-term sustainability of the e-Document format.	A CEN Workshop Agreement is not enough to provide long-term sustainability of an e-Document engineering methodology. It should be leveraged to a CEN European Norm.
HLR-20	Define the maintenance and versioning mechanisms for the e-Document format.	The CWA defines a maintenance mechanism of the e-Document formats.
HLR-21	Establish policies and procedures for the use and maintenance of the e-Document format.	A CWA provides policies and procedures for the use of the e-Document format.

3.3.1.7. Conformance testing

CEN BII has implemented a web site with validation artefacts and a tool to validate instances for conformance testing.

The validation artefacts and the on-line validation tool are non-normative, but they can be downloaded and used by implementers.

ID	Description	CEN BII
HLR-22	Define a testing framework for the e-Document instances.	Yes

The outcomes of this phase should be:

Outcome	CEN BII
Conformance Testing Procedure	The testing artefacts can be downloaded from http://spec.cenbii.eu/BII2/Tools/bii2-download.html

3.3.2.Summary of CAMSS assessment

The CAMSS spreadsheet template allows for the calculation of an automated score for the CEN BII method. This score is summarised in the below table. In Annex III.1.3, an overview of the CAMSS assessments is presented for the CEN BII method.

Table 28 – CAMSS Assessment Score CEN BII Method

Category	Automated Score [#Yes/(#Yes + #No)]	CAMSS Assessment Strength [(#Yes + #No)/#Criteria per category]
Applicability	100%	88%
Maturity	71%	100%
Openness	89%	100%
Intellectual property rights	100%	100%
Market support	50%	80%
Potential	50%	62%
Coherence	25%	67%
Overall score:	69%	85%

Figure 24 refers to the CAMSS assessment score of the CEN BII method. It represents the automated score (%), as well as the amount of "yes" and "no" answers per category of CAMSS criteria.

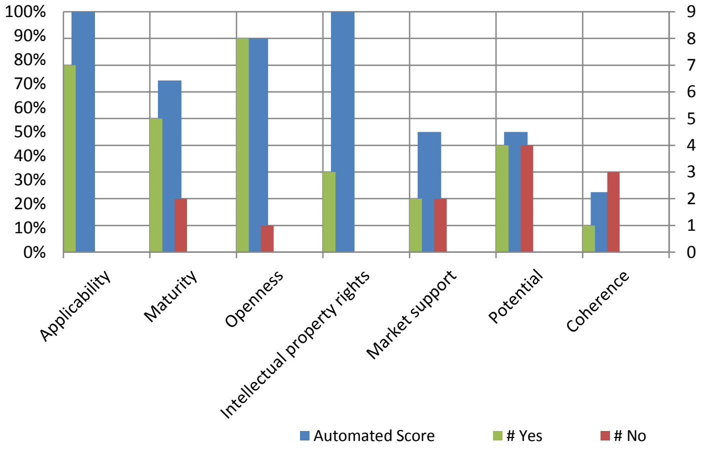


Figure 24 – Summary graph with the CAMSS assessment scores of the CEN BII e-Document engineering method

4. MINI-PILOT ON E-DOCUMENT ENGINEERING

This chapter describes a mini-pilot on e-Document engineering that was carried out in the context of Action 2.15 of the ISA Programme in the period February – March 2014. The pilot was conducted in collaboration with the e-SENS WP 6.2 Competence Cluster on Semantics, processes and Documents. The objectives of the mini-pilot are:

1. To demonstrate the use of a standard e-Document engineering method to create e-Document formats; and
2. To demonstrate how a metadata registry, a *common library of data elements* and *mappings,* can help e-Document engineering.

The pilot has an approach that is depicted in Figure 25:

1. **Create e-Document formats using a standard e-Document engineering method** in close connection with the metadata registry. This is described in Section 4.2. As one of the objectives is to demonstrate how to resolve syntactical interoperability conflicts, it is convenient to use an engineering method not tightly coupled to a particular syntax. For that purpose, we propose to use the **CEN BII e-Document engineering method** to conduct the mini-pilot which caters for syntax bindings to several libraries of data elements. This includes requirements gathering, information modelling and business rule modelling. As there is no standard e-Document to which we will be able to create a mapping to, we will create a new e-Document following the UBL naming and design rules reusing existing libraries of data elements, including the ISA Core Vocabularies.

    - **Requirements gathering**: we use the CEN BII spreadsheets template to gather the goals and requirements for the pilot. We use BPMN to depict the activity diagram of the business process to be covered. All requirements are added to the metadata registry.
    - **Information modelling**: We use a spreadsheet to capture the information requirements for our pilot. The spreadsheet follows the template from CEN BII to capture the information requirements. The information requirements can be depicted using a UML class diagram. All information models are added to the metadata registry.
    - **Business rule definition**: We use the CEN BII spreadsheet template to collect and identify the business rules and their associated syntax binding. All business rules are added to the metadata registry.
    - **Schema production**: We use a script from Crane Software (Genericode-to-UBL-NDR) to demonstrate how the information requirements (business terms) are converted into syntax components. The script will take information as input that is retrieved from the metadata registry.

2. **Create a metadata registry,** a common library of data elements and mappings, for some standard libraries. This is described in Section 4.3. We describe data elements with a uniform vocabulary based on the ISO 11179-3 meta-model. The metadata model is implemented in RDF Schema which

allows flexible integration with other information about the structural metadata included in data standards. The library includes data elements from the UBL Common Library, the W3C Registered Organization Vocabulary and the ISA Core Location Vocabulary. For each data element we record its identifier, definition, and representation. We demonstrate that the metadata registry supports the following use cases:

- Describe libraries of data elements, create links between them, and search for data elements; and
- Describe requirements of e-Document formats, facilitate syntax binding or schema creation, and enrich XML Schema documentation.

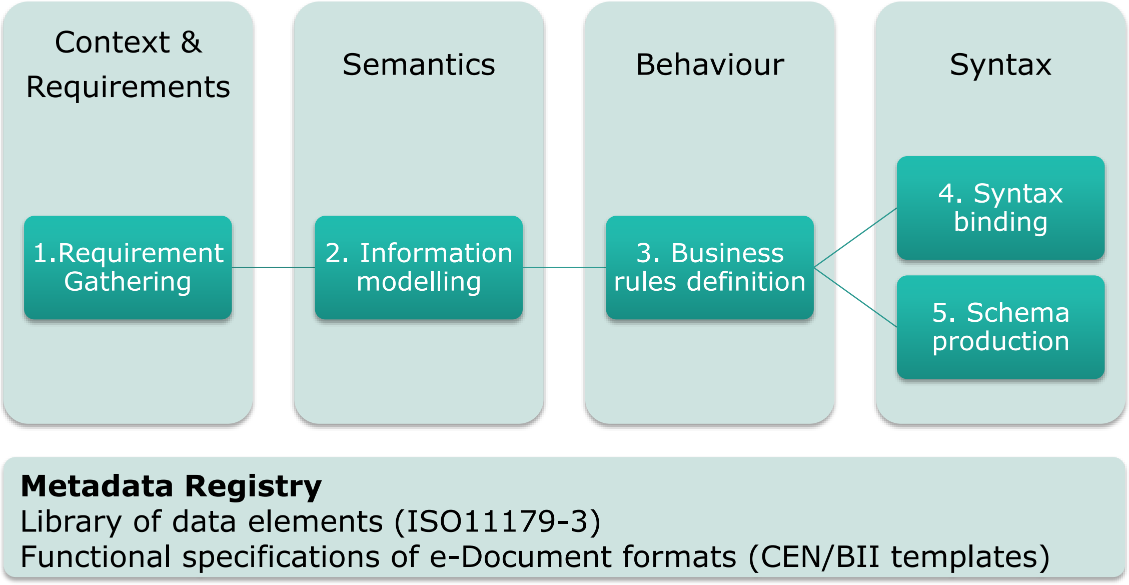


Figure 25 – Approach to the pilot

4.1. Running example: business activity registration

The mini-pilot is based on a use case and more elaborate pilot proposed by e-SENS 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 only pilot the use of the e-Document engineering method to generate e-Document specifications. The actual Activity Registration pilot will be conducted later on by the e-SENS large-scale pilot. The Activity Registration pilot will allow 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 e-SENS pilot intends to simplify the procedure for both the business as well as the authority. Such administrative simplification using the Points of Single Contact

(PSC) is targeted by the Directive 2006/123/EC of 12 December 2006 on services in the internal market[36].

Concerning the e-Document engineering aspect of the mini-pilot of activity registration, we will follow the blueprint depicted in Figure 1. The CEN BII spreadsheet models will mainly be used to gather the information.

4.2. e-Document engineering

4.2.1.Requirements gathering

The first step is to precisely 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.

- **Goals**: The specific goals that need to be achieved with the exchange of e-Documents in the context of the activity registration pilot must first be described. Table 29 gives an overview of the identified goals in the context of this mini-pilot.

Table 29 – Example goals for the activity registration 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

- **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 30.

Table 30 – Scope statement of the activity registration pilot

Scope statement

[36] Official Journal of the European Union, L 376, 27 December 2006

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.

The scope of the pilot can also be depicted in a BPMN diagram, as shown in Figure 26.

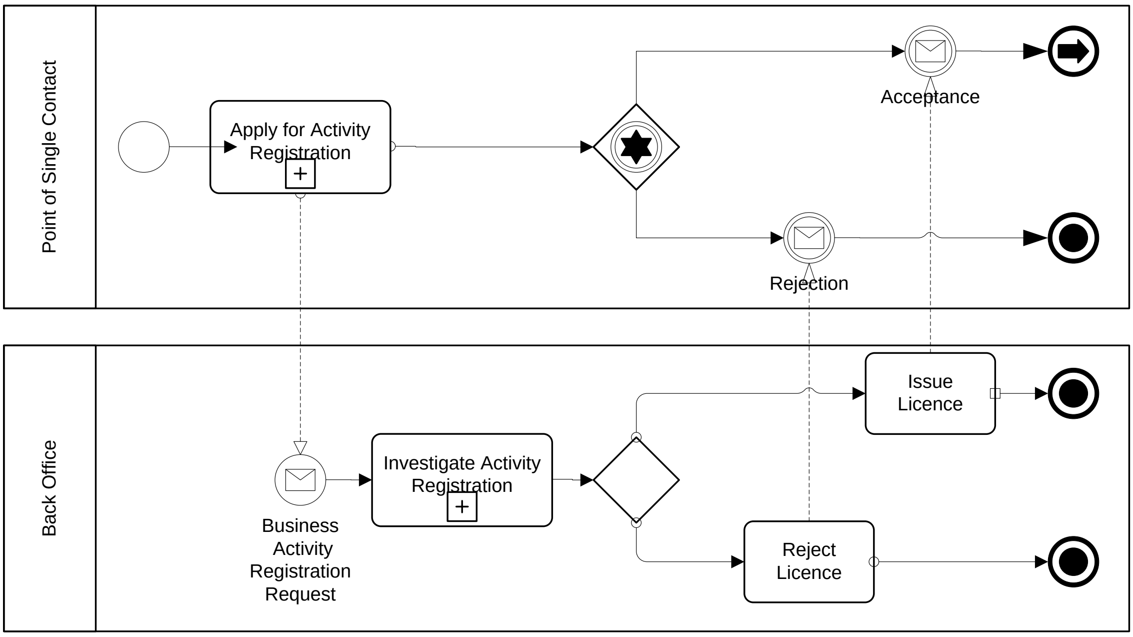

Point of Single Contact
Apply for Activity Registration
Acceptance
Rejection
Back Office
Business Activity Registration Request
Investigate Activity Registration
Issue Licence
Reject Licence

Figure 26 – BPMN diagram: activity registration pilot

- **Key examples**: By means of key examples concerning activity registration given in Table 31, a real-life scenario is represented to give a description of the business process flow.

Table 31 – Key examples of activity registration (provided by e-SENS WP5)

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 may be required an additional administrative verification process..
	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.

- **Specific requirements**: Finally, once the goals, scope and key examples have been identified, the specific requirements that e-Documents must fulfil can be gathered. For the mini-pilot, the specific requirements related to the goals are outlined in Table 32.

Table 32 – High level requirements of the activity registration pilot

Require-ment identifier	Require-ment 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	The business request has to be uniquely	G1, G2, G3

		be identified	identifiable, with information about its issuance.	

4.2.2.Information modelling

The information modelling phase identifies and describes the information to be exchanged in e-Documents according to the requirements specified in Table 32 – High level requirements of the activity registration pilot.

The information modelling covers the following:

- The business terms in an information model describing the explicit semantics of every data element in terms of its attributes and cardinalities.
- The relationships between information components and requirements.
- The information model requirements depicted by a conceptual modelling language (ISO11179 MDR).
- The reuse of semantics and concepts from standard vocabularies.

Table 33 lists three information requirements, being the Business Activity, the Business Name and the Business Legal Form.

Table 33 – Information Requirements of the activity registration pilot

IR ID	**IR 4**	**IR 5**	**IR 6**
Business Term Name	Business Activity	Business Name	Business Legal Form
Usage	Activity performed by the legal entity, which is requested for registration	Name of the legal entity that is requesting the business activity registration	Type of the legal entity that is requesting the business activity registration
Refer to Business Requirement ID	R3	R1	R1
Refer to Business Rule ID	BR1		BR2
Cardinality	1..1	1..1	1..1
Concept location	Registered Organization Vocabulary	Registered Organization Vocabulary	Registered Organization Vocabulary
Standard	Organization Activity	Legal Name	Organization Type

Concept Name			
Concept Description	The activity of an organization should be recorded using a controlled vocabulary. Several vocabularies exist, many of which map to the UN's ISIC codes. The proposed choice for European interoperability is NACE[37].	The legal name of the business. A business might have more than one legal name, particularly in countries with more than one official language.	Familiar company types are SA, PLC, LLC, GmbH, etc. At the time of publication, there is no agreed set of company types that crosses borders. Each jurisdiction needs a limited set of recognized company types and these should be expressed in a consistent manner.

4.2.3.Business rule definition

In 4.2.2 Information modelling, the business terms and facts of the pilot were described. However, there are still action **assertions**, **constraints** and **derivations** concerning some aspects of the e-Document. These business rules are described according to the high-level requirements and information modelling requirements.

Business rules may include the following:

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

We have identified two business rules that are related to the Business Activity and the Business Legal Form in Table 34.

Table 34 – Business rules for activity registration

Business Rule ID	Rule	Refer to Information	Refer to High Level	Error level

[37] **NACE** is the **Statistical Classification of Economic Activities in the European Community** (in French: *Nomenclature statistique des activités économiques dans la Communauté européenne*), a European industry standard classification system consisting of a 6 digit code. http://ec.europa.eu/competition/mergers/cases/index/nace_all.html

		Requirements	Requirements	
BR1	The business activity must refer to a NACE activity	IR4	R3	Fatal
BR2	The legal form of the business must be recognized by the business' country of origin	IR6	R1	Fatal

4.2.4.Syntax binding or schema production

Syntax binding is one of the options to produce physical artefacts in order to help developers implement the e-Documents according to the e-Document format rules. With syntax binding, the information requirement model is mapped to an **existing** syntax model and the usage guidelines are specified.

However, syntax binding is not applicable for the activity registration pilot.

The second option is to **produce** a **new** e-Document format. This option is suitable for the activity registration pilot, as there are no recognized international standards for the industry and business process the project is targeting.

The schema production includes the following process:

- Common information model components are mapped to available Common Vocabulary schemas;
- A new e-Document format is created using a standard NDR to automate the schema production; and
- Validation artefacts for business rules and code lists are created.

In order to produce a new e-Document format, we have used a set of open source tools provided by Crane Softwrights to build a UBL-like schema for the new e-Document following the UBL Naming and Design Rules. Annex I contains a tutorial with a more detailed descriptions of the steps performed and links to all files mentioned here.

We have followed these steps to produce the schema[38]:

1. We have created an OpenOffice spreadsheet following the UBL metadata and according to the information requirement model created in the previous phase. The information requirements have to be transferred to this new spreadsheet.

   The OpenOffice file has two different sheets. One sheet defines the new e-Document and the other sheet with the ISA Core Vocabulary aggregate business information entities used in the project.

[38] See Annex I Tutorial on e-Document engineering for a complete reference

The process of moving from the information requirement model to the UBL spreadsheet requires:

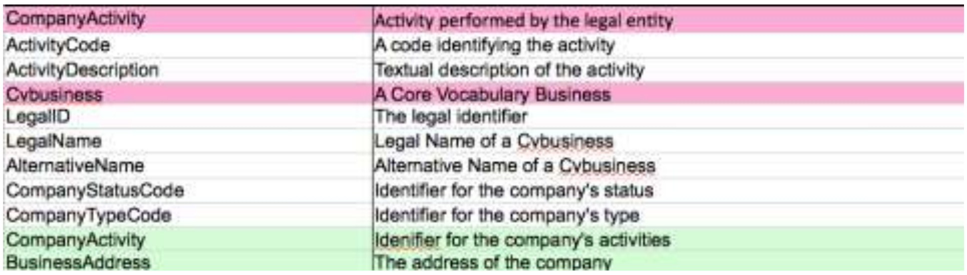
- defining the simple information requirements as properties of the new e-Document,
- identifying reusable components from the ISA Core Vocabulary and UBL common aggregate libraries and
- creating new aggregates when needed (in our case there will be no additional aggregates needed).

We have filled the spreadsheet for the new e-Document with:

- The Dictionary Entry Name identifying the object class (in UBL a document type is an ABIE), the property term and the representation term
- The cardinality as defined in the information requirement,
- The definition as described in the information requirement.

In the process of creating the spreadsheet, we have grouped some concepts from the information requirement model in groups or aggregates:

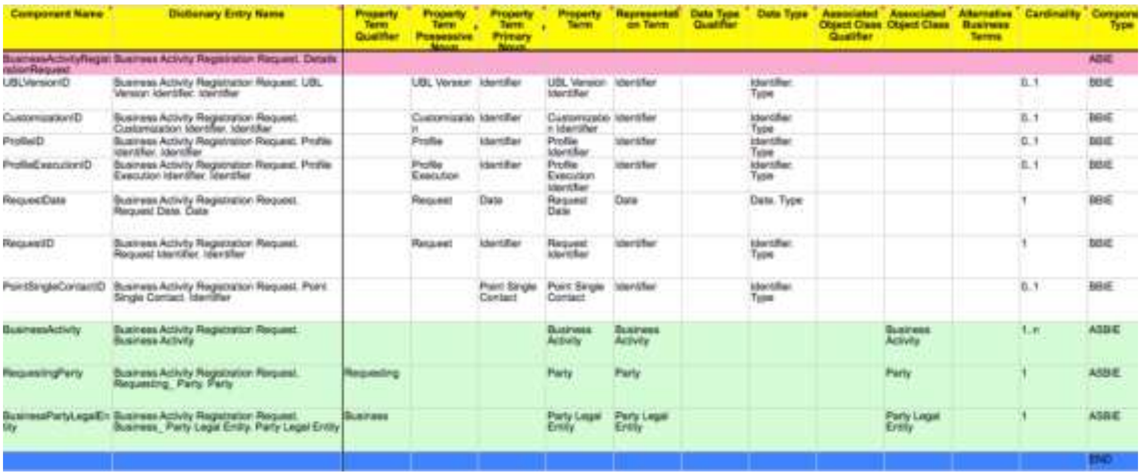
- For the **Business Activity** we have reused an object class from the ISA Core Vocabulary called Business Activity where there are two information entities, the Activity Code and the Activity Description.



CompanyActivity	Activity performed by the legal entity
ActivityCode	A code identifying the activity
ActivityDescription	Textual description of the activity
Cvbusiness	A Core Vocabulary Business
LegalID	The legal identifier
LegalName	Legal Name of a Cvbusiness
AlternativeName	Alternative Name of a Cvbusiness
CompanyStatusCode	Identifier for the company's status
CompanyTypeCode	Identifier for the company's type
CompanyActivity	Idenifier for the company's activities
BusinessAddress	The address of the company

Figure 27 – ISA Core Vocabulary aggregated business information entity

- For the Requesting Party, we have decided to reuse the UBL Party class.
- For the Business Legal Form we have decided to reuse the ISA Core Vocabulary Cvbusiness object class.

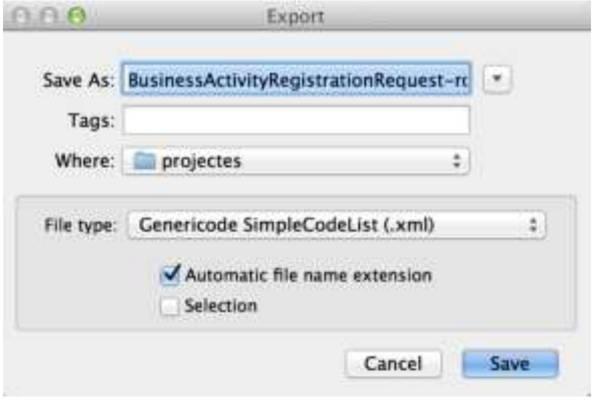


Component Name	Dictionary Entry Name	Property Term Qualifier	Property Term Possessive Noun	Property Term Primary Noun	Property Term	Representation Term	Data Type Qualifier	Data Type	Associated Object Class Qualifier	Associated Object Class	Alternative Business Terms	Cardinality	Component Type
BusinessActivityRegistrationRequest	Business Activity Registration Request. Details												ABIE
UBLVersionID	Business Activity Registration Request. UBL Version Identifier. Identifier		UBL Version	Identifier	UBL Version Identifier	Identifier		Identifier. Type				0..1	BBIE
CustomizationID	Business Activity Registration Request. Customization Identifier. Identifier		Customization	Identifier	Customization Identifier	Identifier		Identifier. Type				0..1	BBIE
ProfileID	Business Activity Registration Request. Profile Identifier. Identifier		Profile	Identifier	Profile Identifier	Identifier		Identifier. Type				0..1	BBIE
ProfileExecutionID	Business Activity Registration Request. Profile Execution Identifier. Identifier		Profile Execution	Identifier	Profile Execution Identifier	Identifier		Identifier. Type				0..1	BBIE
RequestDate	Business Activity Registration Request. Request Date. Date		Request	Date	Request Date	Date		Date. Type				1	BBIE
RequestID	Business Activity Registration Request. Request Identifier. Identifier		Request	Identifier	Request Identifier	Identifier		Identifier. Type				1	BBIE
PointSingleContactID	Business Activity Registration Request. Point Single Contact. Identifier			Point Single Contact	Point Single Contact	Identifier		Identifier. Type				0..1	BBIE
BusinessActivity	Business Activity Registration Request. Business Activity				Business Activity	Business Activity				Business Activity		1..n	ASBIE
RequestingParty	Business Activity Registration Request. Requesting_ Party. Party	Requesting			Party	Party				Party		1	ASBIE
BusinessPartyLegalEntity	Business Activity Registration Request. Business_ Party Legal Entity. Party Legal Entity	Business			Party Legal Entity	Party Legal Entity				Party Legal Entity		1	ASBIE
													END

Figure 28 – New main e-Document spreadsheet

We have also reused the UBL Version and document metadata basic information entities such as the Customization Identifier and the Profile Identifier, commonly used in UBL Schemas.

2. Once we have completed the spreadsheet model, we have used the OpenOffice spreadsheet to genericode subset[39] export filter from Crane Softwrights to produce a Genericode file directly from the OpenOffice document.



Save As: BusinessActivityRegistrationRequest-rc

Tags:

Where: projectes

File type: Genericode SimpleCodeList (.xml)

☑ Automatic file name extension

☐ Selection

Figure 29 – Export the spreadsheet to Genericode

3. The export function creates Genericode XML files. The Genericode to UBL NDR[40] script has been used to generate the new e-Document XSD from the Genericode XML.

   This tool creates a main XSD schema for the new e-Document, an XSD for the ISA Core Vocabularies common aggregate business information entities in the spreadsheet and another XSD with the ISA Core Vocabulary basic information entities.

[39] http://www.cranesoftwrights.com/resources/ubl/#gcExportSubset

[40] http://www.cranesoftwrights.com/resources/ubl/#gc2ublndr

The properties that are already defined in the UBL library are reused in the new XSD schema.

4.3. Using a metadata registry to support e-Document engineering

Problem/need: To improve interoperability of e-Document formats, common data elements should be reused as much as possible. Standard libraries of data elements, such as the Core Vocabularies or UBL, provide building blocks that enable such reuse. However, there is little or no convergence between libraries. Their reuse is hampered by the need to understand how each library is structured. Similarities between libraries are rarely described.

A different problem arises when documenting e-Document formats. Links between data elements and the requirements that led to the inclusion of the data elements provide valuable information for users to understand how to use an e-Document format. However, such information is often lost in the documentation of an XML Schema.

Solution: While the two problems described above may seem unrelated, they can be solved together by setting up a metadata registry. The registry contains uniform descriptions of data elements both from standard libraries and from specific e-Document formats, as well as the requirements of e-Document formats. Data elements are described with a model based on the ISO 11179-3 standard.

We propose to implement the metadata registry using semantic technologies, i.e., RDF. Each data element and each requirement is identified by a unique URI that can be resolved to get more information about the element or requirement. The registry can also contain links between resources, e.g., between similar data elements of different libraries, or between a data element of an e-Document format and its underlying requirements.

The metadata registry has the following use cases, which are described in further detail in the subsequent subsections:

1. **Manage libraries of data elements:**
    a. **Register data elements in standard libraries** in a central point of access. By leveraging the ISO 11179-3 standard, data elements of any library can be described in a coherent way.
    b. **Create links between classes and properties**, providing insight into the similarities and differences between libraries. Links facilitate creating mappings between e-Document formats created with different libraries and naming and design rules.
    c. **Search for data elements**: explore the use of classes and properties in different contexts, facilitating their reuse in similar contexts.
2. **Support of e-Document engineering:**

a. **Register requirements**, information model, and business rules in the context of an e-Document specification and create links between them. Links allow for easy browsing through the requirements, enabling a user to quickly find out the rationale hidden behind a data element or business rule.

b. **Facilitate syntax binding / schema creation**. By combining data element libraries and e-Document requirements in a central place, all information needed to reuse or produce schemas are readily available.

c. **Enrich XML Schema documentation of e-Document formats**. Because every data element and every requirement has a unique URI, links can be included as documentation inside the XML Schema. Through these links, users can acquire a better understanding of the intended use and rationale of a particular element inside an e-Document format.

Benefits: The metadata registry proposed in this section provides the following benefits:

- **Enhanced discovery of reusable data elements** in standard libraries. By using a common model to describe data elements of different standard libraries, the user can understand libraries more easily as she does not have to learn the specific structure and documentation of each library.
- **Convergence of e-Document formats** through higher reuse of existing data elements. The metadata registry contains uniform descriptions of data elements both from different standard libraries and from e-Document format. New e-Document formats can thus reuse elements from both standard libraries and other formats.
- **Expansion of standard libraries** by discovering popular new data elements. Maintainers of standard libraries can use the metadata registry to discover new data elements shared by different e-Document formats. Such data elements make good candidates for inclusion in the libraries.
- **Traceability of data elements** in e-Document formats by providing links between data elements and the underlying requirements. By navigating those links, one can trace back a data element all the way up to the initial goal. Such traceability provides valuable documentation with little development overhead.
- **Easily accessible and linkable** through web standards. By leveraging standard semantic web technologies, the metadata registry makes every data element easily accessible through standard web protocols. As each data element is identified by a unique URI, it can be easily referred to from within documentation or the description of other elements. Users get a detailed description of the data element by simply visiting the URI.

4.3.1.The metadata registry model

The metadata registry brings together the following information:

- Descriptions of data elements that are used in various standard e-Document formats.
- The goals, high-level requirements, information requirements, and business rules defined following our e-Document engineering methodology.

By bringing both together, we can make links between requirements and data elements, allowing for better documentation. We will use RDF to represent the metadata and links. We provide an ontology[41] for describing the elements of the Metadata Registry.

The model to represent data elements is inspired by the ISO 11179-3 Metadata Registry standard[42]. The ISO 11179-3 meta-model also serves as a base for UN/CEFACT CCTS and UBL. The model aims to provide a common description of data elements independently of their representation technique (XML or RDF).

As shown in the example of Figure 30, a **data element** (a BBIE or ASBIE in UBL as described in Section 3.2.1.2) consists of an object class and a property. An **object class** (an ABIE in UBL) is basically a set of data elements. It corresponds to an RDFS/OWL class or to an XSD complex type. A **property** is a reusable attribute, corresponding to an RDFS/OWL property or to an XSD element. There exist two kinds of properties: **basic properties** that take scalar values of a particular **data type**, and **association properties** that represent associations with other object classes.

Note that properties are not required to have definitions. For example, the UBL Common Library only provides definitions for object classes and data elements. The meaning of a property is thus inferred by its uses in different contexts (i.e., data elements).

Data elements, object classes, properties, and data types from different libraries may be linked together using SKOS mapping relations like “exact match”, “close match”, “broad match”, and “narrow match”. Doing so helps bridging libraries and facilitates the creation of mappings between e-Document formats constructed with different libraries and different naming and design rules.

[41] http://mdr.semic.eu/def

[42] http://metadata-standards.org/11179/

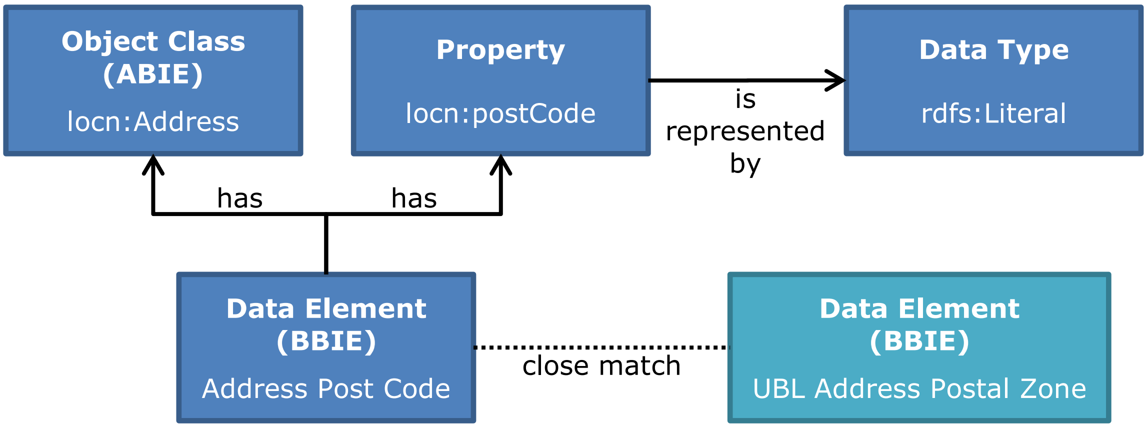


Figure 30 – The "Address Post Code" data element from the Core Location Vocabulary is defined by its object class and its property. The data element is also linked with a similar data element from the UBL Common Library.

The model to represent the requirements of an e-Document format stems directly from Section 4.2. Figure 31 shows the class diagram of the model. Each transaction is mapped to one object class, representing the concrete e-Document format. Information requirements are mapped to data elements. Thanks to these links, one can follow the path from an element in an e-Document all the way up to the initial goals.

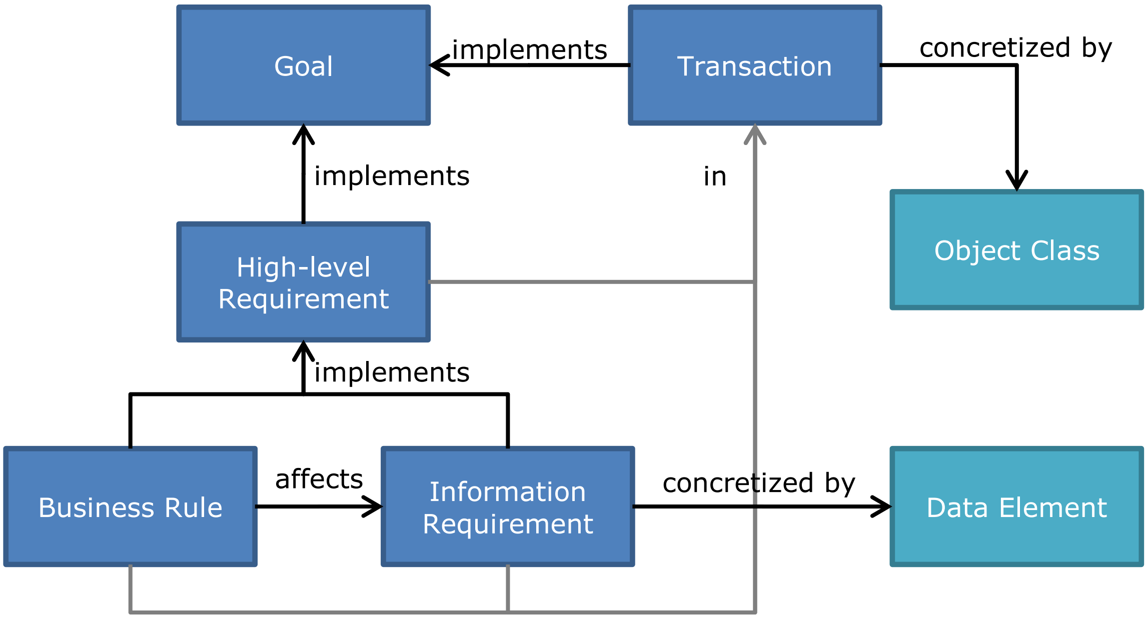


Figure 31 – Class diagram for the e-Document engineering method. Transactions and information requirements are linked to concrete object classes and data elements.

All the elements described above are linked to a **context**. A context is either a library of generic data elements, e.g., UBL, Core Location Vocabulary, or Registered Organization Vocabulary, or an e-Document format, e.g., e-SENS Activity Registration.

4.3.2.RDF implementation of the meta-model

Thanks to its flexible and open nature, the Resource Description Framework (RDF) is a good candidate for implementing the ISO 11179-3 meta-model. For example, the semanticMDR tool[43] of the SALUS project[44] provides a web front-end for managing data elements stored in RDF format. However, as the used ontology is a one-to-one mapping of the ISO 11179-3 meta-model, it introduces complex constructs that are not natural in RDF. Hence, we propose to use a simplified ontology, available at http://mdr.semic.eu/def. The Metadata Registry is available on http://mdr.semic.eu/.

The "Address Post Code" example of Figure 30 is represented as follows using the Turtle syntax:

```
<class/Address> a mdr:ObjectClass ;
    mdr:hasURI locn:Address ;
    rdfs:label "Address"@en ;
    skos:definition "An address representation as defined in the data specifications
of the EU INSPIRE Directive. The locn:addressId property may be used to link this
Address to other representations."@en ;
    skos:closeMatch ublclass:Address .

<property/postCode> a mdr:Property ;
    mdr:hasURI locn:postCode ;
    rdfs:label "post code"@en ;
    mdr:representation rdfs:Literal ;
    skos:closeMatch ublprop:PostalZone .

<element/AddressPostCode> a mdr:DataElement ;
    mdr:objectClass <class/Address> ;
    mdr:property <property/postCode> ;
    skos:definition "The post code (a.k.a postal code, zip code etc.). Post codes are
common elements in many countries' postal address systems."@en ;
    skos:closeMatch ublelem:AddressPostalZone .
```

The ontology also allows describing all the steps of the e-Document engineering methodology. An excerpt of the pilot is represented as follows using the Turtle syntax:

```
<goal/G1> a mdr:Goal ;
    rdfs:label "Improve Business Process Performance"@en ;
    rdfs:comment "To simplify the business activity registration procedure both for
the businesses and competent authorities"@en .

<transaction/T1> a mdr:Transaction ;
    rdfs:label "Business Activity Request"@en ;
    rdfs:comment "The request for the registration of a business activity by the
business person"@en ;
    mdr:implements <goal/G1>, <goal/G2>, <goal/G3>, <goal/G4> ;
    mdr:concretizedBy <class/BusinessActivityRegistrationRequestType> .

<requirement/R3> a mdr:HighLevelRequirement ;
    rdfs:label "Business activity"@en ;
    skos:definition "The business activity to be registered has to be identified"@en
;
```

[43] https://github.com/srdc/semanticMDR

[44] http://www.salusproject.eu/

```
    mdr:rationale "The receiving PSC has to know for which business activity the
requester is registering for."@en ;
    mdr:transaction <transaction/T1> ;
    mdr:implements <goal/G1>, <goal/G2> .

<ir/IR4> a mdr:InformationRequirement ;
    rdfs:label "Business activity"@en ;
    skos:definition "Textual description of the activity performed by the legal
entity that is requested for registration"@en ;
    mdr:transaction <transaction/T1> ;
    mdr:implements <requirement/R3> ;
    mdr:concretizedBy <element/BusinessActivityRegistrationRequestCompanyActivity> .

<br/BR1> a mdr:BusinessRule ;
    skos:definition "The business activity must refer to a NACE activity"@en ;
    mdr:transaction <transaction/T1> ;
    mdr:implements <requirement/R3> ;
    mdr:affects <ir/IR4> .
```

4.3.3. URI naming and design rules

When identifying resources in the metadata registry, we will use the following URI pattern:

```
http://mdr.semic.eu/id/{namespace}/{concept}/{reference}
```

The `{concept}` component represents the element type of the model. Table 35 shows the possible values. The `{namespace}` component allows categorizing the elements. Each library will have its own namespace, as well as each e-Document format. At last, the `{reference}` component is the name of the element.

Table 35 – Values for the concept component in URIs

Concept	Description
class	Object Class
property	Property
element	Data Element
datatype	Data Type
goal	Goal
transaction	Transaction
requirement	High-level Requirement
ir	Information Requirement
br	Business Rule

4.3.4. Use Case 1 – Register data elements in standard libraries

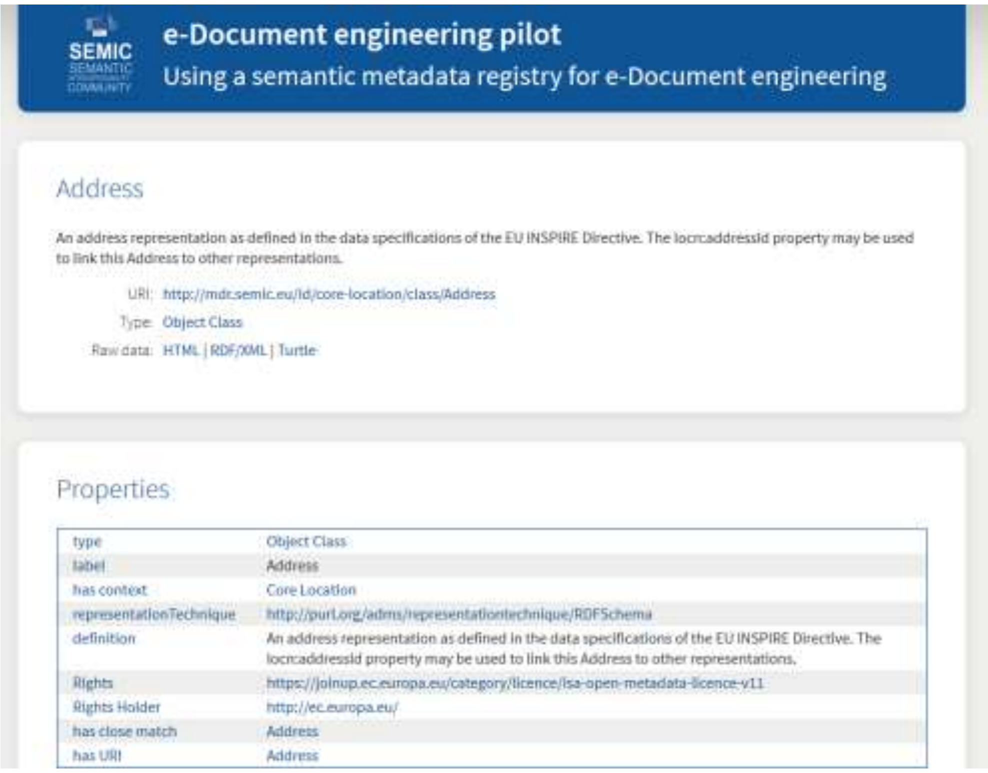
To provide uniform descriptions of data elements in standard libraries, the maintainer of a library (or the registry maintainer) publishes an RDF description of the library using the model shown in Section 4.3.1. If the library is already described in a structured manner, the description can be transformed to the metadata registry model in an automated way.

As examples, we have described the data elements of the following libraries in the metadata registry:

- **OASIS UBL 2.1 Common Library**: The description of the UBL Common Library was generated automatically from UBL's XML Schema files with a Python script[45]. An Aggregate Business Information Entity (ABIE) is represented as an object class. Basic Business Information Entities (BBIEs) and Association Business Information Entities (ASBIEs) are data elements. URIs for object classes and properties are taken respectively from the identifiers for xsd:complexType and xsd:element.
- **ISA Location Core Vocabulary**: The description of the Core Location Vocabulary was done by transforming the RDF Schema description by hand.
- **W3C Registered Organization Vocabulary**: The description of the Registered Organization Vocabulary was done by transforming the RDF Schema description by hand.

The object classes, properties, and data elements of the libraries can then be browsed through the metadata registry web interface. Figure 32 shows the description of the Address class from the ISA Core Location Vocabulary.

[45] https://github.com/SEMICeu/mdr/blob/master/scripts/ubl-xsd2mdr.py



e-Document engineering pilot

Using a semantic metadata registry for e-Document engineering

Address

An address representation as defined in the data specifications of the EU INSPIRE Directive. The locrcaddressid property may be used to link this Address to other representations.

URI: http://mdr.semic.eu/id/core-location/class/Address

Type: Object Class

Raw data: HTML | RDF/XML | Turtle

Properties

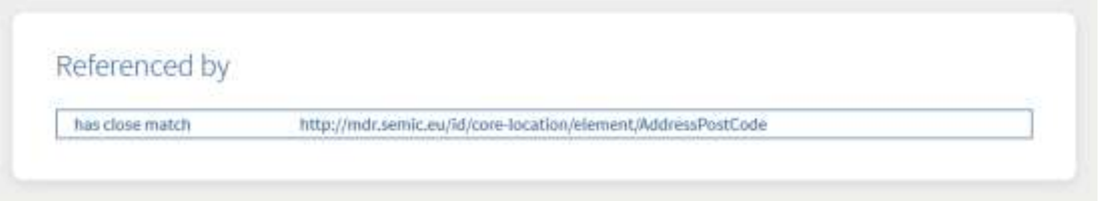
type	Object Class
label	Address
has context	Core Location
representationTechnique	http://purl.org/adms/representationtechnique/RDFSchema
definition	An address representation as defined in the data specifications of the EU INSPIRE Directive. The locrcaddressid property may be used to link this Address to other representations.
Rights	https://joinup.ec.europa.eu/category/licence/isa-open-metadata-licence-v11
Rights Holder	http://ec.europa.eu/
has close match	Address
has URI	Address

Figure 32 – Descriptions from standard libraries, such as the Core Location Address object class depicted here, can be easily browsed through the metadata registry web interface. (**http://mdr.semic.eu/id/core-location/class/Address**)

4.3.5.Use Case 2 – Create mappings between data elements

To highlight similarities between data elements of different libraries, the library maintainers or the repository maintainer adds mapping links between data elements, object classes and properties. Such links can also map data elements using different representation techniques.

For example, we have created a mapping between the "Address" class of the Core Location vocabulary, represented with RDF Schema, and the "Address" class of the UBL Common Library, represented with XML Schema. The SKOS "close match" relation is used on the object class, the properties, and the data elements, as shown in Figure 33.



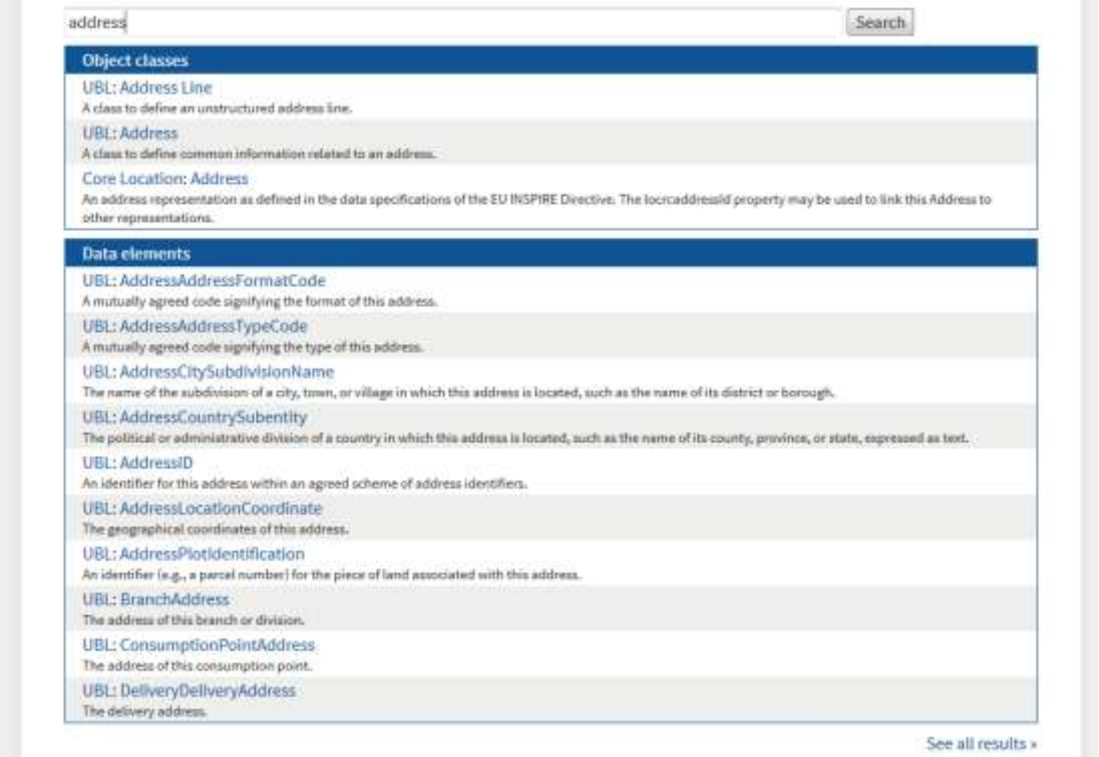
Referenced by

has close match	http://mdr.semic.eu/id/core-location/element/AddressPostCode

Figure 33 – The UBL "Address Postal Zone" data element is matched to the "Address Post Code" data element from the Location Core Vocabulary. (**http://mdr.semic.eu/id/ubl/element/AddressPostalZone**)

4.3.6.Use Case 3 – Search data elements

To facilitate reuse of existing data elements, a user can look up a keyword in the labels and descriptions of object classes and data elements. Using this functionality, one can easily discover and reuse data elements from common libraries. For example, Figure 34 shows the search for an address element.



Object classes

UBL: Address Line
A class to define an unstructured address line.

UBL: Address
A class to define common information related to an address.

Core Location: Address
An address representation as defined in the data specifications of the EU INSPIRE Directive. The locrcaddressId property may be used to link this Address to other representations.

Data elements

UBL: AddressAddressFormatCode
A mutually agreed code signifying the format of this address.

UBL: AddressAddressTypeCode
A mutually agreed code signifying the type of this address.

UBL: AddressCitySubdivisionName
The name of the subdivision of a city, town, or village in which this address is located, such as the name of its district or borough.

UBL: AddressCountrySubentity
The political or administrative division of a country in which this address is located, such as the name of its county, province, or state, expressed as text.

UBL: AddressID
An identifier for this address within an agreed scheme of address identifiers.

UBL: AddressLocationCoordinate
The geographical coordinates of this address.

UBL: AddressPlotIdentification
An identifier (e.g., a parcel number) for the piece of land associated with this address.

UBL: BranchAddress
The address of this branch or division.

UBL: ConsumptionPointAddress
The address of this consumption point.

UBL: DeliveryDeliveryAddress
The delivery address.

See all results »

Figure 34 – The search feature allows quick discovery of existing object classes and data elements. (**http://mdr.semic.eu/#registry**)

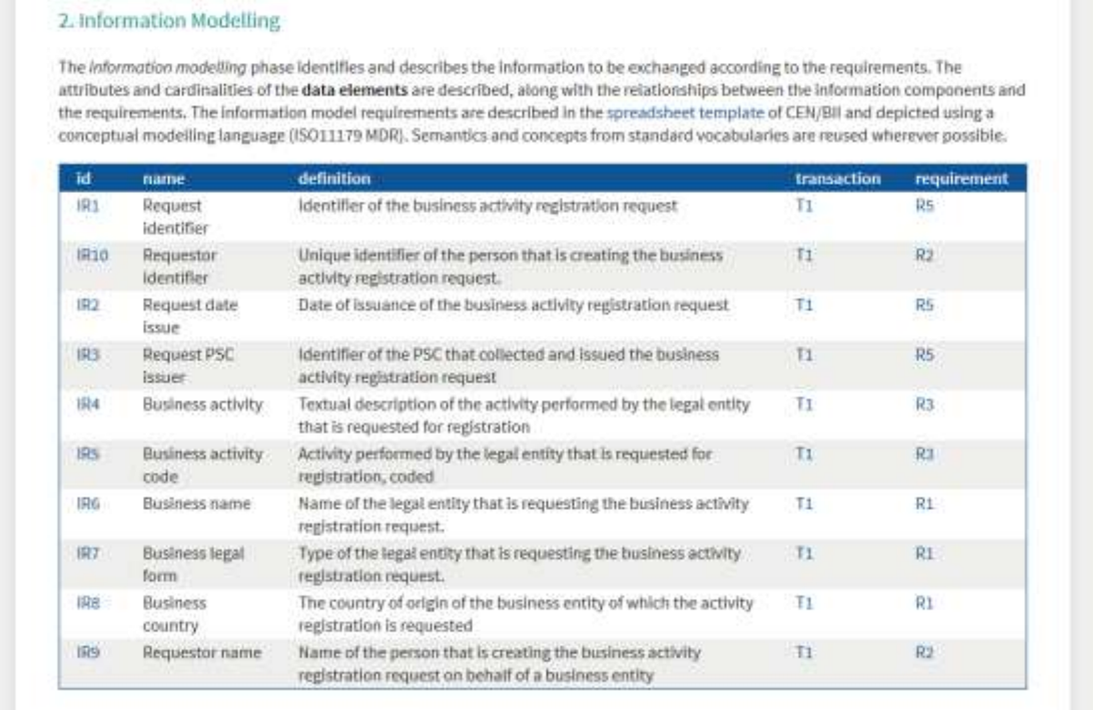
Because the metadata registry contains both data elements of common libraries and data elements of e-Document formats, one can explore the usage of data elements. Popular elements can thus be promoted.

New elements created in the specific context of an e-Document format can also be discovered and reused. If such elements become popular, they could then be included in libraries.

4.3.7. Use Case 4 – Register e-Document requirements

To provide the requirements for an e-Document format in a structured way, the format designer publishes an RDF description of the library using the model shown in Section 4.3.1. A Python script[46] can be used to transform a spreadsheet following the methodology described in Section 4.2 into RDF. Once the RDF data is imported in the metadata registry, one can easily browse through the requirements of an e-Document format. For example, Figure 35 shows the information requirements of the running example.

[46] https://github.com/SEMICeu/mdr/blob/master/scripts/edoc-xls2mdr.py



2. Information Modelling

The *information modelling* phase identifies and describes the information to be exchanged according to the requirements. The attributes and cardinalities of the **data elements** are described, along with the relationships between the information components and the requirements. The information model requirements are described in the spreadsheet template of CEN/BII and depicted using a conceptual modelling language (ISO11179 MDR). Semantics and concepts from standard vocabularies are reused wherever possible.

id	name	definition	transaction	requirement
IR1	Request identifier	Identifier of the business activity registration request	T1	R5
IR10	Requestor identifier	Unique identifier of the person that is creating the business activity registration request.	T1	R2
IR2	Request date issue	Date of issuance of the business activity registration request	T1	R5
IR3	Request PSC issuer	Identifier of the PSC that collected and issued the business activity registration request	T1	R5
IR4	Business activity	Textual description of the activity performed by the legal entity that is requested for registration	T1	R3
IR5	Business activity code	Activity performed by the legal entity that is requested for registration, coded	T1	R3
IR6	Business name	Name of the legal entity that is requesting the business activity registration request.	T1	R1
IR7	Business legal form	Type of the legal entity that is requesting the business activity registration request.	T1	R1
IR8	Business country	The country of origin of the business entity of which the activity registration is requested	T1	R1
IR9	Requestor name	Name of the person that is creating the business activity registration request on behalf of a business entity	T1	R2

Figure 35 – All steps of the e-Document engineering methodology are documented in the metadata registry. (**http://mdr.semic.eu/#information-modelling**)

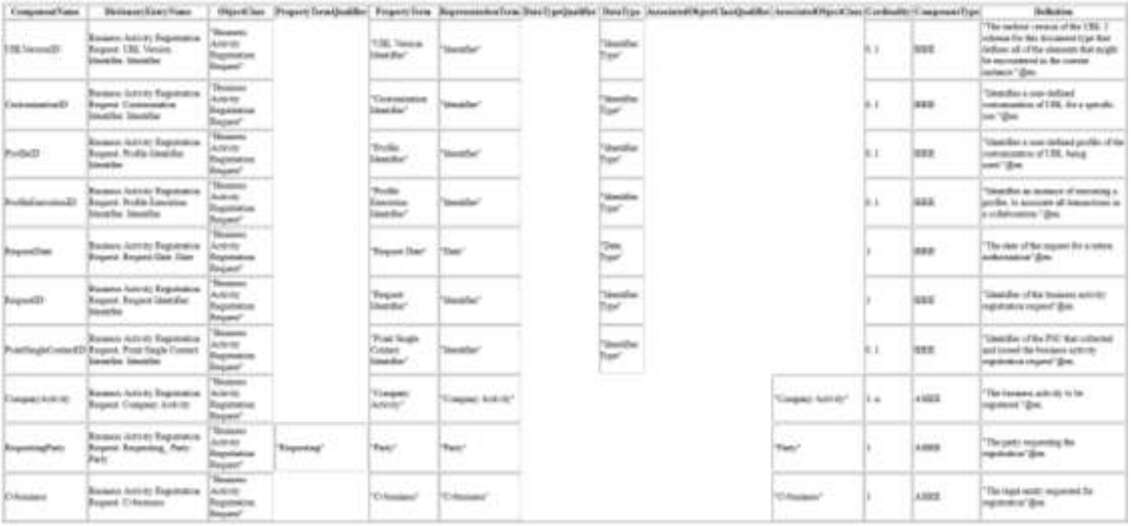
Each goal, high-level requirement, information requirement, and business rule is identified by a URI. Hence, requirements can be easily referred to in any context and links can be made.

4.3.8.Use Case 5 – Facilitate syntax binding / schema creation

To create a syntax binding for a transaction, the format designer creates links between information requirements and data elements, as shown in Figure 31. She can reuse entire e-Document formats by binding the information requirements to existing data elements, or produce a new e-Document format by binding the information requirements to new data elements created in the context of the e-Document format.

Object classes, properties, data types, and data elements contain all the information needed to generate a spreadsheet following UBL conventions. For example, data elements can have cardinality information and can be ordered. Property terms have optional qualifiers. Hence, it is possible to generate a spreadsheet and follow the approach described in Section 4.2.4. The following SPARQL query, whose results are shown in Figure 36, generates such information:

```
SELECT ?ComponentName ?DictionaryEntryName ?ObjectClass ?PropertyTermQualifier ?PropertyTerm
       ?RepresentationTerm ?DataTypeQualifier ?DataType ?AssociatedObjectClassQualifier
       ?AssociatedObjectClass ?Cardinality ?ComponentType ?Definition WHERE {

  <http://mdr.semic.eu/id/esens-activity-registration/transaction/T1> mdr:concretizedBy ?class .
  ?class a mdr:ObjectClass ;
    mdr:objectClassName ?ObjectClass .

  ?element a mdr:DataElement ;
    mdr:objectClass ?class ;
    mdr:property ?property ;
    mdr:order ?order ;
    skos:definition ?Definition .
  OPTIONAL { ?element mdr:minCardinality ?minCard }
  OPTIONAL { ?element mdr:maxCardinality ?maxCard }

  ?property a mdr:Property ;
    mdr:propertyTerm ?PropertyTerm ;
    mdr:representation ?repr .
  OPTIONAL { ?property mdr:propertyTermQualifier ?PropertyTermQualifier }

  OPTIONAL { ?repr a mdr:DataType ; mdr:representationTerm ?reprTerm . }
  OPTIONAL { ?repr a mdr:ObjectClass ; mdr:objectClassName ?AssociatedObjectClass . }
  BIND(IF(BOUND(?reprTerm), CONCAT(?reprTerm, ". Type"), ?reprTerm) AS ?DataType)
  BIND(COALESCE(?AssociatedObjectClass, ?reprTerm) AS ?RepresentationTerm)
  BIND(IF(BOUND(?AssociatedObjectClass), "ASBIE", "BBIE") AS ?ComponentType)

  OPTIONAL { ?property mdr:representationQualifier ?reprQual }
  BIND(IF(BOUND(?AssociatedObjectClass), ?reprQual, ?AssociatedObjectClass)
       AS ?AssociatedObjectClassQualifier)
  BIND(IF(!BOUND(?AssociatedObjectClass), ?reprQual, ?DataType)
       AS ?DataTypeQualifier)

  BIND(REPLACE(STR(?property), "^.*[/:#]([^/:#]*)$", "$1") AS ?ComponentName)
  BIND(CONCAT(?ObjectClass, ". ",
              IF(BOUND(?PropertyTermQualifier),
                 CONCAT(?PropertyTermQualifier, "_ "), ""),
              ?PropertyTerm,
              IF(BOUND(?PropertyTermQualifier) ||
                 ?PropertyTerm != ?RepresentationTerm,
                 CONCAT(". ", ?RepresentationTerm), ""))
       AS ?DictionaryEntryName)
  BIND(IF(BOUND(?minCard) && BOUND(?maxCard) && ?minCard = ?maxCard,
          STR(?minCard),
          CONCAT(COALESCE(STR(?minCard), "0"), "..",
                 COALESCE(STR(?maxCard), "n")))
       AS ?Cardinality)
} ORDER BY ?order
```



ComponentName	DictionaryEntryName	ObjectClass	PropertyTermQualifier	PropertyTerm	RepresentationTerm	DataTypeQualifier	DataType	AssociatedObjectClassQualifier	AssociatedObjectClass	Cardinality	ComponentType	Definition
UBLVersionID	Business Activity Registration Request. UBL Version Identifier. Identifier	"Business Activity Registration Request"		"UBL Version Identifier"	"Identifier"		"Identifier. Type"			0..1	BBIE	"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."@en
CustomizationID	Business Activity Registration Request. Customization Identifier. Identifier	"Business Activity Registration Request"		"Customization Identifier"	"Identifier"		"Identifier. Type"			0..1	BBIE	"Identifies a user-defined customization of UBL for a specific use."@en
ProfileID	Business Activity Registration Request. Profile Identifier. Identifier	"Business Activity Registration Request"		"Profile Identifier"	"Identifier"		"Identifier. Type"			0..1	BBIE	"Identifies a user-defined profile of the customization of UBL being used."@en
ProfileExecutionID	Business Activity Registration Request. Profile Execution Identifier. Identifier	"Business Activity Registration Request"		"Profile Execution Identifier"	"Identifier"		"Identifier. Type"			0..1	BBIE	"Identifies an instance of executing a profile, to associate all transactions in a collaboration."@en
RequestDate	Business Activity Registration Request. Request Date. Date	"Business Activity Registration Request"		"Request Date"	"Date"		"Date. Type"			1	BBIE	"The date of the request for a return authorization"@en
RequestID	Business Activity Registration Request. Request Identifier. Identifier	"Business Activity Registration Request"		"Request Identifier"	"Identifier"		"Identifier. Type"			1	BBIE	"Identifier of the business activity registration request"@en
PointSingleContactID	Business Activity Registration Request. Point Single Contact Identifier. Identifier	"Business Activity Registration Request"		"Point Single Contact Identifier"	"Identifier"		"Identifier. Type"			0..1	BBIE	"Identifier of the PSC that collected and issued the business activity registration request"@en
CompanyActivity	Business Activity Registration Request. Company Activity	"Business Activity Registration Request"		"Company Activity"	"Company Activity"				"Company Activity"	1..n	ASBIE	"The business activity to be registered."@en
RequestingParty	Business Activity Registration Request. Requesting_ Party. Party	"Business Activity Registration Request"	"Requesting"	"Party"	"Party"				"Party"	1	ASBIE	"The party requesting the registration"@en
[illegible]	Business Activity Registration Request. [illegible]	"Business Activity Registration Request"		"[illegible]"	"[illegible]"				"[illegible]"	1	ASBIE	"The legal entity requested for registration"@en

Figure 36 – Results of the SPARQL Query to generate a spreadsheet of data elements following UBL conventions.

Another option would be to directly generate XML Schemas from the results of the SPARQL query, by applying an XSLT file encoding the naming and design rules of UBL.

To generate an RDF Schema, a SPARQL CONSTRUCT query would be sufficient.

4.3.9.Use Case 6 – Enrich XML Schema documentation of e-Document formats

To enrich the documentation of the generated XML Schemas, linking XML elements to the underlying requirements, the format designer adds links in the XSD documentation to the corresponding data elements in the metadata registry. By visiting the included URI, one can retrieve the definition of the elements, and browse through the underlying information requirements, business rules, high-level requirements, and goals.

For example, the "RequestDate" element could contain the following documentation in the XML Schema (the first documentation block is generated for UBL):

```
<xsd:element ref="myb: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:documentation>
      <rdfs:seeAlso rdf:resource="http://mdr.semic.eu/id/esens-activity-
registration/element/BusinessActivityRegistrationRequestRequestDate" />
```

```
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

By visiting the included URI, one accesses the underlying data element, information requirement, high-level requirement, and goals, as shown in Figure 37.

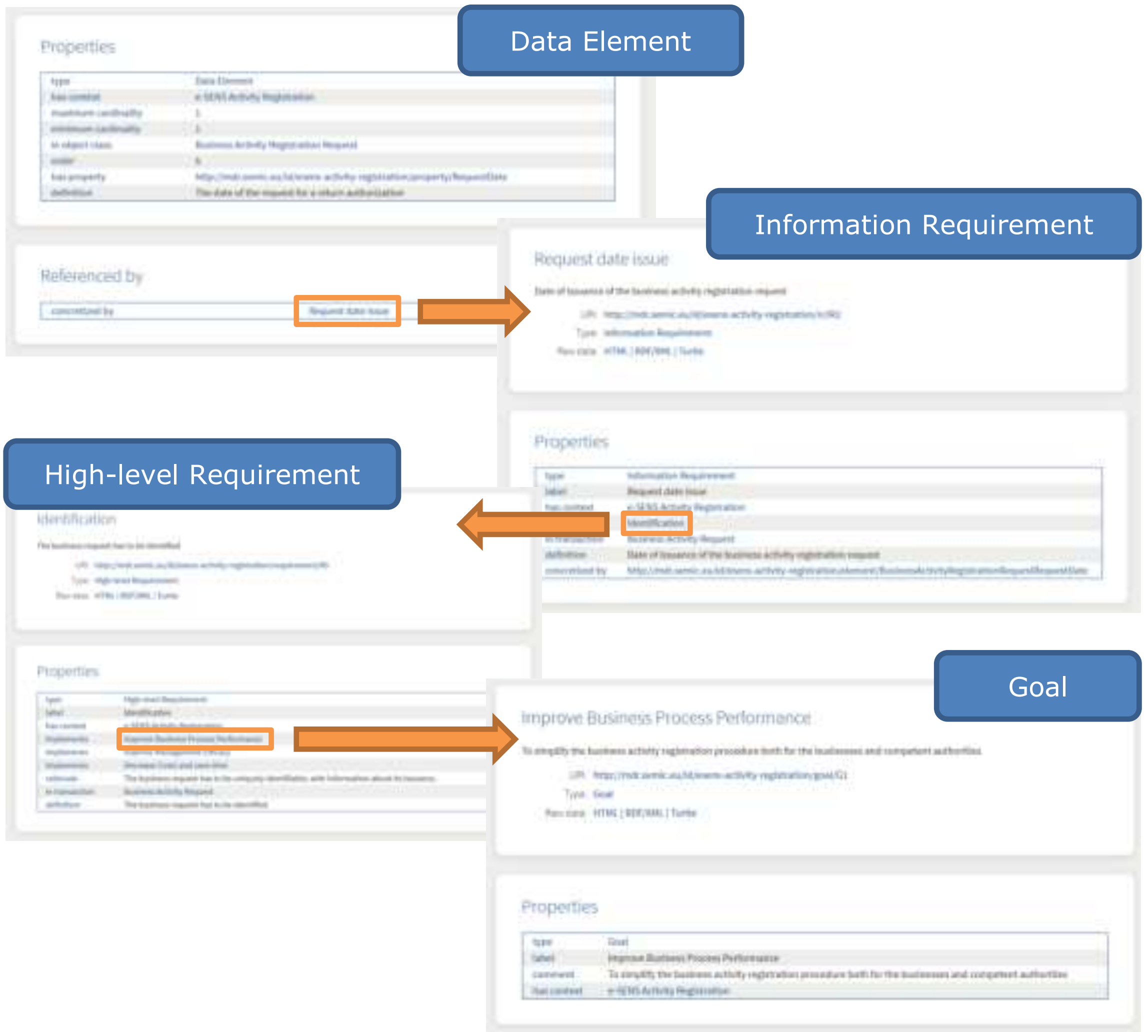


Figure 37 – By visiting the link in the documentation of the XSD element, one can retrieve all the requirements that led to that element, all the way up to the goals.
(http://mdr.semic.eu/id/esens-activity-registration/element/BusinessActivityRegistrationRequestRequestDate)

5. Recommendations on e-Document engineering methods

A recent survey[47] [ISA Programme, 2014] conducted on e-Document formats used in Trans-European Systems revealed that in half of the analysed cases public administrations develop their own formats, methods, and tools. From the assessments of engineering methods (Chapter 3) and the experience gained through the mini-pilot (Chapter 4), we derive a number of recommendations that public administrations can follow when working with e-Documents, which are explained in the remainder of this chapter::

1. Select a standard e-Document engineering method;
2. Use standard libraries;
3. Make e-Document formats available for reuse;
4. Follow good practices for metadata governance and management;
5. Explore the feasibility of operating a federated metadata registry; and
6. Use existing tools.

5.1. Select a *standard* e-Document engineering method

We recommend choosing a *standard* e-Document engineering method. Tools such as the Common Assessment Method of Standards and Specifications (CAMSS)[48] described in Chapter 3 may help in assessing and comparing engineering methods. Standard engineering methods include among others the:

- UN/CEFACT e-Document engineering method (see also Section 3.1);
- OASIS UBL TC method (see also Section 3.2);
- CEN BII e-Document engineering method (see also Section 3.3);
- ISO 20022 method (financial industry);
- HL7 method (healthcare sector);
- The method of the United States National Information Exchange Model (NIEM); and the
- XBRL method (financial reporting).

The preliminary step before assessing an e-Document engineering method using CAMSS is to define the requirements for the e-Document engineering method. Chapter 2 contains a number of *generic* requirements. Administrations should decide which requirements are important and which can be ignored in their specific contexts. From our experience, good e-Document formats do not only define a syntax binding or create a schema for the e-Document format, but also pay attention to information model requirements, definitions, and business rules.

[47] ISA Programme (2014) Analysis of structured e-Document formats used in Trans-European Systems.

[48] https://joinup.ec.europa.eu/community/camss/home

Rationale: The use of a standard e-Document engineering method, library of data elements, and XML naming and design rules makes it easier for public administrations to produce e-Document formats in such a formal way, producing standard outcomes that could be re-used in other projects.

Formalizing an e-Document engineering methodology has several **benefits:**

- **Speeds up the process to create e-Document formats**: A methodology depicts a systematic approach to build the XML Schemas and validation artefacts, reducing the uncertainty of the production process and speeding up the overall project.
- **Enhances documentation**: Every step is documented in a formal way. Project teams have to provide clear definition of goals and rationales for elements and components. Documentation from goals to the final implementation eases the understanding both for business and technical people.
- **Lowers the risk and decreases the cost**: The use of a standard e-Document engineering methods allows for the creation of tools to support the XML Schema production. Tools reduce the production time and the risk of adding errors in the production phase. Hence, the overall production cost is drastically lowered.
- **Facilitates maintenance and governance**: A clear method specifies the maintenance process of the artefacts and can provide guidance on long-term sustainability and governance.

5.2. Use *standard* libraries such as the Core Vocabularies

Wherever possible, public administrations should use data elements from standard libraries. Examples of standard libraries include:

- ISA Core Vocabularies[49]
- OASIS UBL Common Library[50]
- UN/CEFACT Core Components Library[51]
- HL7 Vocabulary[52]
- XBRL Recognized Taxonomies[53]

If new elements are introduced, links with similar elements from standard libraries should be identified and explained when practicable.

Rationale: Common concepts and structure, such as provided by standard libraries, enables quicker understanding of the e-Document format, reducing development and deployment time. Using standard libraries also facilitate data mash-up from different e-Documents and data models.

[49] https://joinup.ec.europa.eu/community/core_vocabularies/home
[50] https://www.oasis-open.org/standards#ublv2.1
[51] http://www.unece.org/cefact/codesfortrade/unccl/ccl_index.html
[52] http://www.hl7.org/permalink/?VocabTables
[53] http://www.xbrl.org/recognized

A standard library provides the following benefits:

- **Improves the implementation of e-Document formats**
    - **Increases reusability of elements and e-Documents**: The produced artefacts can be compared with existing standards and solutions, such as the ISA Core Vocabularies, increasing reusability and building common understanding on semantics.
    - **Eases the deployment of electronic documents in IT systems**: The deployment of electronic documents in IT systems requires IT experts and programmers to get the knowledge and understanding on the e-Document formats and the meaning of the data elements referred therein.

      Reusing common libraries such as the ISA Core Vocabularies or e-Document engineering method – and thus reusing common technical structures – allows IT experts to lower their learning curve, being more efficient and productive when implementing e-Documents in IT systems.
- **Facilitates data mash-up from different e-Documents and data models:** Using shared concepts and structure in different e-Document formats fosters the aggregation of information from disparate, possibly cross-sector, e-Documents. By reusing standard libraries, the probability that two e-Document formats share data elements increases, as does the amount of data that can be automatically aggregated increases. For example, if two e-Document formats share 80% of data elements and reference data, only the remaining 20% has to be mapped together. If links with standard libraries are described, this process can be greatly simplified.

  Some interoperability conflicts cannot be easily resolved through mappings. For example, aggregation conflicts [Peristeras, Loutas, Goudos, & Tarabanis, 2008] occur when data is categorized in different ways. By sharing common reference data provided by standard libraries, such conflict can be avoided.

5.3. Make e-Document formats available for reuse

Public administrations should make the final e-Document formats available for reuse by administrations from other sectors and/or other countries. The process involves the following steps:

1. Choose an open licence such as the ISA Open Metadata Licence[54] or a Creative Commons[55] licence.
2. Make the e-Document formats publicly available for download on the web under the licence chosen above.
3. Rigorously document the e-Document formats and make the documentation publicly available on the web.

[54] https://joinup.ec.europa.eu/category/licence/isa-open-metadata-licence-v11
[55] http://creativecommons.org/

4. Describe the e-Document formats with ADMS and publish it on the Joinup[56] platform.

Rationale: By publishing the e-Document formats with an open licence, other public administration can reuse them, increasing interoperability. Good documentation is essential to enable others to understand and reuse the e-Document formats.

In addition to facilitating interoperability, the reuse of an e-Document format by other public administrations can also provide valuable feedback, helping the initial public administration in improving the e-Document format.

5.4. Follow good practices for metadata governance and management

Public administrations should provide clear governance mechanism and management processes for their e-Document formats. Such mechanisms include for example how updates to the e-Document format are published, or how external contributions are handled.

Note that governance mechanisms are not covered in this deliverable. The reader is referred to deliverable *D4.2 Methodology and tools for Metadata Governance and Management for EU Institutions and Member States* for more information.

An important principle covered in D4.2 is the separation of data models from reference data. For example, UBL 2.0 expresses code list values in separate files using the genericode format, instead of binding those values inside the document schemas.[57]

Rationale: Data models and reference data have different life cycles. These differences are linked to the different needs for stability versus flexibility. Separating data models and reference data allows for independent updates of one or the other.

Data models are strongly linked to the interoperability of applications and therefore changes in a data model have a direct effect on the applications that are based on it. In many cases, software systems will need to be rebuilt importing the new model and upgrading the functionality before they can interoperate with others. In practice, changes in data models will be relatively infrequent (less than annual) and changes will be accompanied by a strongly managed implementation plan aligned with a software upgrade cycle.

Reference data is usually more loosely linked to the basic functionality of applications. Changing or adding a code in a code list will not have a disruptive effect on the existing functionality. These types of changes may also occur with a higher frequency (one or more times per year) than model changes, and are usually easier to propagate through a network.

[56] http://joinup.ec.europa.eu/

[57] https://www.oasis-open.org/committees/ubl/faq.php

5.5. Explore the feasibility of operating a *federated* metadata registry

Public administrations should explore the feasibility of building a metadata registry containing *uniform* descriptions of data elements used in structural metadata. Each registered item would contain at least an identifier, a name, and a definition (possibly in multiple languages). The metadata registry could be either maintained by a central authority or federated from multiple smaller repositories. Section 4.3 describes a proof-of-concept of the metadata registry.

A common metadata registry could be set up in a phased approach. In a first time, machine-readable descriptions of e-Document formats and data elements should be generated. In a second time, those descriptions should be written with a uniform vocabulary. The following technologies can contribute to this:

- *Semantic technologies*, like the Resource Description Framework (RDF), offer a flexible approach to publishing metadata. Their open-world model makes it easy to build a federated registry.
- The *ISO 11179 standard* proposes a meta-model for describing data elements. It is a first step towards a common vocabulary. Descriptions of standard libraries such as the UN/CEFACT Core Components Library and the UBL Common Library are based on the ISO 11179-3 meta-model.

Rationale: Uniform descriptions of data elements increase their discoverability and reuse. By bringing together data elements from standard libraries and from e-Document formats, the metadata registry gives public administrations more insight into the elements that are often reused or that can be reused. The metadata registry can also be used to link data elements to the underlying requirements, improving an e-Document format's documentation through traceability. Section 4.3 elaborates on the use cases of the metadata registry.

Note that a metadata registry is complementary to an interoperability solution repository like Joinup, which only contains high-level descriptions of solutions such as e-Document formats among others. A metadata registry is more granular, at the level of the data elements. Hence, it can be an aid during the e-Document format design. On the other hand, a metadata registry has a narrower scope than a general platform like Joinup.

5.6. Use existing tools

Public administrations should make use of existing tools when possible. Such tools exist for standard e-Document engineering methods. Table 36 lists some tools that can be used when creating e-Document formats.

Rationale: using existing tools decreases the development effort and reduces the risk for errors. Tools that encode naming and design rules from standard engineering methods also ensure that the output is compliant with the standard, increasing interoperability.

Table 36 – Non-exhaustive list of tools supporting the creation of e-Document formats.

Tool	Description
Metadata Workbench	**Schema creation** UN/CEFACT NDR The Large Scale Pilot e-CODEX (http://www.e-codex.eu) has used a Dutch tool, the Metadata Workbench (MWB). MWB is an integrated tool, in which the Core Components can be specified and that allows the derivation of BIEs and the specification of Business Documents. The tool can be acquired free of charge.
XGenerator	**Schema creation** XÖV NDR / any (configurable) XGenerator is the model-driven engineering solution that is part of the German XÖV (XML in der Öffentlichen Verwaltung) initiative, the XML-based data interchange methodology of the German federal Government. XGenerator is an open-source tool capable of validating UML data models created according to the XÖV UML Profile and generating XML Schemas for message interchange from there. https://joinup.ec.europa.eu/software/xgenerator/description
Crane Software GC-to- UBL NDR script	**Schema creation** OASIS UBL NDR Crane’s Genericode to UBL NDR script implements the UBL Naming and Design Rules 2.1 XSD schemas and OASIS CVA (context/value association) files from an input OASIS genericode expression of a compatible UBL NDR 2.1 spreadsheet. This package can be used in any project wanting to create schemas and CVA files following the UBL NDR 2.1, not just files for UBL. This package can also be used to create document extension schemas and additional document schemas for any project using the UBL NDR 2.1. Available at: http://www.cranesoftwrights.com/resources/ubl/index.htm#gc2ublndr
eDoCreator	**Schema creation** OASIS UBL NDR The iSurf eDoCreator is an on-line tool that provides the CCTS-based document schemas. The tool allows users to import their own components by uploading a description in a spreadsheet. The tool has a Web-based user interface and supports collaborative development. Available at: http://www.srdc.com.tr/home/index.php?option=com_content&view=article&id=90&Itemid=84&lang=en
GEFEG.FX	**Information Modelling + Schema creation** UN/CEFACT NDR, OASIS UBL NDR, … GEFEG.FX is a commercial tool for schema development and schema requirement management. Functions in GEFEG.FX include the design of electronic commercial documents as models, XML schemas or EDI standards; the creation of custom specifications in a guideline; syntax and semantics tests; and the visualization of real messages. http://www.gefeg.com/en/index.htm

	http://www.gefeg.com/en/standard/uml/ccts-module.htm
Enterprise Architect + ShapeChange	**Information Modelling + Schema creation** GML NDR Enterprise Architect is a high performance modeling, visualization and design platform based on the UML standard. The tool has built-in requirements management capabilities, tracing high-level specifications to analysis, design, implementation, test and maintenance models using UML, SysML, BPMN and other open standards. http://www.sparxsystems.com/products/ea/index.html

6. CONCLUSION

The goal of the study described in this report was to analyse e-Document engineering methods in order to derive recommendations for public administrations about the engineering of e-Document formats. We have conducted the following activities:

- We have **enumerated generic functional requirements for e-Document engineering methods** (Chapter 2). These requirements can be used by public administrations as a basis for identifying their own needs depending on their specific context.
- We have **conducted an assessment of three standard e-Document engineering methods**: UN/CEFACT, OASIS UBL, and CEN BII (Chapter 3). Such assessments highlight the similarities and differences between the e-Document engineering methods. They help public administrations in choosing the right method based on their needs.
- We have **carried out a mini-pilot on e-Document engineering** in collaboration with the e-SENS WP 6.2 Competence Cluster on Semantics, processes and Documents (Chapter 4). The mini-pilot demonstrates the use of a standard e-Document engineering method to create a new e-Document format. The mini-pilot also includes a **tutorial** (Annex I) on how to generate XML Schema syntax bindings using open-source software.
- We have **explored the feasibility of using a metadata registry** in combination with e-Document engineering (Section 4.3). The metadata registry contains uniform descriptions of data elements both from standard libraries and from e-Document formats, increasing their discoverability and reuse by public administrations. The metadata registry also improves the rigorous documentation of the requirements underlying an e-Document format.

A recent survey[58] [ISA Programme, 2014] that we conducted on e-Document formats used in Trans-European Systems revealed that in half of the analysed cases public administrations even develop their own methods, libraries of data elements, and tools:

- Projects were not generally using standard e-Document engineering methods. Ad-hoc methods are used to define requirements, leading to poorly-documented requirements with little or no traceability, i.e., it is difficult to link data elements back to the underlying requirements.
- Projects were not generally using standard libraries, but create their own data elements. Hence, there is little interoperability between e-Document formats created in different projects.
- Projects tended to develop their own tools to create the e-Document schemas, incurring an increase in development cost. Different tools also result in different outputs, again decreasing interoperability.

[58] ISA Programme (2014) Analysis of structured e-Document formats used in Trans-European Systems.

Considering such observations, we have derived a number of **recommendations** for public administrations (Chapter 5):

1. **Select a *standard* e-Document engineering method**, speeding up the process, enhancing documentation, lowering risk and cost, and facilitating maintenance and governance;
2. **Use *standard* libraries**, increasing interoperability and ease of development and deployment, and facilitating mash-up from different e-Documents and data models;
3. **Make e-Document formats available for reuse**, increasing cross-sector and cross-border interoperability, and gaining additional feedback from peers;
4. **Follow good practices for metadata governance and management**, ensuring the stability and durability of the e-Document formats;
5. **Explore the feasibility of operating a federated metadata registry**, increasing discoverability and reuse of data elements from standard libraries and other e-Document formats, describing mappings between libraries, and enhancing documentation and traceability of e-Document formats through links between data elements and requirements; and
6. **Use existing tools**, reducing cost and risk for errors, and increasing interoperability through uniformity and compliance with standards.

To narrow the gap between our observations and our recommendations, we suggest the following **possible next steps**:

- **Promote the use of Core Vocabularies in e-Document engineering**: An important step towards the adoption of the Core Vocabularies was taken in this report, as we have demonstrated how they can be used as a starting point to engineer e-Document formats. This is demonstrated in a tutorial using open-source software. We were also in contact with other tool developers (e.g. Metadata Workbench of the Dutch Ministry of Justice, XGenerator of the German IT-Standards Coordination Office - KoSIT) to see whether they could integrate the Core Vocabularies as standard libraries within their tool. We recommend these efforts to be continued and reinforced.

- **Elaborate guidelines on process modelling**: The modelling of the complete business process in which e-Documents are used provides valuable insight to understand and formalize the goals and requirements of the e-Document formats. This deliverable mentions such modelling without diving into details. From our work on the mini-pilot in collaboration with the e-SENS Large Scale Pilot, it appears that there is a demand for guidelines on how to model business processes effectively.
- **Consider developing a Core Vocabulary for metadata registries**: Section 4.3 has demonstrated the use cases of a common metadata registry based on semantic technologies containing uniform descriptions of data elements. However, the vocabulary that was created as part of the proof-of-

concept needs additional thought to be used in production. A Core Vocabulary is especially needed to federate descriptions of data elements from various sources. A starting point for the Core Vocabulary could be the ISO11179 standard and the use cases for the metadata registry described in Section 4.3 related to the registration of data elements, the creation of mappings between them and the search for data elements. In addition to this, the visualisation of commonalities and differences in libraries and data models is also to be considered.

- **Perform assessment of additional e-Document engineering methods**: In addition to the three standard e-Document engineering methods analysed in the this report, other standard methods are to be assessed such as the methods linked to the eXtensible Business Reporting Language (XBRL), Health Level 7 (HL7), and ISO 20022.

ACKNOWLEDGEMENTS

Specific acknowledgement is due to:

Person	Organisation
Tomasz Dębicki	Institute of Logistics and Warehousing, Poland
Martin Forsberg	Single Face To Industry, Sweden
Muriel Foulonneau	Tudor Institute, Luxembourg
Antonis Stasis	Head of Directorate- Hellenic Ministry of Administrative Reform and e-Governance
Tim McGrath	Document Engineering Services

REFERENCES

CEN. (2012). *CEN/ISSS WS/BII-Z Implementation and Use of Validation Artifacts - Technical Guideline.* CEN.

CEN. (2012). *CWA 16558 - Annex O - Report Versioning and Change Management Policy*. CEN.

CEN BII 2. (2013). *BII2 workshop review*. Retrieved February 21, 2014, from CEN BII 2: http://www.cenbii.eu/deliverables/bii2-workshop-review/

CEN BII 2. (2013). *Participants*. Retrieved February 21, 2014, from CEN BII 2: http://www.cenbii.eu/about/participants/

CEN BII 2. (2013). *SFTI Sveorder*. Retrieved February 21, 2014, from CEN BII 2: www.cenbii.eu/cen-bii-in-practice/customer-testemonials/sveorder/

CEN BII 2. (2014). *Goals*. Retrieved March 20, 2014, from CEN BII 2: http://www.cenbii.eu/about/goals/

Collibra. (2010). *Business Semantics Management at the Flemish Public Administration.* Collibra.

Digitaliseringsstyrelsen. (2012, May 30). *About OIOXML*. Retrieved November 22, 2013, from Digitaliseringsstyrelsen: http://www.digst.dk/Servicemenu/English/IT-Architecture-and-Standards/Standardisation/Standardisation-creating-digital-Denmark/About-OIOXML

e-SENS. (2013, August 27). Electronic Simple European Networked Services - D6.1 Executable ICT Baseline Architecture.

European Commission. (2010). *Evaluation of the 2004 action plan for electronic public procurement.* Brussels: European Commission.

European Commission. (2011, December 12). Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the regions. *Open data - An engine for innovation, growth and transparent governance*. Brussels, Belgium.

Fabian Büttner, U. B. (2013). *Model-driven Standardization of Public Authority Data Interchange.*

Glushko, R. J., & McGrath, T. (2005). *Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services* . The MIT Press.

Government On-Line Metadata Working Group. (2006). *Records Management Application Profile.* Canada: Government of Canada.

Graux, H. (2009). *Study on electronic documents and electronic delivery for the purpose of the impementation of Art. 8 of the Services Directive.* Brussels: Timelex.

Hohpe, G., & Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* Addison-Wesley.

IDABC Programme. (2012, June 4). *CAMSS Assessment Criteria*. Retrieved November 27, 2013, from IDABC - CAMSS: https://webgate.ec.europa.eu/fpfis/mwikis/idabc-camss/index.php/CAMSS_Assessment_Criteria

Interoperability solutions for European public administrations (ISA). (2011, May 5). eGovernment Core Vocabularies: The SEMIC.EU approach. Brussels, Belgium: European Commission.

ISA. (2011). *European Interoperability Framework (EIF) - Towards Interoperability for European Public Services.* Luxembourg: Publications Office of the European Union.

ISA Programme. (2014). *Analysis of structured e-Document formats used in Trans-European Systems.*

Jaeger, T. D. (2013, June 03). *CAMSS - 05 - Detailed CAMSS Criteria*. Retrieved November 27, 2013, from Joinup: https://joinup.ec.europa.eu/community/camss/wiki/camss-05-detailed-camss-criteria

NIEM. (2013). *NIEM Tools Catalog*. Retrieved 11 25, 2013, from NIEM | National Information Exchange Model: https://www.niem.gov/tools-catalog/Pages/tools.aspx

OASIS. (2004, November 08). *Universal Business Language (UBL) Ratified As OASIS Standard*. Retrieved February 19, 2014, from OASIS: https://www.oasis-open.org/news/pr/universal-business-language-ubl-ratified-as-oasis-standard

OASIS. (2009, December 25). *UBL 2 Guidelines for Customization, First Edition*. Retrieved February 19, 2014, from OASIS: http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.html

OASIS. (2013, July 30). *Technical Committee (TC) Process*. Retrieved February 19, 2014, from OASIS: https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess

OASIS. (2013). *Universal Business Language Version 2.1.* OASIS Open.

OASIS. (2014). *OASIS Universal Business Language TC*. Retrieved February 19, 2014, from OASIS: https://www.oasis-open.org/committees/ubl/faq.php

OASIS. (2014). *Intellectual Property Rights (IPR) Policy*. Retrieved March 17, 2014, from OASIS: https://www.oasis-open.org/policies-guidelines/ipr#licensing_req

OASIS. (2014). *OASIS Universal Business Language (UBL) TC*. Retrieved February 18, 2014, from OASIS: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

Object Management Group, Inc. (2012, October). *Unified Modeling Language™ (UML®)*. Retrieved November 20, 2013, from Object Management Group: http://www.omg.org/spec/UML/

OHIM. (2013). *Community Trademark Registration Process*. Retrieved December 15, 2013, from https://oami.europa.eu/ohimportal/en/registration-process

Open Knowledge Definition. (n.d.). *Open Definition*. Retrieved 11 20, 2013, from Open Definition: http://opendefinition.org/

PEPPOL. (2013). *Virtual Company Dossier*. Retrieved November 22, 2013, from PEPPOL | Pan-European Public Procurement Online: http://www.peppol.eu/peppol_components/virtual-company-dossier

Peristeras, V., Loutas, N., Goudos, S. K., & Tarabanis, K. (2008). A conceptual analysis of semantic conflicts in pan-European e-government services. *Journal of Information Science, 34*(6), 877-891.

Roy, D. (n.d.). *National Information Exchange Model (NIEM): Technical Introduction to NIEM.*

Shukair, G., Loutas, N., Peristeras, V., & Sklarß, S. (2013). Towards semantically interoperable metadata repositories: The Asset Description Metadata Schema. *Computers in Industry*(64), 10 - 18.

Sinaci, A. A. (2012, November 07). Jena based Implementation of a ISO 11179 Meta-data Registry. Retrieved from http://www.slideshare.net/sinaci/jena-based-implementation-of-a-iso-11179-meta-data-registry-15066461

Spanish Ministry of Finance and Public Administration. (n.d.). Decision of 19 February 2013 of the secretary of state for public administration approving the technical interoperability standard for the reuse of information resources.

UN/CEFACT. (2006). *UN/CEFACT Intellectual Property Rights Policy - Note by the UN/CEFACT Bureau.* Geneva: United Nations.

UN/CEFACT. (2008). *UML Profile for Core Components (UPCC).*

UN/CEFACT. (2009). *Core Components Technical Specification - Version 3.0.*

UN/CEFACT. (2009). *XML Naming and Design Rules Technical Specification - Version 3.0.*

UN/CEFACT. (2009). *XML Naming and Design Rules Technical Specification Version 3.0.* UN/CEFACT.

UN/CEFACT. (2011). *Draft revised open development process.* United Nations.

UN/CEFACT. (2011). *Revised rules of procedure for the UN/CEFACT Bureau.* United Nations.

UN/CEFACT. (2011). *UML Profile for UN/CEFACT's Modeling Methodology (UMM) - Base Module - Version 2.0 Technical Specification.* UN/CEFACT.

UN/CEFACT. (2012). *Core Components Business Document Assembly - Technical Specification.* UN/CEFACT.

UN/CEFACT. (2012). *Core Components Business Document Assembly - Technical Specification - Version 1.0.*

UN/CEFACT. (2012, June 27). Core Components Business Document Assembly Technical Specification.

UN/CEFACT. (2012). *Updated executive summary of the UN/CEFACT Intellectual Property Rights Policy.* Geneva: United Nations.

UN/CEFACT. (2012). *Updated UN/CEFACT Intellectual Property Rights Policy.* Geneva: United Nations.

ANNEX I TUTORIAL ON E-DOCUMENT ENGINEERING

This annex contains the e-Document engineering tutorial that was created as a result of this work. The tutorial was published on Joinup: https://joinup.ec.europa.eu/node/78939

This tutorial explains how to create an electronic Document in XSD format using the Genericode to UBL NDR tool of Cranesoftwrights. We have used the tool to create a sample document called 'Business Activity Registration Request' using the ISA Core Vocabularies and the UBL Naming and Design Rules.

The Genericode to UBL NDR is an open-source package provided by Crane Softwrights available under the Modified BSD Licence. This package allows creating UBL 2.1 XSD Schemas and OASIS CVA (context/value association) files according to the UBL Naming and Design Rules. In 2012, the script was already used to produce the original XSD Schemas of the ISA Core Vocabularies. The input for the package is a UBL NDR 2.1 spreadsheet expressed using the OASIS Genericode standard. In order to create this OASIS Genericode file, we have used an OpenOffice UBL NDR 2.1 spreadsheet template and the open-source OpenOffice spreadsheet export to Genericode subset export filter that serializes the contents of the spreadsheet as a set of Genericode rows.

In this tutorial, we use the Genericode to UBL NDR package to create a new XSD Schema for the Business Activity Registration e-Document using the ISA Core Vocabularies and UBL as the main libraries of reusable elements.

In order to create the schema, we have used the files listed in the table below.

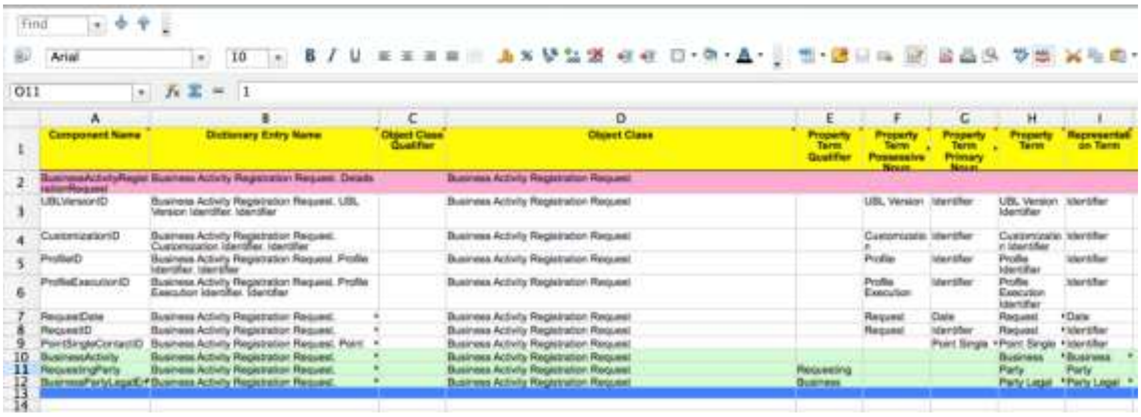
File	Description
Information Requirement Model e-Document Engineering Methods - Template Activity_Registration	Following the e-Document engineering method, we have analysed the goals and requirements for the pilot to create a Business Activity Registration Request. We have collected a set of information requirement models that specify the semantics we have to exchange using the new e-Document. We have captured the information requirement model following the e-Document engineering method in a spreadsheet form containing the goals, scope, requirements and information models, and we will use that model to populate the Pilot e-Document spreadsheet, from where the XSD Schemas will be generated.
ISA Core Vocabularies CoreVocabulary-v1.00.ods	OpenOffice UBL NDR 2.1 spreadsheet with the ISA Core Vocabularies contains reusable components for our e-Document. In our pilot, we will use the ISA Core Vocabularies as the common classes to be used in the final document for maximum reuse. The current pilot does not require additional classes to be created, therefore, there will be only a main document schema for the Business Activity Registration Request main e-Document and two XSD Schemas for the ISA Core Vocabulary aggregate and basic components.

UBL Common Library	The set of XSD Schemas from the UBL 2.1 Standard are used as the layout and as a set of reusable components to be used in the pilot. • CCTS_CCT_SchemaModule-2.1.xsd • UBL-CommonAggregateComponents-2.1.xsd • UBL-CommonBasicComponents-2.1.xsd • UBL-CommonExtensionComponents-2.1.xsd • UBL-CommonSignatureComponents-2.1.xsd • UBL-CoreComponentParameters-2.1.xsd • UBL-ExtensionContentDataType-2.1-original.xsd • UBL-ExtensionContentDataType-2.1.xsd • UBL-QualifiedDataTypes-2.1.xsd • UBL-SignatureAggregateComponents-2.1.xsd • UBL-SignatureBasicComponents-2.1.xsd • UBL-UnqualifiedDataTypes-2.1.xsd • UBL-XAdESv132-2.1.xsd • UBL-XAdESv141-2.1.xsd UBL-xmldsig-core-schema-2.1.xsd
Pilot e-Document BusinessActivityRegistrationRequest.ods	OpenOffice UBL NDR 2.1 spreadsheet with the e-Document model and its aggregated components. This spreadsheet has to contain two sheets, one with the e-Document model and another with the common components for the new e-Document. The main document sheet contains data elements derived from the Information Requirement Model and the common components sheet contain the ISA Core Vocabularies.

STEP 1- Create the spreadsheet with syntax bindings to the Core Vocabularies and UBL

The first step is to create the spreadsheet from which the e-Document format will be generated. The file has to be populated from the information requirement model created in previous stages of the methodology (see e-Document Engineering Methods - Template Activity_Registration).

In this step, we map information requirements to existing ISA Core Vocabulary elements or UBL aggregates when possible. If there are concepts neither in UBL library nor in the ISA Core Vocabularies, we can create them in the BusinessActivityRegistrationRequest common sheet. In our pilot, we will not need to create additional classes in the common sheet.



	A	B	C	D	E	F	G	H	I
1	Component Name	Dictionary Entry Name	Object Class Qualifier	Object Class	Property Term Qualifier	Property Term Possessive Noun	Property Term Primary Noun	Property Term	Representation Term
2	BusinessActivityRegistrationRequest	Business Activity Registration Request. Details		Business Activity Registration Request					
3	UBLVersionID	Business Activity Registration Request. UBL Version Identifier. Identifier		Business Activity Registration Request		UBL Version	Identifier	UBL Version Identifier	Identifier
4	CustomizationID	Business Activity Registration Request. Customization Identifier. Identifier		Business Activity Registration Request		Customizatio n	Identifier	Customizatio n Identifier	Identifier
5	ProfileID	Business Activity Registration Request. Profile Identifier. Identifier		Business Activity Registration Request		Profile	Identifier	Profile Identifier	Identifier
6	ProfileExecutionID	Business Activity Registration Request. Profile Execution Identifier. Identifier		Business Activity Registration Request		Profile Execution	Identifier	Profile Execution Identifier	Identifier
7	RequestDate	Business Activity Registration Request.		Business Activity Registration Request		Request	Date	Request	Date
8	RequestID	Business Activity Registration Request.		Business Activity Registration Request		Request	Identifier	Request	Identifier
9	PointSingleContactID	Business Activity Registration Request. Point		Business Activity Registration Request			Point Single	Point Single	Identifier
10	BusinessActivity	Business Activity Registration Request.		Business Activity Registration Request				Business	Business
11	RequestingParty	Business Activity Registration Request.		Business Activity Registration Request	Requesting			Party	Party
12	BusinessPartyLegalE	Business Activity Registration Request.		Business Activity Registration Request	Business			Party Legal	Party Legal
13									
14									

Figure 38 – Openoffice spreadsheet to create the UBL NDR document

The Pilot e-Document file has to follow the UBL NDR metadata. The elements captured in UBL are based on the ebXML Core Component Technical Specification Version 2.01. Each sheet 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 e-Document will be the same for all properties of the e-Document.
- **Property Term Qualifier** – Property Term Qualifiers specialize or modify the Property Term. For example, when the BIE is used in another context. In our case, when reusing a class like the UBL Party class, we add a qualifier `Requesting` to specify the type of Party.
- **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.

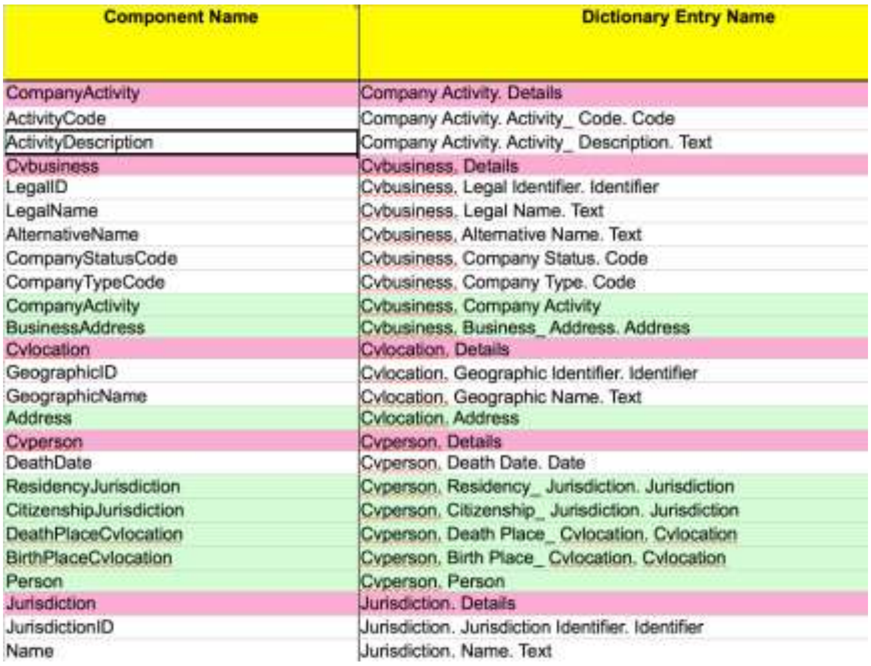
- **Associated Object Class** – This is the object class at the other end of the association. It is an ABIE in this model. We associate properties to classes from other vocabularies such as the ISA Core Vocabularies or the UBL Common Library. For instance, we reuse the class `CvBusiness` from the ISA Core Vocabularies to identify *"the legal entity requested for registration"*, and the class `Party` from the UBL Common Library to identify the "*Party requesting the registration"*.
- **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.

The e-Document sheet has to be filled in following the requirements from the Information Requirement Model:

- Defining the simple information requirements as properties of the new e-Document,
- Identifying reusable components from the ISA Core Vocabularies and UBL common aggregate library and creating new aggregates when needed.

In the process of populating the spreadsheet, some concepts from the information requirement model can be grouped and mapped to the ISA Core Vocabularies:

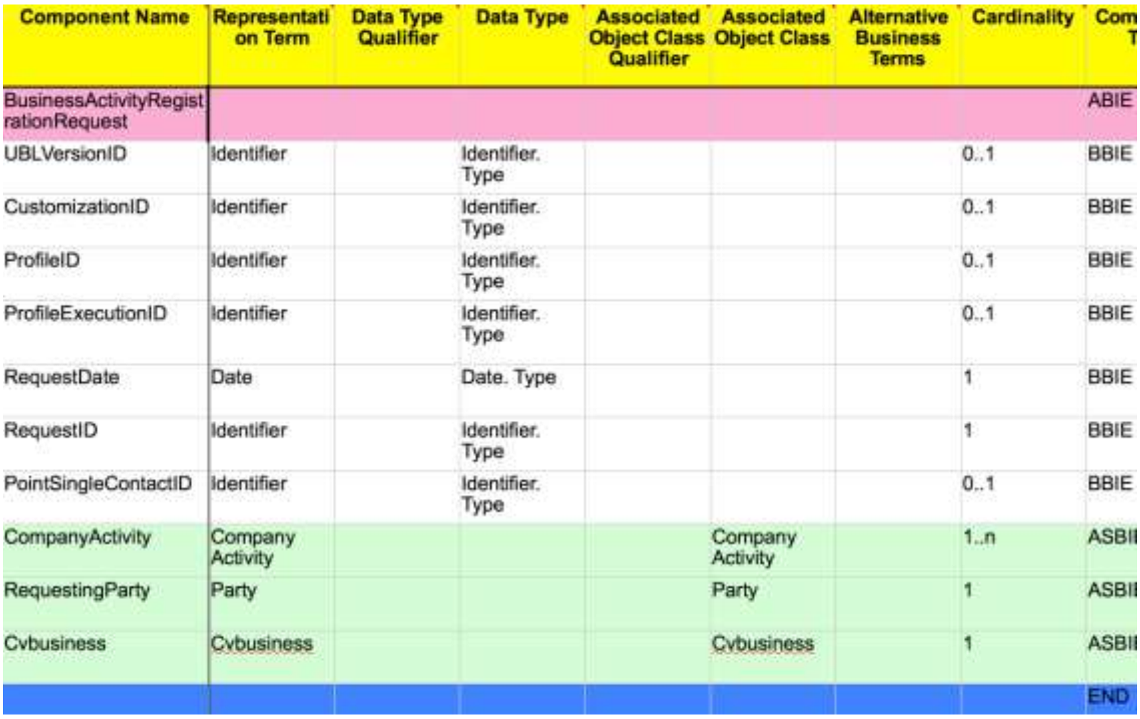
- For the Business Activity Registration Request information requirement model we have reused the Company Activity class from the ISA Core Vocabularies.



Component Name	Dictionary Entry Name
CompanyActivity	Company Activity. Details
ActivityCode	Company Activity. Activity_ Code. Code
ActivityDescription	Company Activity. Activity_ Description. Text
Cvbusiness	Cvbusiness. Details
LegalID	Cvbusiness. Legal Identifier. Identifier
LegalName	Cvbusiness. Legal Name. Text
AlternativeName	Cvbusiness. Alternative Name. Text
CompanyStatusCode	Cvbusiness. Company Status. Code
CompanyTypeCode	Cvbusiness. Company Type. Code
CompanyActivity	Cvbusiness. Company Activity
BusinessAddress	Cvbusiness. Business_ Address. Address
Cvlocation	Cvlocation. Details
GeographicID	Cvlocation. Geographic Identifier. Identifier
GeographicName	Cvlocation. Geographic Name. Text
Address	Cvlocation. Address
Cvperson	Cvperson. Details
DeathDate	Cvperson. Death Date. Date
ResidencyJurisdiction	Cvperson. Residency_ Jurisdiction. Jurisdiction
CitizenshipJurisdiction	Cvperson. Citizenship_ Jurisdiction. Jurisdiction
DeathPlaceCvlocation	Cvperson. Death Place_ Cvlocation. Cvlocation
BirthPlaceCvlocation	Cvperson. Birth Place_ Cvlocation. Cvlocation
Person	Cvperson. Person
Jurisdiction	Jurisdiction. Details
JurisdictionID	Jurisdiction. Jurisdiction Identifier. Identifier
Name	Jurisdiction. Name. Text

Figure 39 – Classes from the ISA Core Vocabularies

- For the Requesting Party, we have reused the UBL Party class.
- For the Business Legal Form information requirement we have reused the CvBusiness ISA Core Vocabulary class.



Component Name	Representation Term	Data Type Qualifier	Data Type	Associated Object Class Qualifier	Associated Object Class	Alternative Business Terms	Cardinality	Com T
BusinessActivityRegistrationRequest								ABIE
UBLVersionID	Identifier		Identifier. Type				0..1	BBIE
CustomizationID	Identifier		Identifier. Type				0..1	BBIE
ProfileID	Identifier		Identifier. Type				0..1	BBIE
ProfileExecutionID	Identifier		Identifier. Type				0..1	BBIE
RequestDate	Date		Date. Type				1	BBIE
RequestID	Identifier		Identifier. Type				1	BBIE
PointSingleContactID	Identifier		Identifier. Type				0..1	BBIE
CompanyActivity	Company Activity				Company Activity		1..n	ASBI
RequestingParty	Party				Party		1	ASBI
Cvbusiness	Cvbusiness				Cvbusiness		1	ASBI
								END

Figure 40 – New main e-Document spreadsheet

We have also reused the UBL Version and document metadata basic information entities such as the Customization Identifier and the Profile Identifier, commonly used in UBL Schemas.

STEP 2- Setup the export filter in OpenOffice

The OpenOffice 3 file has to be exported to a Genericode file.

Install the open-source OpenOffice spreadsheet export to Genericode subset[59] 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. Below there is a summary of the steps to install the filter:

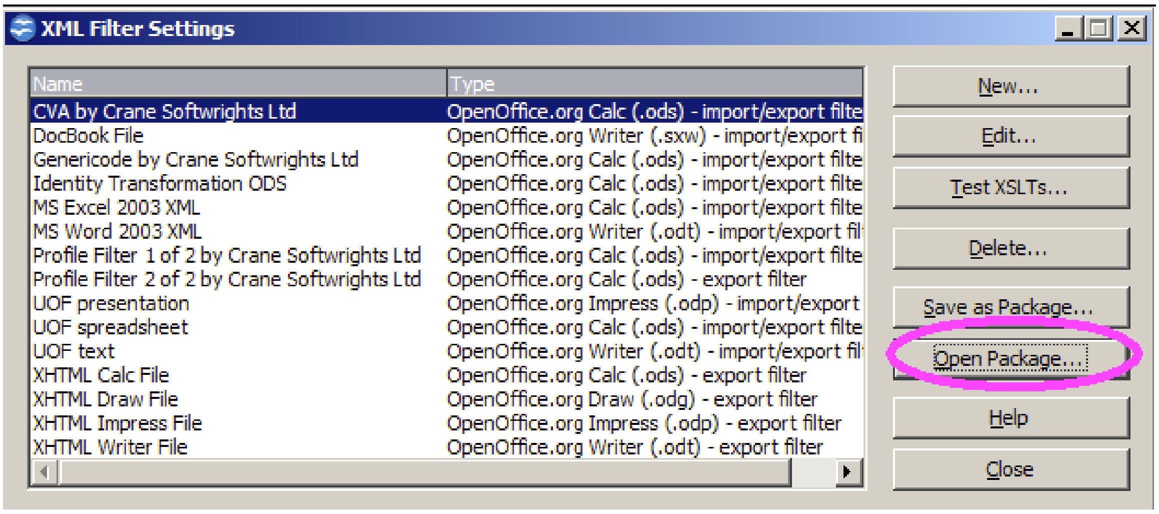
Uninstall the installed version of these filters

It is recommended to first uninstall any old version of the filter before installing a new one. See the OpenOffice spreadsheet export to Genericode subset Readme to learn how to uninstall a filter.

Install the filter

Start OpenOffice 3 and open a new document or spreadsheet. Click the menu item "Tools / XML Filter Settings..." to get to the filter dialogue.

The filter is installed using the following procedure. Press the button "Open Package..." without regard for any existing filter that may happen to be selected:

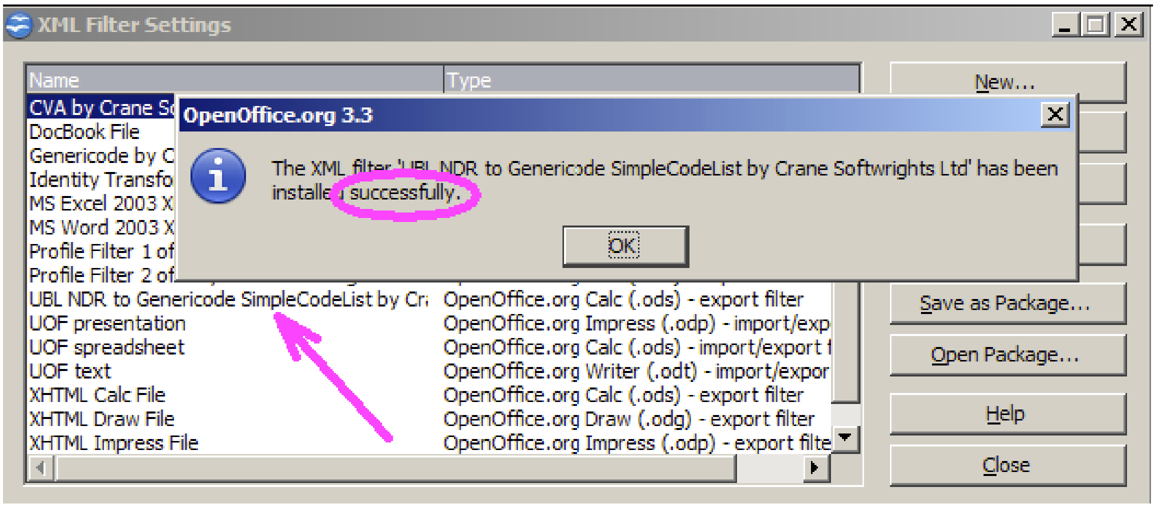


Name	Type
CVA by Crane Softwrights Ltd	OpenOffice.org Calc (.ods) - import/export filte
DocBook File	OpenOffice.org Writer (.sxw) - import/export fi
Genericode by Crane Softwrights Ltd	OpenOffice.org Calc (.ods) - import/export filte
Identity Transformation ODS	OpenOffice.org Calc (.ods) - import/export filte
MS Excel 2003 XML	OpenOffice.org Calc (.ods) - import/export filte
MS Word 2003 XML	OpenOffice.org Writer (.odt) - import/export fil
Profile Filter 1 of 2 by Crane Softwrights Ltd	OpenOffice.org Calc (.ods) - import/export filte
Profile Filter 2 of 2 by Crane Softwrights Ltd	OpenOffice.org Calc (.ods) - export filter
UOF presentation	OpenOffice.org Impress (.odp) - import/export
UOF spreadsheet	OpenOffice.org Calc (.ods) - import/export filte
UOF text	OpenOffice.org Writer (.odt) - import/export fil
XHTML Calc File	OpenOffice.org Calc (.ods) - export filter
XHTML Draw File	OpenOffice.org Draw (.odg) - export filter
XHTML Impress File	OpenOffice.org Impress (.odp) - export filter
XHTML Writer File	OpenOffice.org Writer (.odt) - export filter

Figure 41 – XML Filter Settings Dialogue

Navigate to the directory in which you unzipped the distribution file and select Crane-gcExportSubset.jar to add the "UBL NDR to Genericode SimpleCodeList by Crane Softwrights Ltd" filter to your installation and report successful operation:

[59] http://www.cranesoftwrights.com/resources/ubl/#gcExportSubset



The XML filter 'UBL NDR to Genericode SimpleCodeList by Crane Softwrights Ltd' has been installed successfully.

Name	Type
CVA by Crane S(	
DocBook File	
Genericode by C	
Identity Transfo	
MS Excel 2003 X	
MS Word 2003 X	
Profile Filter 1 of	
Profile Filter 2 of	
UBL NDR to Genericode SimpleCodeList by Cr	OpenOffice.org Calc (.ods) - export filter
UOF presentation	OpenOffice.org Impress (.odp) - import/exp
UOF spreadsheet	OpenOffice.org Calc (.ods) - import/export f
UOF text	OpenOffice.org Writer (.odt) - import/expor
XHTML Calc File	OpenOffice.org Calc (.ods) - export filter
XHTML Draw File	OpenOffice.org Draw (.odg) - export filter
XHTML Impress File	OpenOffice.org Impress (.odp) - export filte

Figure 42 – XML Fitler Successful Installation Dialogue

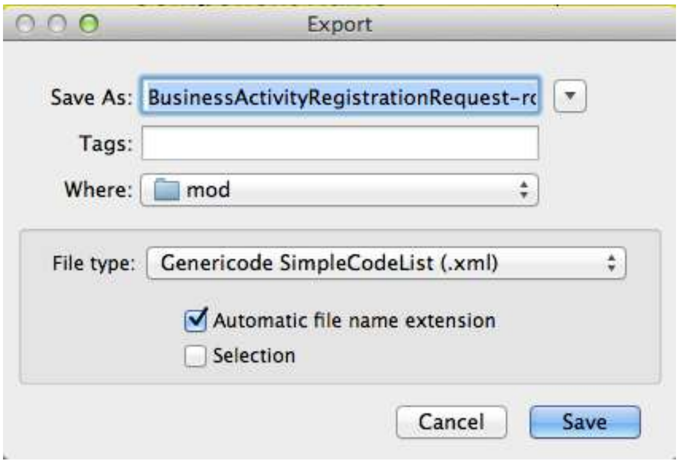
The filter is now successfully installed in your OpenOffice.

STEP 3- Export a Genericode Subset file from OpenOffice

Once we have the correct spreadsheet model, we have to use the OpenOffice spreadsheet export to genericode subset filter from Crane Softwrights to produce a Genericode file from the OpenOffice document. Use the "File / Export …" funtion to open the export dialogue.

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

In our pilot, we have created the BusinessActivityRegistrationRequest-rows.xml file.



Save As: BusinessActivityRegistrationRequest-r(

Tags:

Where: mod

File type: Genericode SimpleCodeList (.xml)

☑ Automatic file name extension

☐ Selection

Figure 43 – Export a Genericode file

STEP 4- Setup the configuration file

Now we have to properly setup the configuration file to run the XSD production script.

There is a configuration file and a launch file. We have created our own files to launch and run the production script.

- createbarr.bat / createbarr.sh – File to launch the generation process.
- config-barr.xml – File with the setup data to generate the XSD schemas.

Launching file

The createbarr batch file has the following commands:

```
echo ISA Programme additional documents...

echo ...checking...

java -jar ../saxon9he.jar -s:mod/BusinessActivityRegistrationRequest-
Entities.gc -xsl:../Crane-ublndrChecker.xsl -o:junk.out configuration-
uri=../config-barr.xml common-config-uri=../config-ubl-2.1.xml common-gc-
uri=UBL-Entities-2.1-PRD2-fix.gc

if [ "$?" != "0" ]; then exit ; fi


echo ...building business activity registration request document...

java -jar ../saxon9he.jar -s:mod/BusinessActivityRegistrationRequest-
Entities.gc -xsl:../Crane-gc2ublndr.xsl -o:junk.out qdt-as-cva=yes
configuration-uri=../config-barr.xml common-config-uri=../config-ubl-2.1.xml
common-gc-uri=UBL-Entities-2.1-PRD2-fix.gc aabie-prefix=barr
```

It uses the Java saxon9he.jar engine to build the XSD schema file using the BusinessActivityRegistrationRequest Genericode file and the config-barr.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 config-ubl-2.1.xml and the UBL-Entities-2.1-PRD2-fix.gc files are used to locate the UBL common vocabulary.

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.

The launching file starts checking the Genericode file provided as input and then creates the schema following the configuration and with the “barr” namespace prefix.

Configuration file

The configuration file is called config-barr.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               version="1">
    <copyright position="end">
  Test copyright statement.
    </copyright>
    <type-documentation>
      <ccts:Component xmlns:ccts="urn:un:unece:uncefact:documentation:2">
        <ccts:ComponentType>ComponentType</ccts:ComponentType>
        <ccts:DictionaryEntryName>DictionaryEntryName</ccts:DictionaryEntryName>
        <ccts:Definition>Definition</ccts:Definition>
        <ccts:Examples>Examples</ccts:Examples>
        <ccts:Cardinality>Cardinality</ccts:Cardinality>
        <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:AlternativeBusinessTer
ms>
      </ccts:Component>
    </type-documentation>
    <dir name="xsd" runtime-name="xsdrt">
      <dir name="mydoc">
        <file type="AABIE" name="BusinessActivityRegistrationRequest.xsd"
              abie="BusinessActivityRegistrationRequest"
              prefix="barr" sabie-prefix="cva" sbbie-prefix="cvc"
              namespace="urn:X-MyCompany:xsd:BusinessActivityRegistrationRequest">
        <comment>
  Library:           Business Activity Registration Request  document
  Module:            %f
  Generated on:      %z
</comment>
        </file>
        <file type="SABIE" name="CoreVocabularyAggregateComponents.xsd"
              prefix="cva"
namespace="http://www.w3.org/ns/corevocabulary/AggregateComponents">
        <comment>
  Library:           Core Vocabulary Common Aggregate Components
  Module:            %f
  Generated on:      %z
</comment>
        </file>
        <file type="SBBIE" name="CoreVocabularyBasicComponents.xsd"
              prefix="cvc"
              namespace="http://www.w3.org/ns/corevocabulary/BasicComponents">
        <comment>
```

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.

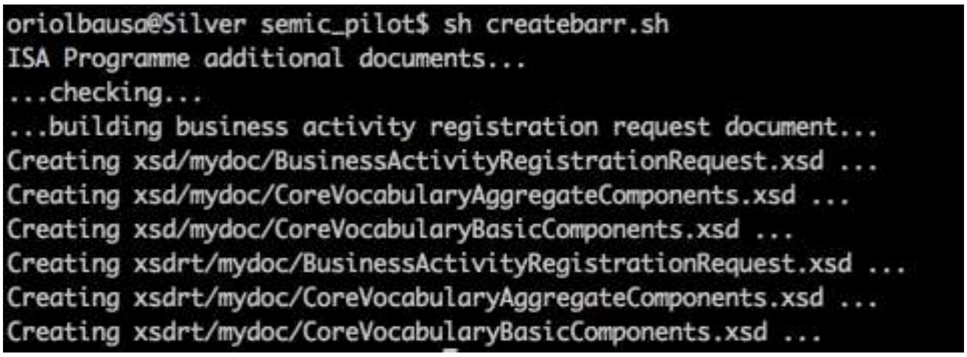
3- A **directory** section to describe the directories where the generated files have to be located.

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.

STEP 5- 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 semic_pilot$ sh createbarr.sh
ISA Programme additional documents...
...checking...
...building business activity registration request document...
Creating xsd/mydoc/BusinessActivityRegistrationRequest.xsd ...
Creating xsd/mydoc/CoreVocabularyAggregateComponents.xsd ...
Creating xsd/mydoc/CoreVocabularyBasicComponents.xsd ...
Creating xsdrt/mydoc/BusinessActivityRegistrationRequest.xsd ...
Creating xsdrt/mydoc/CoreVocabularyAggregateComponents.xsd ...
Creating xsdrt/mydoc/CoreVocabularyBasicComponents.xsd ...
```

Figure 44 – Create the XSD Schemas

When there are no errors in the checking phase, the script generates the following XSD Schemas:

- xsd/mydoc/BusinessActivityRegistrationRequest.xsd – Main document XSD Schema with comments following the CCTS.
- xsd/mydoc/CoreVocabularyAggregateComponents.xsd – Core Vocabulary XSD of aggregated components with comments following the CCTS.
- xsd/mydoc/CoreVocabularyBasicComponents.xsd – Core Vocabulary XSD of basic components with comments following the CCTS.
- xsdrt/mydoc/ BusinessActivityRegistrationRequest.xsd - Main document XSD Schema without comments
- xsdrt/mydoc/CoreVocabularyAggregateComponents.xsd - Core Vocabulary XSD of aggregated components without comments
- xsdrt/mydoc/CoreVocabularyBasicComponents.xsd - Core Vocabulary XSD of basic components without comments

The XSD Schemas follow the UBL naming and design rules.

ANNEX II CAMSS ASSESSMENT CRITERIA

CAMSS is a method for the assessment of technical specifications and standards in the field of ICT. It consists among others of an assessment process and a set of neutral and unbiased assessment criteria [IDABC Programme, 2012][60] according to seven categories. This annex enumerates the 51 CAMSS assessment criteria which are grouped into 7 categories.

I) Applicability

- **A.1 Interoperability**: Does the technical specification or standard address and facilitate interoperability between public administrations?
- **A.2 e-Government**: Does the technical specification or standard address and facilitate the development of eGovernment?
- **A.3 Defined requirements**: Are the functional and non-functional requirements for the use and implementation of the technical specification or standard clearly defined?
- **A.4 Reusability**: Is the technical specification or standard applicable and extensible for implementations in different domains?
- **A.5 Added value**: Does the technical specification or standard provide sufficient added value compared to alternative technical specification or standards in the same area of application?
- **A.6 Compatibility**: Is the technical specification or standard largely compatible with related (not alternative) technical specification or standards in the same area of application?
- **A.7 Dependencies**: Is the technical specification or standard largely independent from specific vendor products or products of single providers (either open source or proprietary)?
- **A.8 Dependencies**: Is the technical specification or standard largely independent from specific platforms or technologies?

II) Maturity

- **A.9 Development**: Has the technical specification or standard been sufficiently developed and in existence for a sufficient period to overcome most of its initial problems?
- **A.10 Conformity**: Are there existing or planned mechanisms to assess conformity of the implementations of the technical specification or standard (e.g. conformity tests, certifications)?
- **A.11 Quality**: Has the technical specification or standard sufficient detail, consistency and completeness for the use and development of products?

[60] Further information on CAMSS can be found in the dedicated CAMSS community on Joinup: https://joinup.ec.europa.eu/community/camss/description

- **A.12 Guidelines**: Does the technical specification or standard provide available implementation guidelines and documentation for the implementation of products?
- **A.13 Reference implementation**: Does the technical specification or standard provide a reference (or open source) implementation?
- **A.14 Backward compatibility**: Does the technical specification or standard address backward compatibility with previous versions?
- **A.15 Underlying technologies**: Have the underlying technologies for implementing the technical specification or standard been proven, stable and clearly defined?

III) Openness

- **A.16 Standardisation organisation**: Is information on the terms and policies for the establishment and operation of the standardisation organisation publicly available?
- **A.17 Development process**: Is participation in the creation process of the technical specification or standard open to all relevant stakeholders (e.g. organisations, companies or individuals)?
- **A.18 Standardisation process**: Is information on the standardisation process publicly available?
- **A.19 Decision making process**: Information on the decision making process for approving technical specification or standards is publicly available?
- **A.20 Consensus**: Are the technical specification or standards approved in a decision making process which aims at reaching consensus?
- **A.21 Review process**: Are the technical specification or standards reviewed using a formal review process with all relevant external stakeholders (e.g. public consultation)?
- **A.22 Participation stakeholders**: All relevant stakeholders can formally appeal or raise objections to the development and approval of technical specification or standards?
- **A.23 Documentation development and approval process**: Relevant documentation of the development and approval process of technical specification or standards is publicly available (e.g. preliminary results, committee meeting notes)?
- **A.24 Documentation implementation**: Is the documentation of the technical specification or standard publicly available for implementation and use for free or on reasonable terms?

IV) Intellectual Property Rights

- **A.25 IPR Documentation**: Is the documentation of the IPR for technical specification or standards publicly available (is there a clear and complete set of licence terms)?
- **A.26 (F)RAND licensing**: Is the technical specification or standard licensed on a (F)RAND basis?
- **A.27 Royalty-free licensing**: Is the technical specification or standard licensed on a royalty-free basis?

V) Market support

- **A.28 Implementations by different vendors/suppliers**: Has the technical specification or standard been used for different implementations by different vendors/suppliers?
- **A.29 Implementations in different industries/sectors**: Has the technical specification or standard been used in different industries, business sectors or functions?
- **A.30 Adoption**: Do the products that implement the technical specification or standard have a significant market share of adoption?
- **A.31 Users**: Do the products that implement the technical specification or standard target a broad spectrum of end-users?
- **A.32 Interest groups** *(Not applicable)*: Has the technical specification or standard a strong support from different interest groups?

VI) Potential

Assessment criteria A.33 to A.40 on the “impact” of the technical specification are not considered relevant in the context of this study. The impact addresses the consequences of the use and adoption of the e-Document engineering method, and is strongly related to the benefits of the use of standard e-Document engineering methods described in Section 5.1.

- **A.33 Impact efficiency and effectiveness**: Is there evidence that the adoption of the technical specification or standard supports improving efficiency and effectiveness of organisational process?
- **A.34 Impact migration**: Is there evidence that the adoption of the technical specification or standard makes it easier to migrate between different solutions from different providers?
- **A.35 Impact environment** *(not applicable)*: Is there evidence that the adoption of the technical specification or standard positively impacts the environment?
- **A.36 Impact financial costs**: Is there evidence that the adoption of the technical specification or standard positively impacts financial costs?

- **A.37 Impact security** *(not applicable)*: Is there evidence that the adoption of the technical specification or standard positively impacts security?
- **A.38 Impact privacy** *(not applicable)*: Is there evidence that the adoption of the technical specification or standard positively impacts privacy?
- **A.39 Impact administrative burden**: Is there evidence that the adoption of the technical specification or standard positively impacts the administrative burden?
- **A.40 Impact accessibility and inclusion**: Is there evidence that the adoption of the technical specification or standard positively impacts the accessibility and inclusion?
- **A.41 Risks**: Are the risks related to the adoption of the technical specification or standard acceptable?
- **A.42 Maintenance organisation**: Does the technical specification or standard have a defined maintenance organisation?
- **A.43 Maintenance resources** *(not applicable)*: Does the maintenance organisation for the technical specification or standard have sufficient finances and resources to be sure of freedom from short- to medium-term threats?
- **A.44 Maintenance process**: Does the technical specification or standard have a defined maintenance and support process?
- **A.45 Version management**: Does the technical specification or standard have a defined policy for version management?

VII) Coherence

- **A.46 International standard**: Is the standard an international standard or does it comply with relevant international standards?
- **A.47 Existing European standard**: Are there existing European Standards which cover the same areas as the technical specification or standard being assessed?
- **A.48 Obsolete European standard** *(not applicable)*: Are there obsolete European Standards which cover the same areas as the technical specification or standard being assessed? (i.e. an obsolete European standard is a standard, which is no longer relevant and not used on the ICT market as a newer IT solution exists)
- **A.49 Technical specification under consideration** *(not applicable)*: Does the technical specification or standard cover areas different from areas addressed by technical specifications being under consideration to become a European standard? (i.e. technical specifications provided by a non-formal standardisation organisation, that is other than CEN, CENELEC or ETSI can be under consideration to become a European standard or alternatively an identified technical specification)
- **A.50 Recommendation status**: Is the standard or specification listed as

recommended in at least one Member State?

- **A.51 Mandatory status**: Is the standard or specification listed as mandatory in at least one Member State?

ANNEX III ASSESSMENT E-DOCUMENT ENGINEERING METHODS BASED ON THE CAMSS CRITERIA

This annex assesses the three standard e-Document engineering methods identified in chapter 3 according to (a relevant subset of) the assessment criteria in the **Common Assessment Method of Standards and Specifications** (CAMSS). CAMSS is a method for the assessment of technical specifications and standards in the field of ICT. It consists among others of an assessment process and a set of neutral and unbiased assessment criteria [IDABC Programme, 2012]. Annex III gives a detailed listing of all 51 assessment criteria. However, in the context of this study, not all assessment criteria are applicable or relevant. Such criteria, for instance, those referring to the "*impact*" of the technical specification, are withheld from the evaluation.

In this section, we will assess the following two standard e-Document engineering methods according to the CAMSS assessment criteria:

1. UN/CEFACT e-Document engineering method;
2. OASIS UBL method;

and the following e-Document engineering method to bind information requirements to existing syntaxes:

3. CEN BII e-Document engineering method.

1.1. UN/CEFACT e-Document engineering method

1.1.1. Applicability

- **A.1 Interoperability – A.2 e-Government**: UN/CEFACT aims to facilitate the exchange of recommendations and electronic business standards.
- **A.3 Defined requirements**: the functional and non-functional requirements for the use and implementation of the e-Document engineering method are clearly defined in an ensemble of documents outlined in Section 3.1.
- **A.4 Reusability**: The UN/CEFACT CCTS is applicable outside the UN/CEFACT e-Document formats. However, the CCBDA method is not yet being reused.
- **A.6 Compatibility**: Standards of UN/CEFACT are compatible with the methods of OASIS UBL and CEN BII.
- **A.7 Dependencies**: The underlying technologies of UN/CEFACT (XSD, XML, and UML) are available under an open licence and independent from specific vendor products.
- **A.8 Dependencies**: The standards of UN/CEFACT are developed in a technology-neutral way.

1.1.2. Maturity

- **A.9 Development**: The Core Component Library and associated e-Document formats based on XML have been continuously revised twice per year since 2006. It is not known how much implementations these have. The technical specifications (CCTS, UMM, NDR) have also been revised several times.
- **A.10 Conformity**: UN/CEFACT has its own validation team to check whether everything that is produced conforms to their standards.
- **A.11 Quality**: UN/CEFACT can be considered as complete and consistent.
- **A.12 Guidelines**: UN/CEFACT provides a commonly agreed method with an implementation guideline.
- **A.13 Reference implementation**: A reference implementation is not available.
- **A.14 Backward-compatibility**: The bi-annual revision policy means that non-compatible new versions are issued every 6 months. Historical versions are always available.
- **A.15 Underlying technologies**: The underlying technologies related to UN/CEFACT's e-Document engineering method are based on XML, UML and XSD. All these technologies are relatively stable and are clearly defined.

1.1.3. Openness

- **A.16 Standardisation organisation**: The UN/CEFACT Bureau ensures that "*UN/CEFACT activities comply with its mandate, terms of reference and procedures*" [UN/CEFACT, 2011].
- **A.17 Development process**: Designated experts to their national delegations can participate to the development process of UN/CEFACT. They must first send an application to the head of delegation of the Member State, who will nominate the participating experts. Experts can come from governmental organisations, non-governmental organisations, industrial organisations, public administrations, business administrations, etc. or be independent individuals.
- **A.18 – A.19 Standardisation and decision making process**: Everything concerning the standardisation and decision making process follows the Open Development Process and is monitored by the UN/CEFACT Bureau.
- **A.20 Consensus**: The Open Development Process (ODP) aims at working collaboratively and effectively [UN/CEFACT, 2011].
- **A.21 Review process**: "*All projects concerned with the development of UN/CEFACT deliverables within the UN/CEFACT Programme of Work need to follow a set of ODP stages related to their deliverable's publication type*" [UN/CEFACT, 2011]. One of those ODP stages includes a public review. Public review is mandatory for deliverables concerning UN/CEFACT Business Standards and UN/CEFACT Technical Standards.

- **A.22 Participation stakeholders**: Stakeholders are free to participate and comment on the development of a project [UN/CEFACT, 2011].
- **A.23 Documentation development and approval process:** Documentation concerning the Open Development Process is publicly available on the website of UNECE[61].
- **A.24 Documentation implementation**: Documentation concerning the UN/CEFACT standards is publicly available on the website of UN/CEFACT[62].

1.1.4. Intellectual property rights

- **A.25 IPR Documentation**: "*The UN/CEFACT IPR Policy is designed to promote the goal of enabling the implementation of UN/CEFACT Specifications without the burden of fees or restrictions. The Policy promotes this goal, by requiring all Participants in UN/CEFACT work to waive their rights to enforce any of their intellectual property that would be necessary to implement or use a Specification developed in that work*" [UN/CEFACT, 2012]. The UN/CEFACT Intellectual Property Rights Policy is publicly available on the website of UNECE[63]. However, UN/CEFACT's technical specifications do not contain a clear and complete set of licence statements. The specifications fall under the UN/CEFACT copyright, implicating the application of UN liability, trademark and document use rules.
- **A.26 (F)RAND licensing:** UN/CEFACT only approves specifications that are available without fees or restrictions UN/CEFACT [UN/CEFACT, 2006], meaning that the licensing is fair, reasonable and non-discriminatory.
- **A.27 Royalty-free licensing**: All specifications[64] published are free of rights: "*In order to promote the widest adoption of Specifications, UN/CEFACT seeks to issue Specifications that can generally be implemented without fees or restrictions*" [UN/CEFACT, 2012]. The UN has a common IPR policy that requires all Participants in UN/CEFACT work to waive their rights to enforce any of their intellectual property.

1.1.5. Market support

- **A.28 Implementations by different vendors/suppliers - A.29 Implementations in different industries/sectors:** There are implementations of UN/CEFACT e-Documents in several sectors such as

[61] Open Development Process for UN/CEFACT: http://www.unece.org/fileadmin/DAM/cefact/cf_plenary/plenary12/ECE_TRADE_C_CEFACT_2010_24_Rev.2E_ODP_revised.pdf

[62] UN/CEFACT: http://www.unece.org/cefact.html

[63] UN/CEFACT IPR Policy: http://www.unece.org/fileadmin/DAM/cefact/cf_plenary/plenary06/trd_cf_06_11e.pdf

[64] Updated UN/CEFACT Intellectual Property Rights Policy (UN/CEFACT, 2012): "'***Specification'**, as used in this Policy encompasses all documents and drafts that are developed or are being developed under the Open Development Process as part of a Project's deliverables."*

construction, agriculture or tourism. The e-Document engineering method is basically used within the UN/CEFACT organization.

- **A.30 Adoption:**. No analysis of the market share was conducted. However, the users listed for A.31 seem to indicate a significant market share.
- **A.31 Users**: In general, users are organisations that participate in UN/CEFACT projects.
    - OAGI have developed implementation validation samples of e-Document formats based on UN/CEFACT CCTS 3.0 and UN/CEFACT NDR 3.0
    - *Direct users of e-Documents formats*:
        - French Construction sector (e-tendering and CFEM);
        - Japanese construction sector (e-tendering);
        - French Accounting industry association (Accounting entry);
        - China/Australia/NZ (Phytosanitary export certification specification);
        - Dutch agriculture sector (Crop Data Sheet);
        - US Defence Contract Management Agency (Project Schedule and Cost Performance Management);
        - Japanese tourism (Small scale lodging house information);
        - e-Justice Communication via Online Data Exchange (e-CODEX) lage-scale pilot; and
        - Modèle de données communies (MDC).
    - *Schema production (partial re-use of the Core Component Library)*: European Insurance association.

1.1.6. Potential

- **A.33 Impact efficiency and effectiveness**: The use of standard methods contributes to efficiency and effectiveness, as argued in Section 5.1.
- **A.34 Impact migration**: The use of standard methods enables the use of standard tooling that can become interchangeable, or that make it easier to migrate from one solution to another, as argued in Section 5.6.
- **A.36 Impact financial costs**: The use of standard methods allows the use of standard tooling and services reducing vendor lock-in.
- **A.41 Risks**: There are no risks identified related to the adoption of this standard.
- **A.42 Maintenance organisation:** The maintenance is organized by the Working Groups and the Bureau.
- **A.43 Maintenance resources:** the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) was established, as a subsidiary, intergovernmental body of the United Nations Economic

Commission for Europe (UNECE) Committee on Trade. No information maintenance resources were collected, but they can be deemed sufficient to be free of short-term and medium-term treads.

- **A.44 Maintenance process:** All work is done by project groups ("project domain area experts"). If there are any change requests, this is processed through an open procedure.
- **A.45 Version management:** UN/CEFACT has a defined policy for version management included in the UN/CEFACT Open Development Process (ODP)[65] for technical specifications.

1.1.7. *Coherence*

- **A.46 International standard**: Within the United Nations framework of the Economic and Social Council, the United Nations Economic Commission for Europe (UNECE) serves as the focal point for trade facilitation recommendations and electronic business standards, covering both commercial and government business processes that can foster growth in international trade and related services. In this context, the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) was established, as a subsidiary, intergovernmental body of the UNECE Committee on Trade, mandated to develop a program of work of global relevance to achieve improved worldwide coordination and cooperation in these areas.
- **A.47 Existing European standard**: There are no European standards – produced by the European Standards bodies ETSI, CEN, or CENELECT – that constitute or related to an e-Document engineering method.
- **A.50 Recommendation status:** The use of the *method* has no recommendations status in EU Member States. Nonetheless, e-Document *formats* created with the method can be subject to the "comply-or-explain" approach, as is the case for some Member States (e.g. The Netherlands).
- **A.51 Mandatory status**: There is no evidence that the UN/CEFACT method has a mandatory status in at least one Member State.

[65] UN/CEFACT Open Development Process (ODP) for technical specifications http://www1.unece.org/cefact/platform/display/public/Open+Development+Process+%28ODP%29+for+TMG

1.2. OASIS UBL method

1.2.1. Applicability

- **A.1 Interoperability - A.2 e-Government**: UBL 2.1 aims at achieving data operability in the global electronic business environment.
- **A.3 Defined requirements**: the functional and non-functional requirements for the use and implementation of the e-Document engineering method are clearly defined in the documents outlined in Section 3.2..
- **A.4 Reusability**: The OASIS UBL method is considered to be modular, reusable and extensible. It has among others been used to generate the schemas of the ISA Core Vocabularies[66].
- **A.6 Compatibility**: The OASIS UBL method is largely compatible with UN/CEFACT Standards (CCTS, CCL) and CEN BII.
- **A.7 – A.8 Dependencies**: UBL is developed in an open and vendor-neutral process, where the entire user community can participate [OASIS, 2014]. Underlying technologies, such as XML and XSD are available in open source.

1.2.2. Maturity

The e-Document engineering method maturity addresses the following topics:

- **A.9 Development**: UBL was first released as an OASIS Standard in 2004. Since then, two version upgrades have been published. UBL is considered one of the most mature and implemented OASIS Standards [OASIS, 2014].
- **A.10 Conformity**: The UBL 2.0 Guidelines for Customization define conformity concerning customization and re-use of UBL.
- **A.11 Quality:** Evidence in the fact that the technical specification has sufficient detail, consistency and completeness for the use and development of products can be found in the already developed software tools to support the UBL method.
- **A.12 Guidelines**: Under the OASIS UBL 2.0 Standard, guidelines are publicly available concerning the naming and design[67] of the UBL schemas and the customization[68] [OASIS, 2014].
- **A.13 Reference implementation**: The OASIS UBL 2.1 standard does not provide a reference implementation, but does provide sample document instances.

[66] The XML Schema of the Core Vocabularies follows the UBL XML Naming and Design Rules. https://joinup.ec.europa.eu/community/semic/og_page/studies#core-vocabularies

[67] UBL 2.0 Naming and Design Rules: http://docs.oasis-open.org/ubl/cs01-UBL-2.0-NDR/cs01-UBL-2.0-NDR.pdf

[68] UBL 2 Guidelines for Customization: http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.pdf

- **A.14 Backward compatibility**: UBL 2.1 offers complete backward compatibility with UBL 2.0. Any document that validates against a UBL 2.0 schema will also validate against the UBL 2.1 version of that schema. However, UBL 2.x versions are not backward-compatible with UBL 1.0 [OASIS, 2013].
- **A.15 Underlying technologies**: Underlying technologies related to the OASIS UBL TC method, including XML, XSD and Schematron, have been proven stable and clearly defined.

1.2.3. Openness

In order for the method to be relevant for adoption, the standardization organisation, the decision-making process and the documentation should be sufficiently open and available.

- **A.16 Standardisation organisation**: UBL is developed in an open OASIS Technical Committee, including the participation of various industry data standards organisations.
- **A.17 Development process**: Participation in the development process of the UBL Standard is open for anyone, by joining OASIS[69] and signing up for the UBL Technical Committee [OASIS, 2014].

  A.18 Standardisation process: The standardisation process of the OASIS Standards is a part of the TC Process and outlined under Section 3 Approval Process [OASIS, 2013]: *"Standards Track Work Products progress as follows:*
  - *Committee Specification Draft;*
  - *Committee Specification Public Review Draft;*
  - *Committee Specification;*
  - *Candidate OASIS Standard;*
  - *OASIS Standard; and*
  - *Approved Errata."*
- **A.19 Decision making process**: The decision-making process within OASIS is publicly described in the *Technical Committee (TC) Process*[70].
- **A.20 Consensus**: For the UBL OASIS Standard, a collaboration of all parties affected by cross-industry standards was involved in the development and requirements phase, leading to a consensus of the standard [OASIS, 2004].
- **A.21 Review process**: The Candidate OASIS Standard is subject to a public review of at least 60 days to indicate the readiness for publication. The public review is announced to the OASIS Membership list and is also open for comments of non-Technical Committee Members.

[69] Joining OASIS: https://www.oasis-open.org/join/

[70] Technical Committee (TC) Process: https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess

- **A.22 Participation stakeholders**: Once the public review is completed, an organizational voting allows the approval of the specification as an OASIS Standard. Each OASIS Organizational Member is entitled to cast one vote.
- **A.23 Documentation development and approval process**: Committee Notes taken during the approval process may be made available as informative and explanatory documentation [OASIS, 2013].
- **A.24 Documentation implementation**: Documentation on the OASIS Standard of UBL 2.1[71] is made publicly available, along with all meeting minutes, discussion notes and technical work[72] produced by the UBL TC. UBL 2.1 is available as HTML file, PDF file and XML file. A ZIP file contains all the files in the UBL 2.1 release.

1.2.4. *Intellectual property rights*

This section addresses the licences that allow the implementation of the engineering method.

- **A.25 IPR Documentation:** The documentation on the OASIS Intellectual Property Rights Policy is publicly available[73]. The OASIS IPR Policy addresses the intellectual property in the production of the OASIS Standards and ensures that these standards can be implemented with confidence.
  The copyright notice of OASIS applies to the UBL method and states the following: "*Copyright (C) OASIS Open 2001-2013. All Rights Reserved.*
  This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.
  The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

[71] *Universal Business Language Version 2.1* 04 November 2013. OASIS Standard:
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.xml
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.pdf
http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.zip

[72] Technical Work produced by the Committee:
https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl#technical

[73] OASIS Intellectual Property Rights (IPR) Policy: https://www.oasis-open.org/policies-guidelines/ipr

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

- **A.26 (F)RAND licensing**: The above mentioned licensing terms and conditions are royalty-free.
- **A.27 Royalty-free licensing:** The UBL TC operates under the "*Royalty Free on Limited Terms*" IPR mode [OASIS, 2014], implicating reasonable terms to the operation and maintenance of the licence relationship [OASIS, 2014].

1.2.5. Market support

The market support is assessed according to the operational implementations, the market share and demand, and the user support.

- **A.28 Implementations by different vendors/suppliers** - **A.29 Implementations in different industries/sectors**: OASIS UBL has widely been adopted for its usage in the European public procurement framework, including the public sector invoicing in Denmark, EHF (Norway), Svefaktura (Sweden), ePrior (European Commission DIGIT), CODICE (Spain) and PEPPOL. Other implementations for public and private sector eInvoicing include E-Fatura (Turkey), Factura Electronica (Peru), SimplerInvoicing (the Netherlands), and Tradeshift (global). UBL is also fundamental in the transport domain: eFreight (European Commission - DG MOVE), DTTN (Port of Hong Kong), TradeNet (Port of Singapore), Electronic Freight Management (US Department of Transportation), and Freightgate (logistics services) [OASIS, 2014].
- **A.30 Adoption**: The European e-Government Core Vocabularies are based on the OASIS UBL TC method, particularly the XML Naming and Design Rules of the OASIS UBL TC. Along with the implementations mentioned above, the market share of adoption is significant.
- **A.31 Users:** US Department of the Navy, New Zealand Education Department, PEPPOL project, DG Move e-freight common framework, eBIZ-TCF, Tradeshift.

1.2.6. Potential

The potential of the method is described according to its impact, consequences and evolution.

- **A.33 Impact efficiency and effectiveness**: The use of standard methods contributes to efficiency and effectiveness, as argued in Section 5.1.
- **A.34 Impact migration**: The use of standard methods enables the use of standard tooling that can become interchangeable, or that make it easier to migrate from one solution to another, as argued in Section 5.6.

- **A.36 Impact financial costs**: The use of standard methods allows the use of standard tooling and services reducing vendor lock-in.
- **A.41 Risks**: There are no risks identified related to the adoption of this standard.
- **A.43 Maintenance resources:** No information maintenance resources were collected, but they can be deemed sufficient to be free of short-term and medium-term treads.
- **A.42 Maintenance organisation:** The OASIS UBL Technical Committee is responsible for the maintenance and future developments of the OASIS UBL Method. The OASIS UBL Technical Committee consists of technical and business experts. [OASIS, 2013].
- **A.44 Maintenance process:** The maintenance process of the OASIS UBL method is not explicitly mentioned.
- **A.45 Version management**: The OASIS UBL method makes use of version identifiers through the development phases of the method.

1.2.7. Coherence

The UBL e-Document engineering methods are considered coherent with other European Standards.

- **A.46 International standard**: The OASIS UBL TC Method is the result of an international collaboration standardised by the industry consortium formed by the members of OASIS. UBL has been submitted for assessment by the Multi Stakeholder Forum on ICT standardization; a positive assessment would allow UBL to be officially referred to in procurement specifications. Furthermore, the UN/CEFACT CCL covers some of the areas of the OASIS UBL 2.1 Standard. Furthermore, the method is coherent with other international standards such as XML Schema Definition (XSD), a de-facto standards by the World Wide Web Consortium (W3C).
- **A.47 Existing European standard**: There are no European standards – produced by the European Standards bodies ETSI, CEN, or CENELECT – that constitute or related to an e-Document engineering method.
- **A.50 Recommendation status**: The use of the *method* has no recommendations status in EU Member States, despite the fact that OASIS UBL e-Document formats are widely used and promoted in several EU Member States.
- **A.51 Mandatory status**: The use of the *method* has no recommendations status in EU Member States. In Denmark, Norway and Sweden, UBL e-Document formats are mandated for invoices to government agencies.

1.3. CEN BII method

1.3.1. Applicability

- **A.1 Interoperability - A.2 e-Government**: The mission of the CEN BII workshop is to spread and facilitate the use of e-procurement standards by suppliers and buyers, and especially public administrations.
- **A.3 Defined requirements**: the functional and non-functional requirements for the use and implementation of the e-Document engineering method are clearly defined in the documents outlined in Section 3.3.
- **A.4 Reusability**: Even though the method of the CEN BII workshop is targeted in the context of e-Procurement, the CEN BII method can be applied in a more generic context.
- **A.6 Compatibility**: The CEN BII initiative has used an e-Document engineering method that reuses and complements the methods of UN/CEFACT and the OASIS UBL Technical Committee.
- **A.7 – A.8 Dependencies**: BII specifications are neutral of any technical implementation and syntax.

1.3.2. Maturity

The e-Document engineering method maturity addresses the following topics:

- **A.9 Development**: The method is stable for almost four years. The results of the first CEN ISSS BII workshop were published under CWA 16703:2010 after its conclusion in 2009. The CWA was followed-up and improved in the second workshop, CEN BII2 in 2010 [CEN, 2012]. The follow-up of the CEN BII2 deliverables is the new CEN BII3 Workshop starting on 2013.
- **A.10 Conformity:** No conformance tests are applicable to the CEN BII *method*.
- **A.11 Quality**: CEN BII CWA 16558, which contains the e-Document engineering method, and its annexes contain sufficient detail. Although the given documentation of the method is targeted in the context of e-Procurement, the guidelines included in the CEN Workshop Agreement can be applied in a more generic context.
- **A.12 Guidelines:** The CEN Workshop Agreement CWA16558 BII2 Architecture[74] provides detailed guidelines for applying the CEN BII *method*.
- **A.13 Reference implementation**: There is no reference implementation available.
- **A.14 Backward compatibility:** Backward-compatibility is not applicable for the CEN BII method, as there was no previous version.

[74] http://www.cenbii.eu/deliverables/cwa-16558-bii-architecture/

- **A.15 Underlying technologies**: The XML Schemas for which CEN BII provides mappings are published by UN/CEFACT and UBL and have been proved stable and clearly defined as previously mentioned. GEFEG.FX, which is used to create these mappings, is also considered as a stable tool.

1.3.3. Openness

In order for the method to be relevant for adoption, the organisation, the decision-making process and the documentation should be sufficiently open and available.

- **A.16 Standardisation organisation**: CEN, the European Standardization Organization is an association that brings together the National Standardization Bodies of 33 European countries. It provides a platform for the development of European Standards and other technical documents in a wide range of fields and sectors.
- **A.17 Development process**: All participants who are interested and willing to join, from both the public and the private sector, are allowed to participate in the CEN BII workshop [CEN BII 2, 2013].
- **A.18 Standardisation process**: The CEN BII is a CEN Workshop Agreement (CWA). A CWA is an agreement developed and approved in a CEN Workshop. It is open to direct participation of anyone with an interest in the development of the agreement and it has no geographical limit on participation. A CWA does not have the status of a European Standard. It involves no obligation at national level. It may not conflict with any European Standard.
- **A.19 Decision making process**: The decision making process for changing the guidelines are described in the report on versioning and change management[75] in the CWA 16558 BII Architecture. The change management mechanism aims to capture and correct errors and inconsistencies, and to handle appropriate change requests. These modifications should be communicated towards the stakeholders [CEN, 2012].
- **A.20 Consensus**: The decision-making process and change management aim at reaching a consensus.
- **A.21 Review process**: The review of the CEN BII2 workshop helps to progress the deliverables towards their final publication. It is intended to make all interested parties aware of which deliverables will be published and allows them to comment [CEN BII 2, 2013].
- **A.22 Participation stakeholders**: Change requests are usually initiated by major stakeholders, through formalized requests or incident reports. These are processed by the Change Control Board (CCB).
- **A.23 Documentation development and approval process:** Documentation on the development process, including meeting of

[75] CWA 16558 – Annex O Report Versioning and Change Management Policy: ftp://ftp.cen.eu/public/CWAs/BII2/CWA16558/CWA16558-Annex-O-BII-Report-VersioningAndChangeManagement-V1_0_0.pdf

teleconference calls and face-to-face meetings, capturing issues and resolutions are captured and shared within the working teams. The CEN BII deliverables are published for general public revision after the internal review process has finalized. The internal review and public review processes are defined in the CEN process[76].

- **A.24 Documentation implementation**: Information concerning the CEN Workshop Agreement CWA16558 BII2 Architecture[77] is publicly available, along with their guidelines and reports.

1.3.4. Intellectual property rights

This section addresses the licences that allow the implementation of the engineering method.

- **A.25 IPR Documentation**: All rights of exploitation in any forms and by any means are reserved worldwide for CEN national members.

  This statement is present in the five CEN BII CEN Workshop Agreements, which have annexes. Even if the statement is not repeated in each Annex, the Annexes are part of the CWA itself, being then affected by the same copyright.

  CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

  Information on copyright, licenses and IPR for a CWA can be obtained from the CEN site[78].

- **A.26 (F)RAND licensing**: All CEN national members have the rights of exploitation.
- **A.27 Royalty-free licensing**: All CEN BII deliverables are provided using a royalty-free licensing.

1.3.5. Market support

The market support is assessed according to the operational implementations, the market share and demand, and the user support.

- **A.28 Implementations by different vendors/suppliers - A.29 Implementations in different industries/sectors**: To the best of our knowledge, the CEN BII2 method has not been used in different industries.

[76] http://boss.cen.eu/developingdeliverables/CWA/Pages/default.aspx

[77] CEN Workshop Agreement CWA16558 BII2 Architecture: http://www.cenbii.eu/deliverables/cwa-16558-bii-architecture/

[78] http://boss.cen.eu/REFERENCE%20MATERIAL/Pages/default.aspx

The CEN BII2 method uses several commonly used tools such as the GEFEG.FX tool, spreadsheet templates, Word documents, etc.

- **A.30 Adoption**: The method of CEN BII is used by PEPPOL, which has a significant "market-share" and amount of users in the Nordic countries. The guidelines are in considerable demand.
- **A.31 Users:** The CEN BII method is currently mostly being used in the e-Procurement community, in different countries, and in different projects, including open PEPPOL, e-PRIOR, and SFTI Sveorder.

1.3.6. Potential

The potential of the method is described according to its impact, consequences and evolution.

- **A.33 Impact efficiency and effectiveness**: The use of standard methods contributes to efficiency and effectiveness, as argued in Section 5.1.
- **A.34 Impact migration**: The use of standard methods enables the use of standard tooling that can become interchangeable, or that make it easier to migrate from one solution to another, as argued in Section 5.6.
- **A.36 Impact financial costs**: The use of standard methods allows the use of standard tooling and services reducing vendor lock-in.
- **A.41 Risks**: There are no risks identified related to the adoption of this standard.
- **A.42 Maintenance organisation:** The CEN BII method is currently being maintained by the workshop participants, however, there is no formal CEN Technical Committee (TC) that could guarantee its long-term sustainability and maintenance at this point [CEN, 2012].
- **A.44 Maintenance process**: There is no defined maintenance and support process available. The CEN BII work is maintained by CEN Workshop Agreements but there is not an organization set up to provide long-term maintenance and support.
- **A.45 Version management**: CEN Workhop Agreements do not have a a defined policy for version management. This means that there is no policy for the version management of the CEN BII *method*. The management of versioning of *e-Documents formats*, which impacts backward compatibility, dependencies and audit trails, is described and publicly available in the "Versioning and Change Management Policy"[79].

1.3.7. Coherence

The e-Document engineering methods are considered as coherent with other European Standards as long as they do not conflict with each other.

[79] Versioning and Change Management Policy: http://www.cenbii.eu/wp-content/uploads/Report-BII2-Versioning-and-Change-management.pdf

- **A.46 International standard**: CEN BII 2 is not a European Standard, as it does not follow the normal CEN standardisation procedures, but rather the lightweight procedures of the CEN Workshop Agreement. However, the CEN BII 2 method is coherent with UN/CEFACT CCTS and OASIS UBL as both XML syntaxes are used as the background of the CEN BII2. Furthermore, the CEN BII method is coherent with other international standards such as the Unified Modelling Language (UML), the Semantics of Business Vocabularies and Rules (SBVR) standard, Schematron, XSD, etc. These de-facto standards by international industry consortia such as the Object Management Group (OMG) and the World Wide Web Consortium (W3C).
- **A.47 Existing European standard**: There are no European standards – produced by the European Standards bodies ETSI, CEN, or CENELECT – that constitute or related to an e-Document engineering method.
- **A.50 Recommendation status**: There is no Member State that recommends the use of the CEN BII method.
- **A.51 Mandatory status**: There is no Member State that mandates the use of the CEN BII method.