Submitted to the EC on 25/07/2012

COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME
ICT Policy Support Programme (ICT PSP)

e-CODEX

e-Justice Communication via Online Data Exchange

ICT PSP call identifier: CIP-ICT-PSP-2009-4
ICT PSP main Theme identifier: CIP ICT PSP 2010 5.2 3: E-JUSTICE SERVICES
Project full title: e-Justice Communication via Online Data Exchange
Grant agreement n°: 270968

D6.3 Concept for Implementation of WP6

Deliverable Id :	D6.3
Deliverable Name :	Concept of Implementation of WP6
Status :	v1.0 final
Dissemination Level :	RE, PU
Due date of deliverable :	30. Apr 2012
Actual submission date :	26. Jul 2012
Work Package :	WP6
Organisation name of lead partner for this deliverable :	Ministry of Security and Justice the Netherlands
Author(s):	Enrico Francesconi, Fineke Beukema, Harry Biersteker, Ronald Smit
Partner(s) contributing :	IT, NL, DE, BE, PT, GR, MT, FR

Abstract:
This deliverable introduces the methodology for modelling (incl. modelling steps), refines requirements and gives functional specifications for the individual building blocks. An overview is given on what standards might be selected and short term choices are presented.

This document will serve as the basic WP6 reference to start the pilot implementation.

This is a living document which will be updated towards a long term sustainable solution.

History

Version	*Date*	*Changes made*	*Modified by*
0.5	03-April-2012	1st draft version for review	
0.9	25-April-2012	Comments from WP6 partners. Additional chapters.	NL
0.99	16-May-2012	First complete version. Comments from PD, EQM, MB and WP6 partners.	NL
1.0 draft	03-June-2012	Comments from PD, EQM, IQM, MB and WP6 partners. Work version	NL
1.0 final	25-July-2012	Final version	NL

Table of contents

List of Figures

List of Tables

List of Abbreviations

Acronym	*Explanation*
ACC	Aggregate Core Component
AEW	Airborne Early Warning
APIs	Application Programming Interfaces
BSCW	Basic Support for Cooperative Work
B2B	Business to Business
BB	Building Block
BIE	Business Information Entity
BRITE	Business Register Interoperability Throughout Europe
CAM	Content Assembly Mechanism
CBC	Cross Border Cooperation
CIP	Competitive and Innovation Programme
CCTS	Core Components Technical Specification
CC	Core Component
CSV	Comma-Separated Values
CODICE	Componentes y Documentos Interoperables para la Contratación Electrónica
DLP	Digital Light Processing
EAW	European Arrest Warrant
ebMS	ebXML Message Service
ebXML	Electronic Business using eXtensible Markup Language
EC	European Commission
ECLI	European Case Law Identifier
eID	electronic Identification
EIF	European Interoperability Framework
EPO	European Payment Order
EJN	European Judicial Network
epSOS	Smart Open Services for European Patients

Acronym	*Explanation*
EESSI	European electronic signature standardization initiative
ESTRELLA	European project for Standardized Transparent Representations
EU	European Union
EuroVoc	EuroVoc is a multilingual, multidisciplinary thesaurus covering the activities of the EU, the European Parliament in particular.
FRBR	Functional Requirements for Bibliographic Records
GJXDM	Global Justice XML Data Model
Guid	Globally unique identifier
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
JoinUp	Joinup is a new collaborative platform created by the European Commission and funded by the European Union via the Interoperability Solutions for Public Administrations (ISA) Programme.
ICT	Information and Communications Technology
IDABC	Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens.
IJIS	Integrated Justice Information Systems Institute
IRI	Internationalized Resource Identifier
ISA	Interoperability Solutions for European Public Administrations
ISO	International Organization for Standardization
LKIF	Legal Knowledge Interchange Format
LRI	Legal Response Initiative
LSP	Large Scale Pilot
MoJ	Ministry of Justice
MS	Member State
N/A	Not Applicable
NDR	Non Delivery Receipt
NDR	Naming and Design Rules
NES	Northern European Standard
NIEM	National Information Exchange Model
NIF	National Interoperability Framework
ODT	OASIS Open Document Format for Office Applications

Acronym	*Explanation*
OWL	Web Ontology Language
OWL-DL	Web Ontology Language-Description Logic
PDF	portable document format
PEC	Posta elettronica certificata
PEGS	Pan-European eGovernment Services
PEPPOL	Pan-European Public Procurement Online
PKI	Public Key Infrastructure
PSC	Point of Single Contact
RDF	Resource Description Framework
REM	Registered E-Mail
RQ	Requirement
RTF	Rich Text Format
S/MIME	Secure / Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SKOS	Simple Knowledge Organization System
SPOCS	Simple Procedures Online for Cross- Border Services
sTESTA	secure Trans European Services for Telematics between Administrations
STORK	Secure Identity Across Borders Linked
SWRL	Semantic Web Rule Language
TBC	To Be Clarified
UBL	Unified Business Language
UML	Unified Modelling Language
UN/EDIFACT	United Nations/Electronic Data Interchange For Administration, Commerce and Transport
URN:LEX	Uniform Resource Name Namespace for Sources of Law
UTF-8 format	8-bit Unicode Transformation Format
UUID	Universally Unique IDentifier

Acronym	*Explanation*
VPN	Virtual Private Network
WP	Working Package
Xjustiz	A data model is used in the judiciary data exchange throughout Germany
WSDL	Web Services Description Language
XBRL	eXtensible Business Reporting Language
X-road	Provides domain name registration and transfers, website redirection, e-mail catch-all and redirection
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XÖV	XML öffentliche Verwaltung
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

Table 1: Abbreviations

Executive Summary

This third deliverable of e-CODEX WP6 'Document standards and semantics' defines specifications to enable meaningful exchange of information between the systems supporting legal procedures across the Member States by providing semantic interoperability. The planning as part of the technical annex to the Grant Agreement for e-CODEX mentions the start of piloting by November 2012. Because of this demand for operational evidence of the e-CODEX solutions from November 2012 WP6 has to deliver XML Schemas on short notice. However, e-CODEX also aims for long term sustainability demanding for a comprehensive approach. This paradox has been overcome through the agreement on having short and long term strategies as well on the content of the short term strategy.

The short term strategy for WP6 aims at delivering XML Schemas on time to have e-CODEX support a maximum of 5 use cases as of November 2012. However, the amount of legal procedures to be supported and the wide array of concepts in the domain of Law will eventually lead to the conclusion that a three level semantic framework based on widely accepted standards is inevitable. The framework, consisting of a conceptual model, the logical model, and the physical model, should be kept in a repository. In order to ensure the long term sustainability of the semantic assets of e-CODEX decisions on several topics like the standard for data modelling (CCTS or NIEM), selecting an organisation to host the repository or the application of EPOC IV data model for the EAW use case. The full impact of these decisions is not clear yet. Therefore additional research into the impact will start after summer 2012.

The work on semantic interoperability is divided into activities for the *WP6 core team*, the *WP6 user council* and the *schema creation group*. The initial modelling of CCTS and XML Schemas is done by WP6 core team. This core team or concept creators will identify and model the appropriate concepts. Concept is used by WP6 in its philosophical meaning[1]: "an idea or mental image which corresponds to some distinct entity or class of entities, or to its essential features, or determines the application of a term". The concepts created by the core team will be reviewed by the WP6 user council. The consecutive development of XML Schema for a use case is done by the schema creation group based on the input from core team and user council.

Automated processing of information requires a deeper structure from forms as is available from the available (PDF-) forms. The Working group on e-Law (e-Justice) in their meeting at April 16th supported the idea to use a deeper structure at the transportation level to allow for automatic processing of the legal procedures supported by e-CODEX for the life-time of e-CODEX.

Future work in WP6 consists mainly of preparing and taking decisions on standards and methodology and setting up requirements to select the proper organisation to host the semantic assets from e-CODEX.

[1] Oxford Dictionary

1. Introduction

1.1. Scope and Objective of Deliverable

This document is the deliverable D6.3 "Concept for implementation of WP6" of Work Package (WP6) of the e-CODEX project. The document is stored on the e-CODEX website: http://www.e-codex.eu/index.php/downloads2/category/1-deliverables

This deliverable introduces the methodology for modelling, refines requirements presented in D6.1 "Requirements" and gives functional specifications for the individual building blocks identified in D6.2 "Reusable assets and missing building blocks". For this, D6.3 gives an overview on what standards might be selected and short term choices are presented. In addition, the modelling steps are presented.

1.2. Relations to Internal e-CODEX Environment

It is clear that there are dependencies between the different WPs in e-CODEX context. WP6 is strongly linked to WP5 that provides the overall functionality for transporting messages in the domain of e-Justice Services from a sender to a receiver. Another link is to WP7 that provides the IT-groundwork and architecture for interoperability between the systems to be connected, including the security and legal aspects. Beyond that, WP4 will establish the identification and electronic signature requirements. WP3 is defining the underlying business processes of the use cases to be implemented within e-CODEX. These business processes give input to the creation of XML Schemas to be created by WP6.

1.3. Relations to External e-CODEX Environment

WP6 has a relation to other LSPs and National Solutions. The products, services and documents produced by the LSPs and National Solutions were analyzed in preparation of possible adaptation during the implementation of e-CODEX's use cases.

The Working group on e-Law (e-Justice) on April 16th 2012 has agreed with the use of harmonization at a deeper structure to support automated processing of cross border legal procedures during the time the e-CODEX LSP is running. The working party asks for and awaits our technical guidelines on how to deal with the issue now and in the future. This will be part of the future work of WP6.

The ISA-program has launched Core Vocabularies on Person, Location and Business. WP6 has participated in the discussion towards the creation of the Person vocabulary in particular. These Core Vocabularies will be integrated within the semantic assets of e-CODEX. Next to adopting these Core Vocabularies WP6 also started the discussion on Roles and Mandates as candidate Core Vocabularies. The outcomes of the discussion will be integrated in the long term strategy for semantic interoperability for e-CODEX.

EuroJUST and WP6 have been in close contact regarding the data model for the European Arrest Warrant (EAW). EPOC IV has delivered a successfully tested and applied data model for EAW. WP6 currently analyses how the data model can be applied for e-CODEX. The cooperation between EuroJUST and e-CODEX will extent from WP6 only to e-CODEX wide cooperation starting with WP3.

1.4. Structure of the document

The document is structured as follows:

Chapter	Description
1. Introduction	Present the document and describe the work done
2. Methodology of work	Description on how the work presented in this document has been developed
3. Two types of communication	The use cases demand from e-CODEX to support several types of communication
4. Use case centric approach to modelling	Possible strategies resulting in scenarios for the methodology to modelling
5. Requirements to Building Blocks	Requirements to Building Blocks in order to come in practice
6. Towards XML Schemas	Steps to be taken to create XML Schemas
7. Conclusions	The main conclusions derived from the work presented in present document

Table 2: Document structure

1.5. Guide to the reader

This deliverable D6.3 'Concept for Implementation' is highly technical. Reading and understanding its content demands either knowledge of data modelling or familiarity with the preceding deliverables D6.2 or D6.1. All readers that are engaged with implementation of use cases are called upon to also take notice of other e-CODEX materials as background.

For those that want to understand the process of how semantic interoperability will be achieved however, unfamiliar with data modelling, you are recommended to read chapters 6 and 7. Paragraph 4.3 offers 'newcomers' to semantics a short introduction on the distinction between the elements of the 3-layer semantic framework: the conceptual layer, the logical layer and, the technical layer.

2. Methodology of Work

This deliverable D6.3 “Concept for implementation” builds on the deliverables D6.1 “Requirements” and D6.2 “Reusable assets and missing building blocks”. The requirements introduced in D6.1 are further refined and additional requirements have been introduced to support the implementation.

The structure of deliverable 6.3 resembles the way the analysis for the deliverable has been performed. First we looked into the use cases that will be piloted for e-CODEX. As the use cases stem from Civil Law and Criminal Law, the types of communication that need to be supported are relevant and therefore they were investigated for specific needs for implementation. The next step in the analysis was to bring the needs of the use cases together with the 3-level framework for semantic interoperability. D6.3 describes the practical implications of the 3-level framework and its meaning for the implementation of the use cases. The reader will notice a difference in the short term application and the long term strategy towards sustainability of the semantic interoperability. The final part of D6.3 is an instruction for the modelling of the XML messages.

The use case centric approach demands that individual attention is paid to every use case and the legal basis of the use case. To align the modelling method with the context presented in the individual use cases the tracks “public to government” and “government to government” are identified. The tracks give rise to nuances and technical choices that must be reflected in the method and standardization.

Some use cases do have associated forms others don’t. For this reason the preconditions differ from one use case to the other. These differences influence also the preconditions for the modelling approach. Next to these preconditions the legal implications of semantic interoperability have surfaced. Automated processing of information requires a deeper structure from forms as is available from the current ones. The Working group on e-Law (e-Justice) in their meeting at April 16th supported the idea to use a deeper structure at the transportation level to allow for automatic processing of the legal procedures supported by e-CODEX for the life-time of e-CODEX.

3. Two types of communication

The Lisbon Treaty set as the objective for citizen and company-oriented European services to be available and accessible. This objective is applied to IT-solution through three principles in the European Interoperability Framework (EIF):

1. User centricity
2. Inclusion and accessibility
3. Multi channelling.

Although e-CODEX does not eliminate the channel of paper and post delivery for citizens and companies, paper handling of cross border legal procedures is a topic not covered by e-CODEX.

The European e-Justice portal is a front end for citizens and companies, for instance to file a claim within the domain of civil law. The e-CODEX infrastructure connects with the e-Justice portal to enable citizens and companies to start cross border legal procedures. However, according to the EIF[2] principles the e-Justice portal cannot be the sole channel to file claims. The implications of the EIF principles might need some closer attention to exemplify the consequences for e-CODEX. When filling a claim, citizens and companies are allowed to seek assistance from legal professionals (please note; citizen/company should be interpreted in this case as a natural or legal person from outside the government). The earlier mentioned EIF principles hold also for legal professionals. Therefore legal professionals must be enabled to use the channel of their preference, for example through specific software. Of course e-CODEX is not obliged to provide specific software for all kinds of legal professionals. But, e-CODEX is obliged to provide specifications to enable legal professionals and partners to interface with the e-Codex infrastructure and services.

Next to the issue of multi channelling, digital signing of a claim needs attention. The European e-Signature Directive 93/1999 promotes cross-border legal recognition of electronic signatures and ensures free circulation within the internal market of e-Signature supporting products and services. The ability to select and support different signing services and, the chance for business to provide such services is stipulated in directive 93/1999 art 3. Token providers and means to digitally sign are not restricted by government. Citizens must be enabled to digitally sign the document under the conditions the citizens prefer, with respect to the requirements from Directive 93/1999. The government of each Member State supervises the quality control[3] of the signature.

Although the e-Justice portal might be considered the primal front end for citizens and companies – at least during the lifetime of e-CODEX- for electronic communication also other channels should be supported. Regarding WP6 the support of these other channels consists of the availability of the XSDs to parties outside e-CODEX. The XSDs include provisions to support both other channels of communication and the use of the Trust OK token, as will be developed by e-CODEX's WP4, to enable cross-border recognition of digital signing services.

[2] European Interoperability Framework (EIF) for European public services, Brussels, 16.12.2010 COM(2010) 744 final, Annex 2. http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf accessed on July 18th 2012

[3] See D10.2 requirement ALL-RQ-F-001/WP4-RQ-F-003

3.1. Communication type 1: Citizen to government

Citizen to government communication with a cross border context concentrates for e-CODEX around civil law. The use cases from civil law are the procedures EPO and Small Claims. The use cases need to be supported for at least as long as the lifetime of e-CODEX. Both regulations provide forms as part of the legislative act. The e-Justice portal currently provides the forms on EPO and Small Claims to citizens, legal professionals and companies, and also for the courts. As such the choice is given to use an electronic PDF document or a web form. e-CODEX provides the means to enhance the back office handling of EPO and Small Claims in MSs by creating the XML format representing the data structure for the PDF or web form. For the claimant the infrastructure is of no concern. The services regarding the form, filling in, guidance and signing will be present in the e-Justice portal.

3.2. Communication type 2: Government to government

e-CODEX goal is not only to enhance and improve the accessibility of legal procedures for citizens and companies. The improvement of judicial cooperation between legal institutions in the Member States is also of major importance. Professional organizations are enabled to deliver highly structured messages. For bilateral and multilateral communication a higher semantic precision enables more efficiency. Data that leaves no room for multiple interpretations will enable Member States to be interoperable in a semantic sound fashion and to be efficient in proceeding cross-border legal procedures.

However, e-CODEX has to take into account that governments maintain proprietary coding schemas that have to be transformed in order to be interpreted. This transformation is better known as mapping. Mapping is the common method to come to a solution to accept and adapt to the legacy of IT-systems in the Member States, the national legal systems and national IT standards. e-CODEX has to and will provide the means to connect rightfully and meaningfully that data that is presented in a different format and may carry different interpretations within in the Member States. So the data is clearly and uniformly understood when exchanged through the e-CODEX infrastructure. Therefore multilateral and bilateral communication mappings for Member States have to be developed. Code lists are the solid and auditable method to communicate information with high precision. The existence of various, well maintained, code lists on the European level offers e-CODEX the opportunity to reuse these

4. Use case centric approach to modelling

Use case centric modelling is concentrated around one use case. For e-CODEX a use case is an EU regulation, directive or council decision. Every use case involves implementing the legal procedure, the electronic services and the electronic forms and documents to be exchanged between the partners involved.

The use case dictates the set of constructs and their meaning as part of a semantic model. For e-CODEX "Work Package 6 Document standards and semantics" the XML schema is the primary objective. The three-level framework consisting of the conceptual, logical and physical model has one major purpose and that is to reach a high quality message specification or document structure expressed with XML Schemas.

4.1. Scenarios

We distinguish three scenarios for the methodology.

Scenario I

The use cases EPO, Small Claims, EAW and Mutual Recognition of Financial Penalties all have predefined forms. The first scenario is an act of reengineering and formalisation. The generic approach for this scenario is the application of existing data definitions from the predefined forms. The XML Schemas for each use case will be deduced from the original form, part of the legislation. As a consequence, no standardisation will take place until the start of the piloting phase. This non standardization is only applicable to the use case specific part of the messages.

Scenario II

In the second scenario, predefined forms are not available or data definitions from predefined forms are not used (or only partly used). This scenario applies to messages based on interpretation; as no data definitions are present or seemingly alike concepts are defined differently. Legislation and the legal procedure should determine the information required at what moment in the procedure. The business process is leading for the identification of information to be exchanged. As a consequence, standardisation can take place from the beginning, while the piloting starts.

Scenario III

The third scenario is based on the presumption that the data definitions of the available forms lack consistent structure, meaning or scope. As such it would imply reconsideration and improvement of the data definitions for the use cases with forms available and so existing data definitions might need to be adapted.

The second and third scenario involve more modelling. Because of this the deduction of what data elements must be part of the message might take more iterations.

The management of WP6 consulted the Working group on e-Law (e-Justice) on April 16th 2012. The working group supported the idea to use a deeper structure at the transportation level to allow for automatic processing of the legal procedures supported by e-CODEX for at least as long as the life-time of e-CODEX. The working group asked for and awaits our technical guidelines on how to deal with the issue now and in the future. e-CODEX WP6 will use scenario III to develop XML Schemas. Scenario II will be used in case no predefined forms are available.

4.2. Information

For e-CODEX WP6 'Document standards and semantics' the focus is on information exchange. For WP6 the interpretation and the structure of the data that is to be sent and received are the main areas of interest. The media to carry the data are PDF documents and plain XML messages.

To interpret a message the information must meet a set of quality requirements. In order to achieve a high quality, the modelling of the message or document starts by introducing the concepts that must be expressed in the message. A message or document only carries a subset of the concepts identified as part of a legal act. For this it must be clear which concepts and properties are selected to be part of a message. For e-CODEX the concepts and the properties to be used are predetermined by official forms whenever available. If forms are not integral part of the legislation involved it is up to e-CODEX to provide the necessary concepts and their properties methodologically. The description and justification of the methodology is the subject of this deliverable.

4.3. Modelling

The essence of automated information exchange is to limit as much as possible differences in interpretation and understanding by sender and recipient. Therefore, it is required that before information is communicated it must be structured and expressed in abstract but meaningful terms.

As described in the previous deliverables, e-CODEX uses a 3-level framework towards semantic interoperability. Every level of abstraction serves a different objective. Semantic interoperability most of all serves the alignment of business and IT. The alignment to the business processes starts from the conceptual model. The conceptual model (for definition see page 20) is the model for communication and harmonization. Currently, the main purpose of e-CODEX is providing input for the logical model (for definition see page 20) to eventually support the XML Schema generation. However, the prospect of an increasing number of legal procedures to be supported by e-CODEX makes the need for a conceptual model straightforward. The conceptual model will guide and support business and IT to create the foundation for information exchange, through reuse of experience and application of already known and used concepts. For composition and reusability of concepts, the logical model facilitates the design process by selecting, composing, fine tuning and coding logical elements to use in messages supporting the execution of legal procedures. In contrast to the logical model the conceptual model may be cross domain. The logical model is domain oriented and therefore more use case centric. The physical (or technical) layer (for definition see page 19) is exclusively related to the use case or project.

The conceptual model

The conceptual model is a semantic description of a use case or represents a cross domain view. The description of the concepts must provide enough information to be able to communicate their meaning, individually or in conjunction. The conceptual model gives semantic expressive power. The resources, part of the conceptual model, receive a clear definition in natural language and gain additional semantic nuances relating them to each other using semantic constructs. In solitude they have little meaning but in conjunction they provide additional context.

Definition: A concept is a representation of a real object or abstract term.

For e-CODEX

Definition: The conceptual model is an abstract representation of the legal domain or legal act incorporating the concepts that are essential for communication.

A conceptual model will generally contain a lot of contextual information. Information that is common to both sender and recipient and as such does not deliver any added value in support of the process involved. A description of the entity type 'Car' in a conceptual model will mention that a car usually has four wheels, a steering wheel, seats, an engine and gears. A car manufacturer is not going to include all those facts in a message to a retailer if the topic of the message is the delivery of a certain amount of cars. The logical model provides the opportunity to delimit the amount of information to be exchanged.

The logical model

The logical model is domain oriented and more use case centric than the conceptual model. The logical model is the place to compose the message specification and the individual components that will be part of the message. For this a selection of related concepts becomes a component or the individual concept may become an item as part of a code list.

A logical model is different from a conceptual model in that it contains only those components that are used to create the specifications for exchange of structured data. The logical model for any given domain only contains components that are relevant to that domain. The logical model can be delimited to only those components that will actually be exchanged.

Definition: The logical model is the codification of the conceptual model where concepts become part of a composition of components that are the building blocks for electronic messages.

The logical message specification resides in the logical model or is a submodel.

Definition: The logical message specification is an ordered selection of components representing related concepts.

The physical model

The technical implementation uses the logical model to derive the technical equivalent. The technical model is supportive to the business processes and not related to any presentation and consists of XML Schemas.

Definition: The physical model is the (set of) XML Schema (s) representing the message specification in technical terms.

In the long term, e-CODEX knowledge modelling is supposed to develop a solution with a high degree of expressivity in order to describe the complexity of the scenario to be addressed and to cope with sustainability requirements. For this reasons a more complex knowledge modelling solution can be foreseen. However, the long term strategy will next to add abstraction by means of knowledge modelling also add readily understandable modelling. The distinction between modelling for a domain and modelling for specific documents is a very concrete, though complex activity. The distinction and handling with its implications are of such complexity that suggesting having a definite solution available by November 2012 would be unrealistic.

The table below provides a basic overview of the 3-level framework towards semantic interoperability.

Level of Abstraction	**Goal**	**Application**	**Method**	**Implication**
Conceptual	Alignment business and IT through formalized description	Domain Cross domain	3 useful approaches •Relation •Rule •Logics	State of the art, proven technology not yet known in large scale information exchange
Logical	Ensure that data definitions are derived methodologically to enhance reusability at the technical level	Domain	•NIEM •CCTS •Proprietary	Methods maintained by USA, UN or private companies
Physical or Technical	Ensure mutual understanding between systems of information exchanging partners	Project Use case	XML & XSD PDF	Every day business in organization EU-wide

Table 3: Basic overview of the 3 level framework towards semantic interoperability

4.4. Supportive metadata

Citizens, companies or legal professionals might need additional information when filling the forms or creating the documents. The information could concern the procedure, the status of the form and

the role of the document in the procedure. If the information concerns the right for Legal Aid, the additional metadata should be linked to the relevant form or document. If the assistance concerns the human-computer interaction, it is considered out of scope for e-CODEX. However, e-CODEX will address this issue with relevant partners.

4.5. The connection of the 3-level framework to known standards

The basic figure underneath presents a first glance of how the levels from the theoretical model are connected through existing standards.

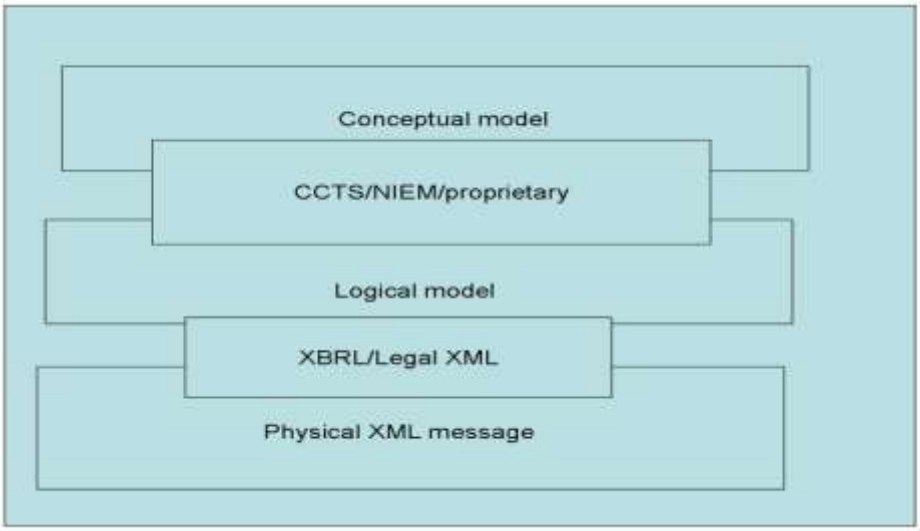

Figure 1: Connection of existing standards to the theoretical model

The logical model may be expressed in the standards CCTS, NIEM or a proprietary solution. The logical model is an enhancement of the conceptual model. Concepts that are part of the conceptual model are therefore presented as components that reflect the original concept, and are further fine tuned at the physical level. For this some overlap between the conceptual and the logical model exists. The logical model is more restrictive.

The logical model quintessentially supports the assembly of messages and lives close to the technical presentation layer and must be in line with the naming and design rules (NIEM, CCTS, etc). The logical representation of the XML message is translated to XML Schema or is directly expressed in XML with a supportive user interface. The standard XBRL[4]for the financial domain is illustrative for hybrid solutions that model with XML to be the model and exchange format in one. For this solution no additional generation of XML is necessary. Legal XML[5] provides a set of messages based on the NIEM standard.

[4] www.xbrl.org/ accessed on July 18th 2012

[5] http://www.legalxml.org/about/index.shtml accessed on July 18th 2012

5. Requirements to Building Blocks

The table Buildings Blocks below finds its origin in D6.2 “List of Standards, Reusable Assets and Missing Building Blocks”. This chapter follows the structure of the table underneath in order to come in practice for WP6.

Data model	BB1 - Conceptual Model
	BB2 - Tool for the composition and maintenance of conceptual models
	BB3 - Link between the conceptual and the logical models
	BB4 - Logical model
	BB5 - Standards for logical modelling
	BB6 - Modelling principles
	BB7 - Tool for logical model composition and maintenance
	BB8 - Tool for composing data exchange messages according to NDR
	BB9 - NDR and for message composition template(s)
Identifiers	BB10 – Method to generate unique id’s
Repository	BB11 – Repository complete with legal body for maintenance
Container	BB12 – Container format
	BB13 – Tool to support container life cycle management
Mapping	BB14 – Mapping methodology

Table 4: Building blocks

5.1. BB-1 Conceptual Model

The conceptual model is identified as a building block for e-CODEX. Its primary functional purpose is to give input for the construction process of a message. The conceptual model is present to express the semantic meaning. The conceptual model is useful for communication purposes and to promote harmonization in the data analysis phase. The current status of e-CODEX allows for the conceptual model to be limited to be written on a piece of paper. e-CODEX supports as of November 1st 2012 the maximum of 5 use cases. It will not be too difficult to keep them oversight. However, the amount of legal procedures to be supported and the wide array of concepts in the domain of Law will eventually lead to the conclusion that a conceptual model, based on widely accepted standard, is inevitable. Just a piece of paper won’t do the job than anymore.

The remainder of this section dives into the necessities to consider when selecting a standard to construct the conceptual model. A decision on the actual selection of standards and tooling for a conceptual model will be a deliverable of WP6 at a later time in e-CODEX.

A conceptual model that aims to support a domain as broad, nuanced and volatile as Law, the method, to develop a conceptual model, must be flexible, maintainable and as such future proof. To be more than just a preliminary model for message design, the conceptual model must have expressive power. For that the semantic richness of the constructs enables people to express the meaning.

Taking into account all that has been written in this paragraph so far and, sound principles towards computerization, the functional requirements in order to be future proof have been derived. As such the functional requirements are:

1. An open and inviting modelling method
2. An open standard
3. The ability to support growth
4. The ability to express semantics
5. The ability to act flexible and be use case driven
6. The ability to act cross domain
7. The ability to be open in combination with precision
8. The ability to refactor and transform
9. The ability to promote parts of a model.
10. The ability to have uniqueness
11. The ability to relate concepts
12. A broad selection of supportive tooling.

The conceptual model must be open in the sense that different projects and communities must be able to share their models. The conceptual model must be expressed in an open standard for vendor independent application. This is a main requirement for interoperability in the European Union and in line with EIF. A model may grow over time and be a composition of a number of theme oriented models. An upper ontology, generic model, theme and use case centric model may live side by side or may form 'layers' that import each other to have the maximum ability to share and reuse. Concepts part of use case centric models that may be promoted to a generic model when used frequently, have to be easy to extract and to rearrange.

To have semantic expressive power in the conceptual model is of great importance. Modellers must be free to choose the level of semantic richness for the model. Having a standard that is rich in nature, gives a flexible mechanism to introduce additional constructs and gives freedom of choice what constructs to incorporate makes that different use cases under different conditions may be supported.

The ability to adopt and to share different models and to inherit concepts from different models gives flexibility. Where concepts are promoted to a higher level, refactoring the model must be made easy.

Semantic expressive power, the constructs supported and the precision given by the model must be balanced. To be truly open, the models must be comprehendible for different persons and groups.

The ability to share and the uniqueness of the concept are close related. The name the concept carries must be unique cross domain. Two concepts with the same local name must not result in a collision when both are required for modelling purpose. Uniquely referencing concepts part of a model and referencing concepts part of different models is a powerful concept in its own right that makes reuse feels natural. For instance, inheritance may be cross domain and homonyms may exist side by side. In cases where no unique references are present name collisions may occur.

For a standard to be adopted the set of supportive tooling must have a considerable volume. For this the openness and the adoption level by industry must be high.

Candidate standards

For the conceptual model different paradigms may be used. The main stream paradigms are hierarchical, relational, object oriented and logical. The hierarchical and relational paradigm is in close relation with database design. The object oriented and logical paradigm is for broader usage.

The standards RDF/OWL and UML are the candidates that are open standards and are widely accepted. Alternatives are less main stream or open. UML specifies different views. For conceptual modelling the object oriented class diagram is presented. The presentation format is easy to interpret for a large group of people. For exchange the XML format XMI is presented. The drawback for UML is the graphical orientation and the vendor dependent interpretation of the exchange format XMI. For NIEM and CCTS the logical model might be expressed in UML where a small set of UML tools generate compliant XMI for further processing and XML Schema generation.

RDF/OWL is a crossover of the object oriented and logical paradigm and expressed in different textual formats where XML is the dominant flavour. The graphical presentation is tool dependent. A large amount of tools gives the ability to edit the model both graphical and textual. This gives a higher flexibility. A presentation of RDF/OWL like a UML class diagram gives the audience the best comfort.

The formulated generic functional requirements are translated into their technical functional equivalent.

Technical requirements	UML	RDF/OWL
Textual representation in XML	No	Yes
Graphical presentation	Yes	Vendor dependent
Expressive power	Class diagram, average	High
Semantic richness of the existing set of constructs	Medium	High
Ability to choose primary and supporting constructs	No	Yes

Ability to introduce new constructs	No distinction is made between different types of properties	Yes
Ability to import different models	Tool dependent	Part of the standard
Unique reference expressed as a URN	Manual introduction	Part of the standard
Export format	XMI, vendor dependent influences present	XML, the textual presentation

Table 5: UML and DFF/OWL related to technical requirements

Presenting two options for editing and presenting the model, a textual representation in XML and a graph presentation. The latter provides more flexibility, flexibility

- to add new constructs,
- to refactor or to rearrange the model and
- to keep track of the changes.

The set of constructs, the ability to enhance and to introduce new constructs and the expressive power they have are representative for the strength of the standard. The ability to import models and to reuse constructs present in different models, reveals its high modularity. The support for URN respectively Name spaces makes that the model may support constructs that resemble URN:LEX and ECLI. The support of URNs makes that the standardization and generation of names is supported and the provision for unique references is present.

RDF/OWL in depth

OWL is a W3C standard and part of the semantic web stack. The standard naturally blends in with the XML standard.

Different from other modelling standards (eg. UML) the standard gives structure for both concept and the instance. Whereas a natural person is a concept, the person Jason Drake is an instance. Both occurrences are constructed with RDF/OWL and may be expressed in XML.

For OWL different levels of sophistication and semantic expressive power may be chosen. Where other standards lack the capabilities to have strong relational meaning, OWL gives freedom to define different types of relations for additional nuances. OWL incorporates a set of standard constructs to express relationships, the properties. For instance: reflexive, inverse, transitive and symmetric. A concept may reference itself (reflexive). A property may have a counterpart pointing in the other direction (inverse). Including a relationship may automatically imply having secondary relationships (transitive). A property may hold in both directions (symmetric).

In addition to the formal properties present in the conceptual model language an upper ontology gives even more nuances if necessary. To give an impression four additional nuances may be given:

1. Containment: An entity is placed in a room. The entity is not part of the room.
2. Membership: A person belongs to a committee or sub committee.
3. Connections and branches: A computer is connected to the World Wide Web.
4. Constituents: The painting and the paint used to create the painting

5.2. BB-2 Tool for the composition and maintenance of conceptual models

A tool for conceptual modelling gives support for presenting, editing, refactoring, merging and (re)arranging. A good tool gives flexibility in representing the model in graphical and textual format. Interactive GUI support for connecting elements gives the modeller direct visual feedback. An advanced editing mode to edit the textual representation offers the modeller the opportunity to introduce repetitive adjustments to the model efficiently. The ability to define transformation rules and the ability to execute the rules automatically enables the modeller to achieve complicated refactoring tasks.

The ability to work on more than one (sub) model and the ability to transfer and merge constructs gives the modeller the means to make a modular design. Where a conceptual model consists of a dozen of sub models, maintained by different team members, team collaboration must be supported by the tool. An open API makes that different tooling and functionality may operate on the same model.

To share and align definitions part of the model, multi-language support where the resources names and definitions are presented in different languages, makes that the model may be communicated in all the EU languages.

To store the models in a persistent way, a repository that is modelling tool independent and is accessible from different locations gives additional flexibility. The modelling tool must be flexible in supporting different import and export formats. In line with the European Interoperability Framework open source and open standard are promoted to be a first choice when alternatives exist. For e-CODEX the open standard RDF/OWL must be supported. SKOS is the international open standard for thesauri. The combination with OWL is wide spread and therefore support for SKOS should be incorporated.

With RDF/OWL the discovery of knowledge and rules are supported by connecting the model to an inference engine. An inference engine allows consistency checks to check the model for contradictive statements.

Open source modelling tools, RDF repositories and inference engines are present.

From all stated above the functional requirements for the modelling tool are:

Requirement WP6_RQ_T1: Open Source
Requirement WP6_RQ_T2: Support for the open standard RDF/OWL or RDF/OWL in combination with SKOS
Requirement WP6_RQ_T3: Multi-language support
Requirement WP6_RQ_T4: An advanced editing mode, web based and with lexical support
Requirement WP6_RQ_T5: Import and export ability
Requirement WP6_RQ_T6: An intuitive graphical presentation, graphs view
Requirement WP6_RQ_T7: The ability to inference and to check consistency
Requirement WP6_RQ_T8: The ability to collaborate c.q. multi user mode
Requirement WP6_RQ_T9: Support for refactoring and transformation, merging
Requirement WP6_RQ_T10: The ability to design modular
Requirement WP6_RQ_T11: The presence of an open API
Requirement WP6_RQ_T12: The ability to connect with a tool independent data repository

5.3. BB-3 Link between the conceptual and the logical models

The 3-level framework to semantic interoperability implies that each of the levels is connected. The link from the conceptual to the logical level is created through a profile. The profile assures the models to be compatible so that information introduced at the conceptual level is preserved and directly applicable in the logical model. Preferably any foreseen or unforeseen addition to the logical model would be recognized at the conceptual level.

The models should be compatible in the sense that the logical model is a further specification of the conceptual model. For this, the logical model has to be enhanced or restricted with additional knowledge, to connect to and apply on a specific domain. The design decisions made in the conceptual model are leading for the logical model. One has to acknowledge and accept that transforming the model, establishing the link between the conceptual and the logical level means an increasing chance of losing nuances available at the conceptual level. In order to support an easy transition from the conceptual modelling towards the logical modelling, the profile makes that the conceptual constructs may be kept intact or may be transformed according to the set of naming and design rules (NDR). ISO 11179[6] is the foundation for both CCTS and NIEM. The implications for NDR are so few that the profile is almost alike for either choice.

The conceptual model incorporates constructs for inheritance and a rich set of constructs to express relationships. The concept is a class that inherits the characteristics from its parent(s). Multiple inheritances are an option. The same counts for properties. This being said, a child is more specific, a parent more generic. Inheritance is not supported for CCTS and NIEM directly. A translation rule must be introduced. There are two choices to simulate inheritance. Either one has its disadvantages.

[6] http://metadata-standards.org/11179/ accessed on July 18th 2012

The first solution is to incorporate the properties of the parent directly. The disadvantage of this choice is that the parent child relationship is not recognizable once incorporation has taken place. The second, preferable, solution is to reference the parent as part of the child. The child inherits the properties of one or more parents. The inheritance remains recognizable because of the introduction of a reference to the parent 'container(s)'. The reference to a container doesn't provide additional semantic nuances. The disadvantage here is that any nuances of an object property will be lost in the logical model. However, the idea that an object property references a class holds if one realizes and accepts that the object property and the reference to a container are equivalent. A `RDF/OWL data property is equivalent with a property in the NIEM or CCTS model. The cardinality of the properties may be adopted, without any change, or the preset cardinality introduced in the conceptual model may be overridden for the message specific components. NIEM and CCTS have naming rules for the individual container, reference and property. These names given for the class, object and data property are input for constructing these names.

5.4. BB-4 Logical model

A logical model presents the features of knowledge for concepts, and the coherence between these features on a specific domain. A logical model is used to provide a framework of modelled components that can be used in a message to exchange information for a specific procedure or process. As such, an XSD is formed within the boundaries set by the logical model. Therefore the logical model should contain all of the components that are needed for data exchange on that domain. The logical model should be easy to understand.

Although a logical model does identify concepts it is different from a conceptual model. The logical model for any given domain only contains components that are relevant to that domain. The logical model can be delimited to only those components that will actually be exchanged. A conceptual model is used to identify a concept with its entire context, according to the approach characterizing the conceptual model.

A conceptual model will generally contain a lot of contextual information. Information that is common to both sender and recipient and as such does not deliver any added value in support of the process involved. A description of the entity type 'Car' in a conceptual model will mention that a car usually has four wheels, a steering wheel, seats, an engine and gears. A car manufacturer is not going to include all those facts in a message to a retailer if the topic of the message is the delivery of a certain amount of cars. A logical model is different from a conceptual model in that it contains just the components that are used to create the specifications for exchange of structured data.

ISO 11179[7] is the widely adopted standard to construct logical models. Government institutions, organizations and businesses in Europe and in the rest of the world that exchange data in a structured way use methods based on the ISO 11179 standard. ISO 11179 recommends a logical model to distinguish between entity types and attributes that are of a certain data type, like 'Date', 'Description' or of an object type. In ISO 11179 terms these attributes are called properties.

Requirement WP6_RQ_F13: ISO 11179 should be adopted as the standard to construct a logical model.

[7] http://metadata-standards.org/11179/ accessed on July 18th 2012

Components in the logical model should be either continuants (entity types that stay basically the same during their lifetime) or occurrences (happenings, events, acts that evolve through time). Dependent entity types like 'Name' are surely necessary. However, aggregates that do not have a well defined meaning, or might have multiple meanings should be avoided. Intersection (also called associative) entity types for modelling N: M relationships should be avoided if they can be remodelled as reification.

A logical model is built for efficiency in the process of creating messages to support information exchange. Code lists exemplify the required efficiency. Code lists standardize the details of items such as country names, languages, academic titles etc. The reuse of code lists on a specific domain or cross domain support interoperability, mutual understanding and reduce the effort for maintenance.

Requirement WP6_RQ_F14: Code lists should be used for at least Country, Nationality, Currency, City, Gender and to denote different types/ kinds of Components, unless these special kinds have additional properties that need to be in the message.

The logical model is not intended as a model to derive a physical model for a database. A logical model is a model for data exchange. Normalization is important, but third Normal Form is generally considered sufficient when the object is creating messages.

The use cases that need support of a logical model are European Procedure for Payment Order, Small Claims, European Arrest Warrant, Secure cross-border exchange of judicial data and Mutual recognition of financial penalties. Because of the demands of automated processing a deeper structure from forms on transportation level will be used for the life-time of e-CODEX (Working group on e-Law (e-Justice) April 16th). Further standardization at the logical level can be achieved for the non use case specific components as message headers, routing information, Trust OK token etc. The use case centric approach offers the opportunity as well to harmonize or standardize as to diversify data definitions depending on the legal prerequisites.

The logical model should contain a header component that can be added to the message as part of the payload. The UN/CEFACT Standard Business Document Header (SBDH) is a good candidate of proven value. It has been used for several applications around the world as e-billing 'Svefaktura' in Sweden; **eXite®** professional service for electronic business transactions, connected to all Austrian and several Czech banks, connected to the Austrian social security system.

Requirement WP6_RQ_F15: The logical model should contain a header component

Components should have a property denoting the period the data transmitted in the component is valid. The logical model should at least specify the cardinality of every property and contain metadata describing definitions of component and their properties, the source of the definition, the creator of the component, date of creation. The set of metadata in Dublin Core should be used.

So judging by the forms the logical model for e-CODEX should contain at least components for Party as a role of a Natural Person or a Legal Person, Natural Person, Legal Person, components Name , Birth, Demise, Address, Account, Claim, Payment, Countries , Places and Currency. All of the components that are important in each of the use cases should be included in the model.

Furthermore, the logical model should state the relationships between the components in the model.

The use cases of e-CODEX need components from the domains of criminal law and civil law. In order to keep the components that describe one domain together, the logical model should be made up of different domains. The logical model should consist of at least one domain for components to do with criminal law and one domain for components about civil law. Apart from that, components for Natural Person and Legal Person are needed, together with components for Name, Address, and Nationality etc. The components that have been or are in the process of being specified in Join-up (formerly SEMIC.eu) should be used. These components can't be assigned to either the domain of criminal law or the domain of civil law. The components to do with Person should reside in their own domain, which should be a Join-up domain.

5.5. BB-5 Standards for logical modelling

The logical model must be created according to a standard. The use of a standard is a necessary (although not sufficient) condition for a consistent model. There are several standards for logical modelling: Entity Relationship modelling, Extended Entity Relationship modelling, UML. These standards are often used when the purpose is to create a database model.

For the purpose of designing messages, CCTS and NIEM are widely used standards, based on the ISO 11179[8] standard. NIEM is used predominantly in the USA, CCTS is used world-wide. For both CCTS and NIEM Naming and Design Rules exist and tooling is available for creating a schema from the logically defined message.

CCTS and NIEM – a comparison

CCTS and NIEM are both widely used standards for data exchange. Both are based on ISO 11179. NIEM is a complete package to support the exchange of information. It includes a data model, so called IEPDs (information exchange package description). These IEPDs are off-the-shelf structures for exchanging messages. NIEM also provides a methodology to extend the existing components. NIEM is a set of predefined components that are used throughout the federal government in the USA. NIEM has a set of IEPD's, which are predefined messages. Use of NIEM seems to imply that this set of messages is to be used, but adaptations and extension are possible. CCTS is a method for structuring information. It provides a methodology for identifying a set of reusable building blocks, called Core Components. The Core Component Library is a set of commonly used Core Components. CCTS is not so much a set of predefined components, rather a method to structure the data in a technology neutral way.

Although both CCTS and NIEM are based on the same ISO 11179, there are differences. One of the differences is the extensibility at the component level. In CCTS it is not permitted to add attributes or associations to an existing Core Component. This is considered a feature, to prevent a component from being adopted in different ways for different needs by different modellers. The prevention of divergence is the key to the idea of a single core component. Initially CCTS was to allow both restriction and extension. The team that drew up the specification for CCTS discussed the matter of

[8] http://metadata-standards.org/11179/ accessed on July 18th 2012

extension together with restriction and concluded it to be undesirable. One could remove all properties of an Aggregate Core Component (ACC) and replace them by entirely different ones, loosing all existing relationships between the Business Information Entities (BIEs) that were based on the same CC and effectively loosing CC-BIE relationship[9] . The relationship of a Core Components with its BIEs is explained below, after the description of NIEM.

NIEM on the other hand, does allow extensions to its components. Another striking difference is the management of inheritance. CCTS does not support inheritance whereas NIEM states all acts and events inherit from a top level entity *Events* the attributes *Start* and *Begin*. Below is shown the high-level data model of NIEM. The part with People, Places, Events and Things is called the core; the other parts are 7 Domains where NIEM has been implemented.

Figure 2: High level data model of NIEM[10]

[9] http://lists.ebxml.org/archives/ebxml-dev/200407/msg00188.html accessed on July 18th 2012

[10] NIEM User guide.pdf

An example clarifies how inheritance works in NIEM. An example of the modelling of inheritance in NIEM is shown through **TheftLocation.**

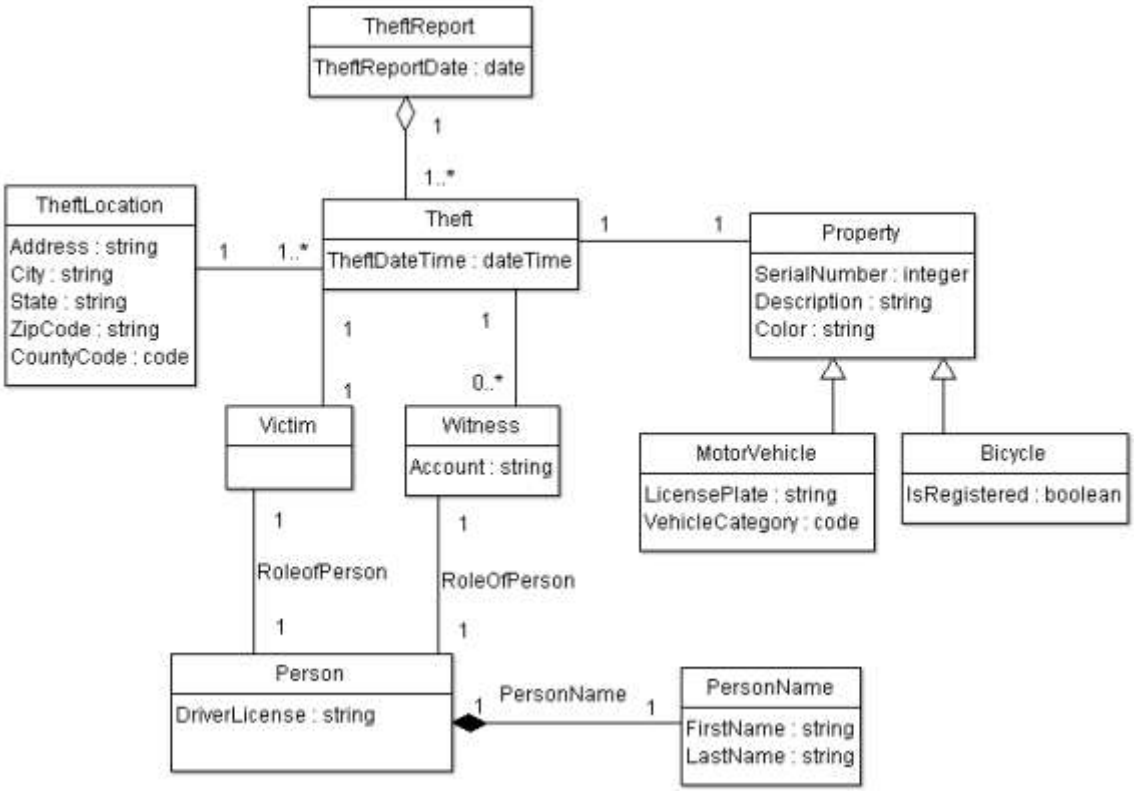

Figure 3: Example of the modelling of inheritance in NIEM through TheftLocation[11].

Check Table 6 for inheritance path of 'TheftLocation'

One should include enough steps towards the basic component to uniquely identify it. Sometimes, multiple paths can lead to an element in NIEM, and the entire path is needed for precision. In the example, the CountyCode property, which is a state-specific county code, is not found in NIEM, so it will require an extension. Therefore, the **Ext?** Column is set to Y, and the Xpath, the trace towards the original component is left blank for now.

Source type	**Source property**	**...**	**Ext?**	**XPath**
TheftLocation		...		nc:Location
TheftLocation	Address	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationStreet /nc:StreetFullText

[11] https://www.ibm.com/developerworks/xml/library/x-NIEM2/ accessed on July 18th 2012

TheftLocation	City	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationCityName
TheftLocation	State	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationState USPostalServiceCode
TheftLocation	Zip	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationPostalCode
TheftLocation	CountyCode	...	Y	

Table 6: TheftLocation[12]

NIEM and CCTS have some constructs in common. Both use Property Terms and Representation Terms, like the use of Types (e.g. PersonType) and the use of Properties (Date, Time) and Code Lists, which are generic representations of enumerated code values. NIEM employs several constructs that are not known in CCTS[13]:

Container Elements

Elements whose presence in types represents semantically weak relationships.

An example is j:DriverLicenseDrivingIncidentAssociationType, which represents an association between a driver's license and a driving incident and contains an element, nc:Person of nc:PersonType. The presence of the nc:Person element does not establish what kind of relationship exists between j:DriverLicenseDrivingIncidentAssociationType and nc:PersonType, only that there is a relationship. This is an example of a semantically weak relationship. In such a case, the element nc: Person is called a "container element" because it only serves the purpose of containing an object of nc: PersonType, while leaving the exact meaning unstated.

In CCTS the DriverLicenseDrivingIncidentAssociation would not be considered a core component, which is supposed to refer to things present in the real world.

Roles

Representations of the different roles e.g., VictimType, WitnessType. In many cases, there is a need to capture additional information about the role. In such cases, a new type is created to represent the role and its properties. For example in NIEM, a person whose whereabouts are unknown is modelled as j:MissingPersonType, which represents a particular role of nc:PersonType. Additional information about the person specific to his/her role as a missing person is modelled as the properties of j:MissingPersonType. Such information may include the date and the location at which the person was last seen, represented as the properties j:MissingPersonLastSeenDate and j:MissingPersonLastSeenLocation.

[12] https://www.ibm.com/developerworks/xml/library/x-NIEM2/ accessed on July 18th 2012

[13] NIEM NDR 1_3.pdf

Figure 4: Roles [14]

In CCTS roles are used, but in a different way. If no additional attributes are required, the role is made an item in a code list. If additional attributes are needed, a new component is created.

Augmentation

Augmentation is a mechanism prescribed by NIEM to create a new NIEM-derived type, via the extension of an existing NIEM type, by adding a block of elements bundled together in another type called an "augmentation type." Augmentation is not supported in CCTS.

To illustrate the mechanism of augmentation, consider the following example. Suppose we wish to use an element of nc:PersonType in our schema, but that we also need to capture additional properties, such as the person's driver's license and place of birth, which nc:PersonType does not provide. NIEM provides several person-related properties in j:PersonAugmentationType, including nc:DriverLicense and j:PersonBirthPlaceCode, which match the two additional properties we want. Therefore, we create a new NIEM-derived type, ext:PersonType, which extends nc:PersonType by adding an element of j:PersonAugmentationType to nc:PersonType. Representation of a reusable bundle of properties (e.g., PersonAugmentationType containing properties DriverLicense, PersonFootPrint," etc.) for the purpose of augmenting the definition of an existing type (e.g., "PersonType") in a way that does not create multiple, and possibly conflicting, specializations of the type in question ("PersonType").

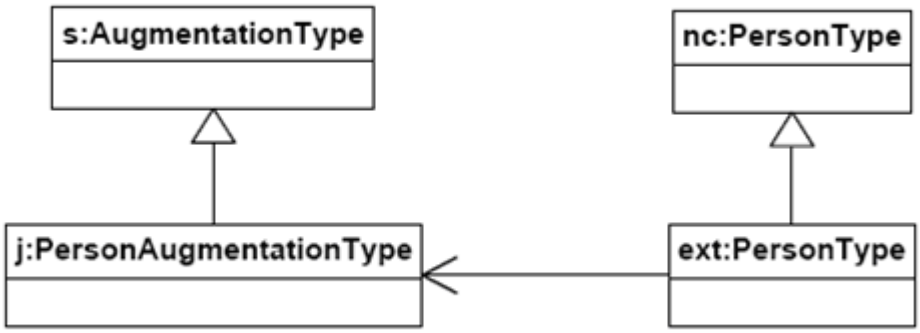

Figure 5: PersonAugmentation Type[15]

[14] http://reference.niem.gov/niem/guidance/user-guide/vol1/user-guide-vol1.pdf accessed on July 18th 2012

[15] NIEM User Guide

Associations

An association is a kind of relationship between two or more objects. The objects and the relationship must have the following characteristics for the relationship to be an association. The objects must be "peers" of one another. This means that no object is hierarchically or logically a part of another. In other words, each object can exist independently of others, and none of the objects lose meaning if separated from one another. The relationship between the objects may exist only if all the participating objects exist and it has its own set of properties separate from the properties of the participating objects.

For example in NIEM, a single or a set of related actions, events, or process steps is represented by an "ActivityType" and a person is represented by a "PersonType." Further, the relationship between an activity and a person, signifying the involvement of the person in the activity, is represented by "ActivityPersonAssociationType."

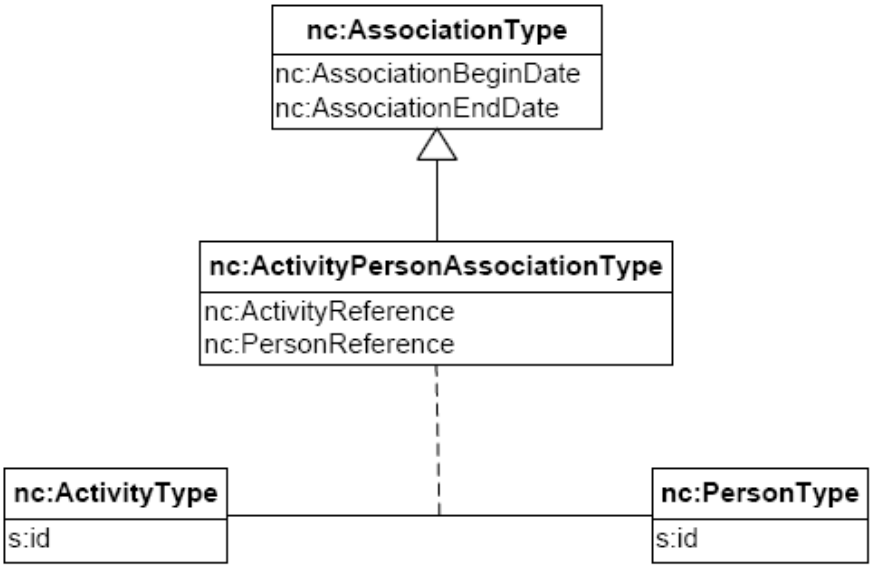

Figure 6: AssociationType[16]

Associations are widely used in CCTS, but they are used in a different way. CCTS uses containment to express that one class is associated with another class.

In comparison to NIEM, CCTS has a small number of modelling concepts. This is not necessarily a disadvantage, as it provides for conceptual clarity. CCTS builds on Core Components that refer to (at least in principle) real things in the real world, things that can be counted or given an identifier. Things like these are put together in classes, the object classes. For a visual presentation refers to the illustration underneath.

Core Components have two kinds of properties. There are data type properties like 'moment in time when something happened' with a representation of data type Time or 'description' with a representation of data type Text. The other kind of property is an object type property, or 'association' as it is called in CCTS.

[16] NIEM User Guide

The idea of Core Components is, that they denote rather broad classes that can be 'tailored', for use in specific contexts. The tailored versions are called BIEs (short for Business Information Entity). The total set of properties of the Core Component can be restricted in the BIE, and the properties can be given so called Qualifiers, which make the meaning of the properties more specific. In the document CCTS_V2 final the Core Components and Business Information Entities are depicted as shown below:

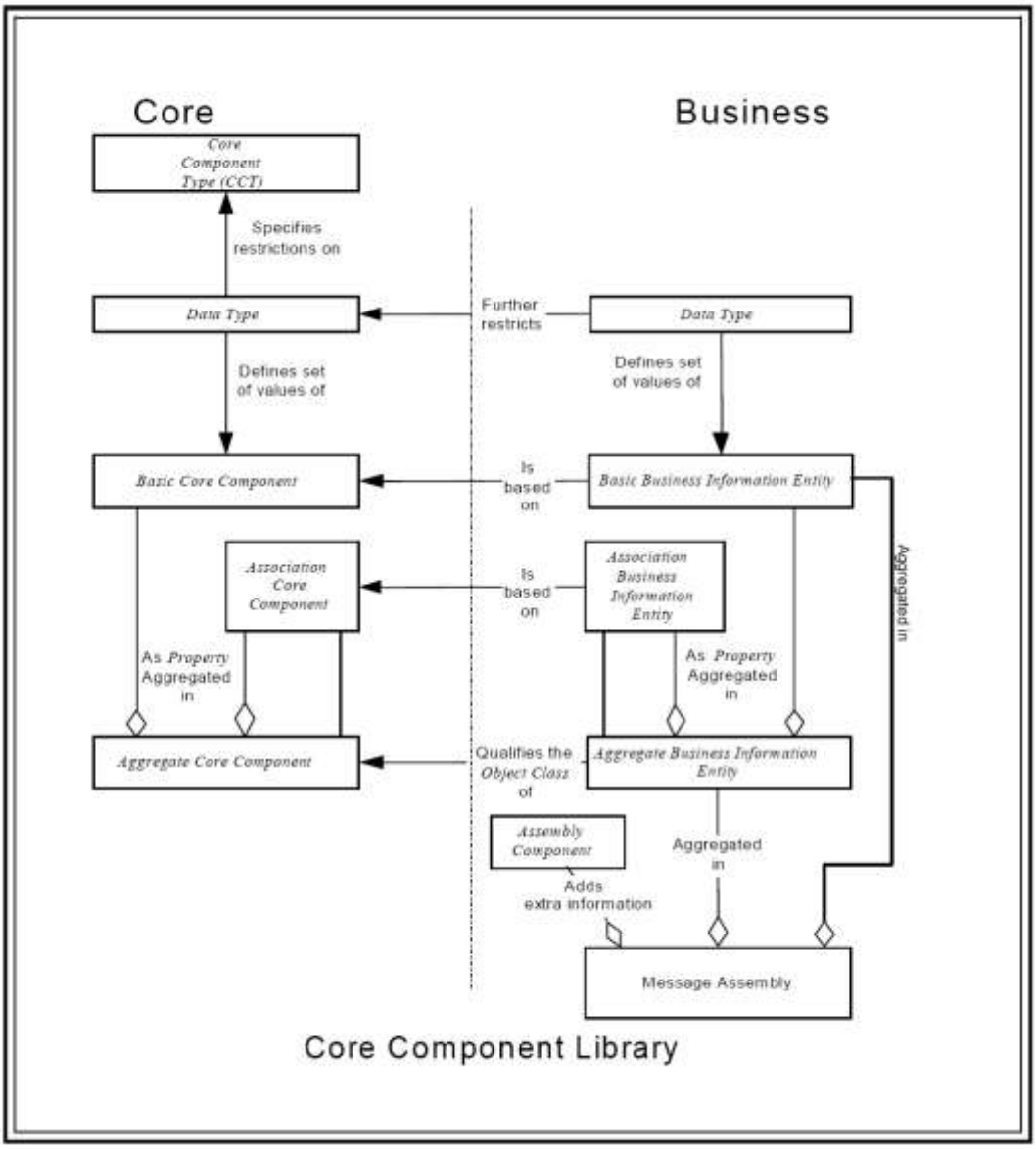

Figure 7: Core Component Library[17]

The way an event is modelled according to CCTS can be illustrated by the example of the modelling of a 'Theft' (we are back to the example used previously). First of all, a theft takes place at a time and a place, so the Core Component Theft is assigned a property 'Time of Theft' with a representation term 'Time' and a property Place with a representation term Text.

[17] http://www.unece.org/fileadmin/DAM/cefact/ebxml/CCTS_V2-01_Final.pdf page 18 accessed on July 18th 2012

Next, it may be necessary to tell who did it and who the victim was. The perpetrator and the victim are natural persons. The way they are linked to Theft is by association. Below is shown how a CC can be restricted in a BIE.

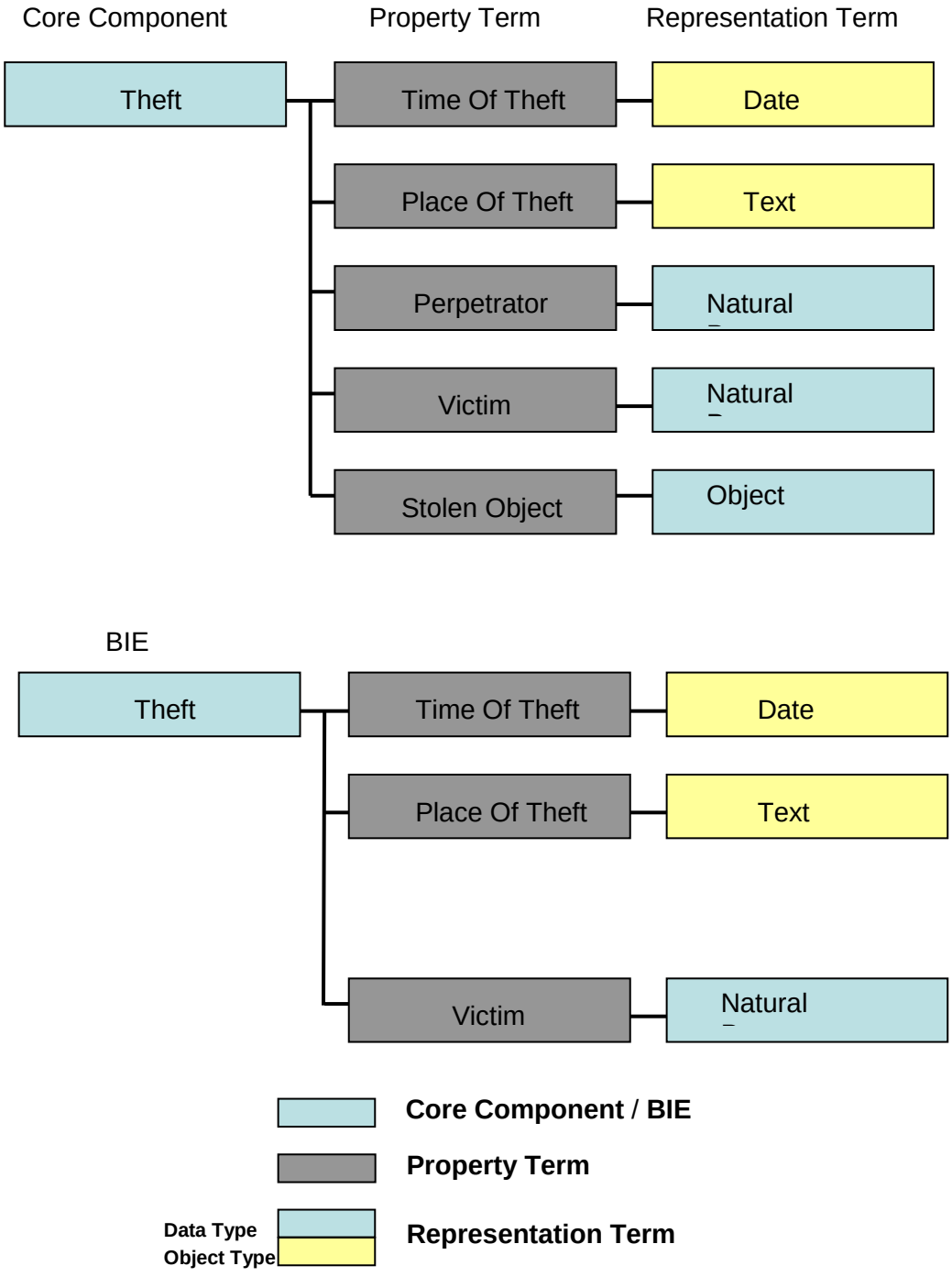

Figure 8: How a Core Component can be restricted in a Business Information Entity[18]

[18] Dutch Judicial Information Agency unpublished documents)

This way of modelling uses a 3 part scheme: Object class (Theft), Property term (perpetrator, victim) and a Representation Term, in the way expounded in ISO 11179. No other constructions are used.

Choosing between CCTS and NIEM as the standard in e-CODEX

The survey of National Solutions and existing LSPs that was conducted at the start of WP6 and presented in D6.2[19],produced two candidates as standard to construct the logical model for e-CODEX: CCTS and NIEM. There clearly are differences between CCTS and NIEM, as shown above. CCTS has a limited range of modelling concepts, providing for conceptual clarity. As shown above, CCTS is more in line with ISO-11179, the standard that is the basis for both CCTS and NIEM. In NIEM Container Elements are allowed**: e**lements that represent semantically weak relationships. This is in violation of ISO-11179 part 4 entitled "Formulation of Data Definitions" that provides the following recommendations[20] for defining data components:

- state the essential meaning of the concept
- be precise and unambiguous
- be concise
- be able to stand alone

However, in deciding which standard is to be preferred, some other aspects can be taken into account as well:

- Recommendations from ISA;
- The availability of tooling
- Possibility of participating in the development of the standard
- The standards in use in EU Member States and affiliated countries not participating in e-CODEX
- Previous commitments to a standard in other European projects
- Advice about NIEM by a US standards institute that is involved in the NIEM standard.

These aspects will be addressed below.

The availability of tooling

The NIEM provides various tools to facilitate the specification of business documents, called Information Exchange Package Definitions in NIEM. These tools include the Schema Subset Generation Tool (SSGT), the NIEM Wayfarer, and conformance validation tools.

So, tooling is available.

The NIEM standard is being adopted in the domain of health care in the US, but the ONC (The Office of the National Coordinator for Health Information Technology) anticipates the need for additional tools for the healthcare domain21. It is not made clear why additional tooling is needed for a new domain, though. These NIEM tools are all of them on-line tools.

For CCTS several tools are available none of which can be used in an on-line version. The German tool consists of a UML editor (MagicDraw) and custom made application for taking the output of

[19] http://www.e-codex.eu/index.php/downloads2/category/1-deliverables

[20] National Information Exchange Model Naming and Design Rules NIEM Technical Architecture Committee (NTAC) October 31, 2008, Version 1.3, p 31

[21] http://efasoft.blogspot.com/2010/07/adopting-niem-for-health-information.html accessed on July 18th 2012

MagicDraw and creating schemas. Magicdraw is a proprietary tool, the "custom made application" (Xgenerator) is free.

The Turkish tool eDOCreator is an on-line tool and provides the UBL components for specifying Business Documents like Order, Quote, Quotes Request etc. The tool allows users to import their own components by uploading a description in a spreadsheet. The use of the eDOCreator is free of charge.

The Dutch tool (MDW, MetaData Workbench) 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. There is (as yet) no version of the tool on the web. The Dutch tool can be acquired free of charge.

Requirement WP6_RQ_NF1: The tooling for maintaining the logical model and creating message specifications should be open source and free of charge.

The short term strategy for WP6 aims at delivering XML Schemas to have e-CODEX support a maximum of 5 use cases as of November 2012. To that purpose the XML Schemas will be created from CCTS through the use of the Metadata Workbench.

Possibility of participating in the development of the standard

CCTS is a standard that is intended for the whole world to use. The NIEM standard is designed for the exchange of information within the US federal government. Currently the Canadian government is experimenting with NIEM. Although NIEM has committees to extend and deepen the model, the participation is at the moment not open to all, like there is for CCTS.

On the subject of participation in the development of a standard EIF, the European Information Framework, has something to say: "The standard is adopted and will be **maintained** by a nonprofit organization, and its ongoing development occurs on the basis of an **open decision-making procedure** available to all interested parties (consensus or majority decision etc.)"[22].

Requirement WP6_RQ_NF2: The standard that is to be adopted should be maintained in a way that is compliant with EIF.

Standards in use in EU-countries not participating in e-CODEX and previous commitments to a standard in other European projects

CCTS seems to have taken root in Europe. The document on the web UBL in Europe[23] tells the story of UBL/CCTS in Europe:

"Denmark adopted the UBL invoice at a very early stage and this was part of their OIO-XML standard for the Danish public sector. Requirements to use this, and later developments including orders, were made law. The further developments were called OIO-UBL. The UK Government took steps to standardize UBL for most procurement documents but did not go as far as Denmark in making it a

[22] European Interoperability Framework (EIF) for European public services, Brussels, 16.12.2010 COM(2010) 744 final, Annex 2. http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf accessed on July 18th 2012

[23] http://www.documentengineeringservices.com/events/london/UBLinEurope.pdf accessed on July 18th 2012

requirement. Resources were focused on seeking wider public sector adoption in Northern Europe with a joint effort to agree a subset called NES.

European Government studies were part of the development of UBL v. 2. The European joined up government initiative (IDABC) favoured adoption of UBL for e-Procurement. Following the standardization of UBL 2 the momentum continues within Europe's CEN standards body as a CEN/ISSS project called BII - Business Information Interoperability to provide a basic framework for technical interoperability in pan-European electronic transactions. This is compatible with UN/CEFACT in order to ensure global interoperability .The requirements and final specifications will be input into UN/CEFACT. The starting point NES and CODICE customizations of OASIS Universal Business Language 2.0. The BII project involves France, Germany, Italy, Spain, the UK, Denmark, Sweden and Iceland[24].

The data model chosen by PEPPOL as the basis for its specification is the aforementioned CEN/BII, which is CCTS.

A further example of the application of CCTS in Europe is ADMS (Asset Description Metadata Schema), a recent development. The European Commission, under the ISA Programme, is undertaking a number of initiatives to raise awareness of the importance of metadata management for e-Government system development in Europe. These initiatives are part of the ISA semantic methodologies action. In this context, the ISA Programme has created the Community of European Semantic Asset Repositories (CESAR) and has the overall objective to share experiences with setting up and maintaining semantic asset repositories.

Recommendation WP6_RQ_NF3: The modelling standard that is to be adopted in e-CODEX should not deviate from choices made previously in European programmes.

5.6. BB-6 Modelling principles

The semantics of ISO 11179 are not made explicit in the documentation. The implication is that ISO 11179 does not provide ample guidance for consistent and sustainable modelling of a domain as extensive as the justice domain. Additional rules for modelling will be needed to provide enough guidance to do the modelling in a consistent way.

Requirement WP6_RQ_F16: modelling is to be done according to a set of rules, in addition to the ISO 11179 to ensure consistent and sustainable modelling.

When a new component needs to be included in a message, the addition must be done according to a set of rules. Knowledge of the exact nature of the concept that is to be added, greatly enhances the chances of correctly modelling the concept as a logical component.

A set of possible modelling rules is given below. This set is the result of desktop research in combination with personal experience of the writer of this Building Block.

[24] http://www.en.ds.dk/bii accessed on July 18th 2012

Modelling concepts

Messages are often sent in order to convey information about actions. The filing of a claim is an action, as is the claim itself. A lot of the components that are relevant in the justice domain denote acts: Decision, Conviction, Suspension, Appeal, Revision, Arrest to name a few. Actions can be direct objects of other actions. Modelling of actions is to be done where possible.

Actions and thematic roles

If some component is of an 'action' –nature, it has at least attributes for time and a location. Apart from that, modelling the who, what, why, what with is useful for the modelling of actions. This way of modelling is known as modelling with thematic roles. For example, a claim has a claimant and something that is being claimed; the claim may be filed by a claimant or by a representative for the benefit of a claimant; there is a reason why the thing is being claimed.

The roles that should be used in modelling actions are at least thematic roles denoting Place, Time, Agent, Direct Object, Indirect Object (Beneficiary), Reason, Result, Means (Instrument), Source.

Not all components are about actions of course. Components like Person (both natural and legal), Document, Physical Objects are also needed. Components that stay more or less the same during their lifetime are called 'endurants' in philosophy. Endurants can be the (in) direct objects of actions. A person can be arrested, jailed, sentenced etc. This is to be modelled as Arrest – Person and not the other way round, starting with Person and making an association with Arrest. A notable exception is Natural Person which has associations with the life-events Birth, Demise and Marriage.

Role, state and phase

Specializations of Natural Person like Suspect, Claimant, and Defendant are not modelled as separate classes in the logical model. For example, a person can be a suspect in some case, and when he is no longer a suspect, he still is a person. Suspect denotes a role of a Person in a crime. Claimant denotes the role of a person in a claim. Roles of Natural Persons in general are to be modelled as Natural Person with a Role.

A natural person starts life as a baby and progresses to be an adult. In between he is a child, a teenager, an adolescent. These are all phases. Phases are one-way in time. There is no going back and that aspect makes a phase different from a state.

Some things in the world can exist in different states. An order for example can be received, accepted, under consideration, withdrawn, rejected, fulfilled. The order of states is not fixed and not all states need to apply. It is possible to skip some of them.

Part-whole relationships

Part-whole relationships are allowed, like Name with parts Forename and Surname. Hierarchical relationships are allowed, like in Sanction with associations of Penalty and Measure. If the parts have (for the greater part) the same attributes in common, the parts should not be expressed as components. If Penalty and Measure share most of their attributes, they are not to be expressed as

components. In that case Sanction takes an attribute Type with possible values of Penalty and Measure.

Identifiers

Member States may have national solutions for assigning identifiers for Natural Person, legal persons and things like cars, courts. Persons are usually registered in Registries, and in most countries have special identifying numbers like civil service number or social security number. Cars are registered in special registers. Cars have codes engraved somewhere on the body of the car, the Vehicle Identification Number.

Instances of an action can be identified by place and time and other properties. An instance of an action can also be uniquely identified by assigning a uuid of a guid.

If unique identifiers for persons must be included in data exchange, the identifier should be modelled in the following way: a component Identification contains an identifier of type string, and a code telling how the identifying string is to interpreted.

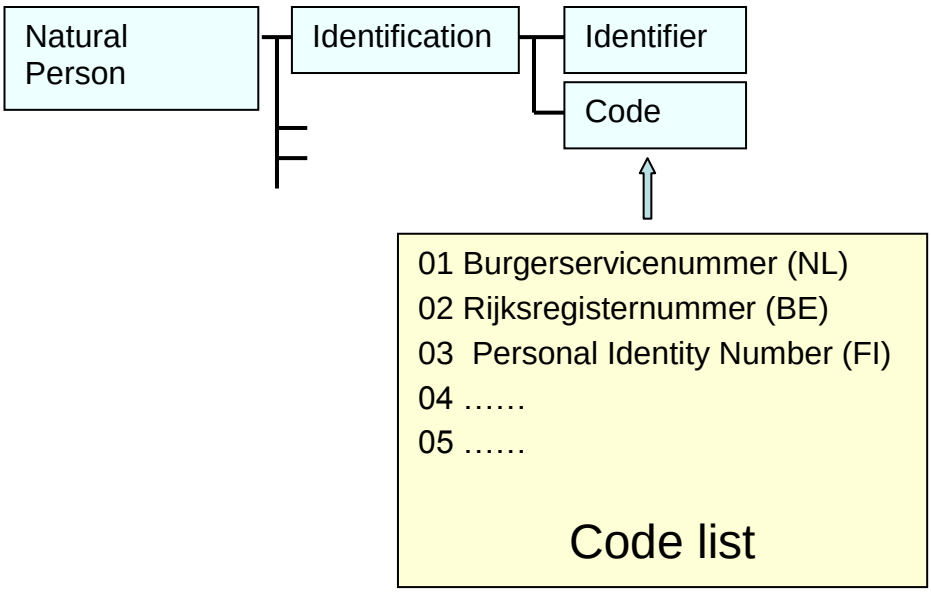

Figure 9: How the identifying string is to interpreted[25]

This modelling of Identification should be applied when Member States have their own system of assigning identifiers and identifiers need to be exchanged.

[25] from Dutch Judicial Information Agency unpublished documents

Data type Property vs. Object Type Property

In modelling a component, it can be a problem to choose between a data type property or an object type property. For example, in modelling 'Identification document' the issue date and the issuing authority and country usually are important. One way of doing this, is shown in A. This is a 'flat' structure. In B a separate component 'Issue' is created with its own components.

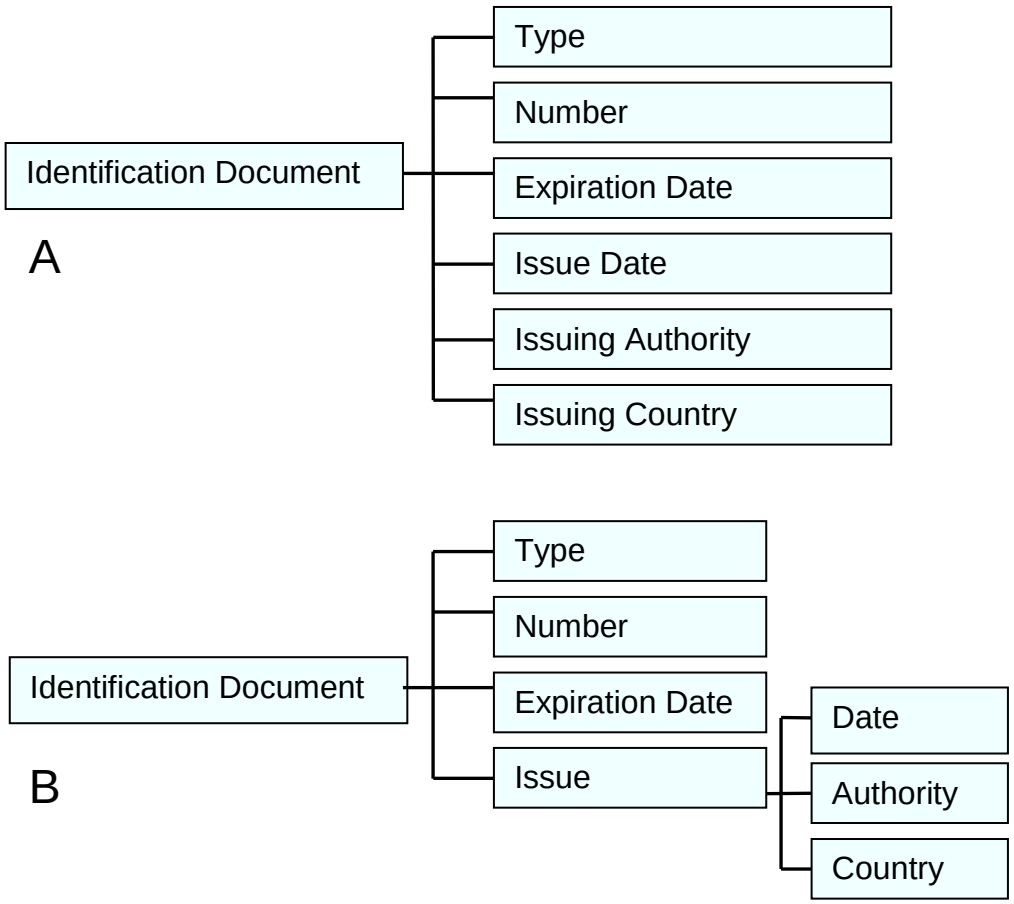

Figure 10: Data type Property vs. Object Type Property[26]

Model A should be chosen when no further attributes are needed in the foreseeable future. Model B should be chosen when not sufficient information is available as to the need for additional attributes.

5.7. BB-7 Tool for logical model composition and maintenance

The tool for the composition and maintenance of a logical data model should allow the user to create a repository of components that can and should be used in message specification. The degree of standardization stipulates the domains for which a set of shared components can be used as basis for messages. As mentioned before, non use case specific elements are expected to be standardized from the start of the operational phase of e-CODEX. The standardization of use case specific components that resemble components from other use cases but differ in the associated forms is

[26] from Dutch Judicial Information Agency unpublished documents

subject to discussion within e-CODEX and most probably with DG JLS. However, although standardization might not be implementable from the very start, it still is the main goal of the e-CODEX approach to semantic interoperability.

Requirement WP6_RQ_F17: The tool must support:

- Specification of components;
- specification of attributes and associations of a component with other components complete with the semantics of the attributes and associations;
- creation of components with subsets of the attributes and associations of a component;
- Specification of data definitions in the repository;
- Specification of additional metadata like Dublin core elements, or ASDM (SEMIC –Join-up);
- Creation and maintenance of code lists, with code list elements having at least beginning and end dates;
- W3C qualified datatypes (including regular expressions);
- A choice of header-component that can be added to the message as part of the payload ;
- Specification of business rules;
- A user-friendly GUI
- Versioning

The tool should be free of charge, maintained by a nonprofit organisation

Below a short-list of tooling is presented.

NIEM tooling

The NIEM provides various tools to facilitate the specification of business documents, called "Information Exchange Package Definitions" in NIEM. These tools include the Schema Subset Generation Tool (SSGT), the NIEM Wayfarer and conformance validation tools. As these tools are on-line tools, they are linked to the US NIEM model. A way to create additional NIEM conformant components is mentioned.

CAM is a NIEM tool that can be used as a standalone tool.

Oracle27 has dedicated resources to supporting NIEM. Their intention is to turn NIEM into a better standard. On the Oracle site it is said that "NIEM currently is a collection of complex XML Schema, an attempt to marry modelling techniques and XSD Schema syntax with embedded context in names that is currently facing significant scaling challenges with inconsistencies, too much manual management, slow lifecycles and an incubating dictionary technology". From this judgement it seems significant changes in the NIEM standard are to be expected in the near future.

[27] http://www.slideshare.net/drrwebber/niem-and-oracle-overview-october-2011 accessed on July 18th 2012

CCTS tooling

For CCTS several tools are available. Apart from the tools that were mentioned in Building Block 5 there is the VIENNA Add-In as a possible choice. Parties in Austria have created an add-in for Enterprise Architect. The VIENNA Add-In (Visualizing Inter ENterprise Network Architectures) is an extension to the UML modelling tool Enterprise Architect for modelling B2B collaborations. Using the two technologies a business process modeler can unambiguously define inter-organizational business processes and business documents on a technology independent level. The platform independent models may be used to derive platform specific artifacts for service oriented architectures e.g. BPEL, WSDL, XSD[28].

5.8. BB-8 Tool for composing data exchange messages according to NDR

The tool should allow the user to create messages by first specifying a root element and adding as many of the components that were defined in the tool of BB-7 as needed. The tool should allow importing components that are created in BB-7. The tool should support versioning, cross-domain selection of components to make up the business document.

It should support creating a subset of a business document in the way of the Northern European Subset specification. Creating schemas the NES way allows parties to 'prune' a schema for general use down to the parts that are needed in a specific exchange, customizing the schema to provide the elements that are really used, thus avoiding needless complexity. This also allows the introduction of additional elements in a general schema. This would typically be done in cases where the additional elements have no meaning for some of the parties that have implemented the schema. (The way to make additions, is to do them in the Core Components first, then adjusting the BIE's, followed by creation of the schemas which can then be customized by picking the desired items).

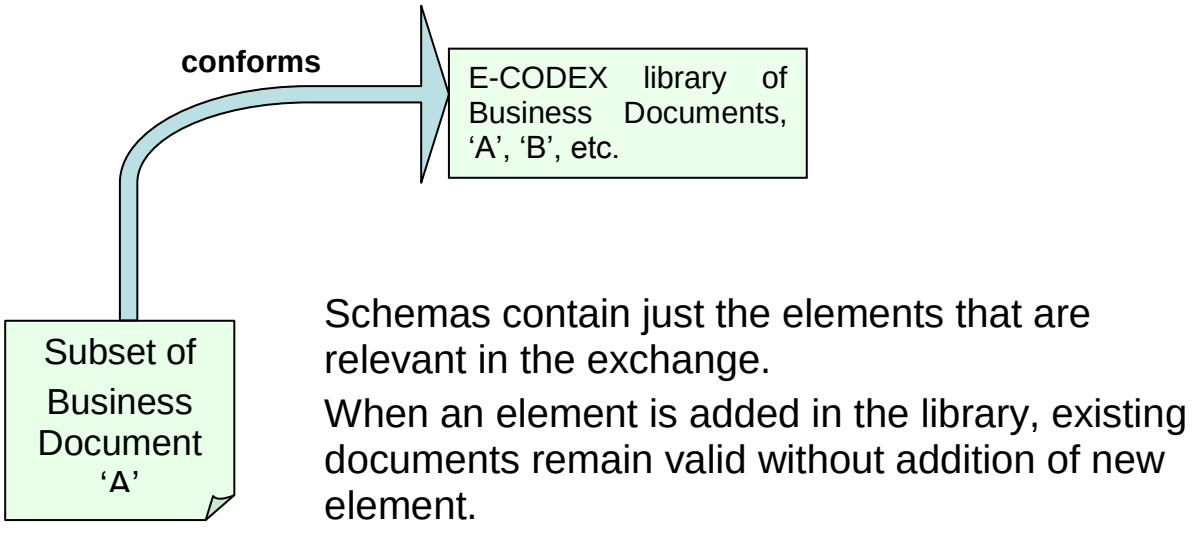

Figure 11: Library function[29]

[28] http://code.google.com/p/vienna-add-in/wiki/VIENNAAddInOverview accessed on July 18th 2012

[29] from Dutch Judicial Information Agency unpublished documents

Requirement WP6_RQ_F18: The tool should allow the user to create messages by first specifying a root element and adding as many of the components that were defined in the tool of BB-7 as needed. The tool should allow importing components that are created in BB-7

Requirement WP6_RQ_F19: The tool should support versioning, cross-domain selection of components to make up the business document.

Requirement WP6_RQ_F20: The tool should support creation of a subset of a business document in the way of the Northern European Subset specification.

Schemas themselves, when used in actual exchange of data, usually do not specify the exchange completely without the addition of Business rules. Schematron[30] should be used to impose constraints on the contents of the message not specifiable with XML-Schema

Requirement WP6_RQ_F21: . The tool should support Schematron, to impose constraints on the contents of the message not specifiable with XML-Schema

5.9. BB-9 NDR and for message composition template(s)

Naming and Design rules are methods to consistently derive a specification of the physical (technical) implementation from the logical specification. It is about general naming rules, like the use of Upper Camel Case and Lower Camel Case, naming of elements, decisions about global definitions versus local definitions. Logical standards usually come with their own set of NDR. The NDR should be proven, so it is not advisable to pick the latest version of the NDR that goes with the logical model of choice. The NDR should be maintained by an international nonprofit organization.

A message should be composed in a consistent, predictable way. A template for message composition is needed, in which the parts of the message like internal header(s), metadata of attachments and additional required parts are specified.

In addition to the headers for transportation, a message should have a header that can be considered part of the payload. This additional header stays a part of the message and is not discarded at receipt of the message. The template for message composition should specify this header. The reason for having an additional header is to not lose the information about the sender, intended receiver, time of sending and other metadata that is needed in the processing of the message. The template should specify in what way the header is to be part of the business document and which parts of the header should be filled and how they should be filled. As it is good practice to use a standard when a standard is available, the UN/CEFACT Standard Business Document Header could be used. The **Standard Business Document Header (SBDH)** is a protocol-independent message format based on XML for the exchange of eBusiness messages.[31]

[30] http://www.schematron.com/ accessed on July 18th 2012

[31] the official SBDH document: UN/CEFACT, Standard Business Document Header, Technical Specification 3.0, ODP step 3 Working Draft, Revision [A1]), 30 August 2010

The token as proof of successful verification of the identity of the sender is to be realized as a SAML token. The message template should provide a field for a receipt proving payment of court fees.

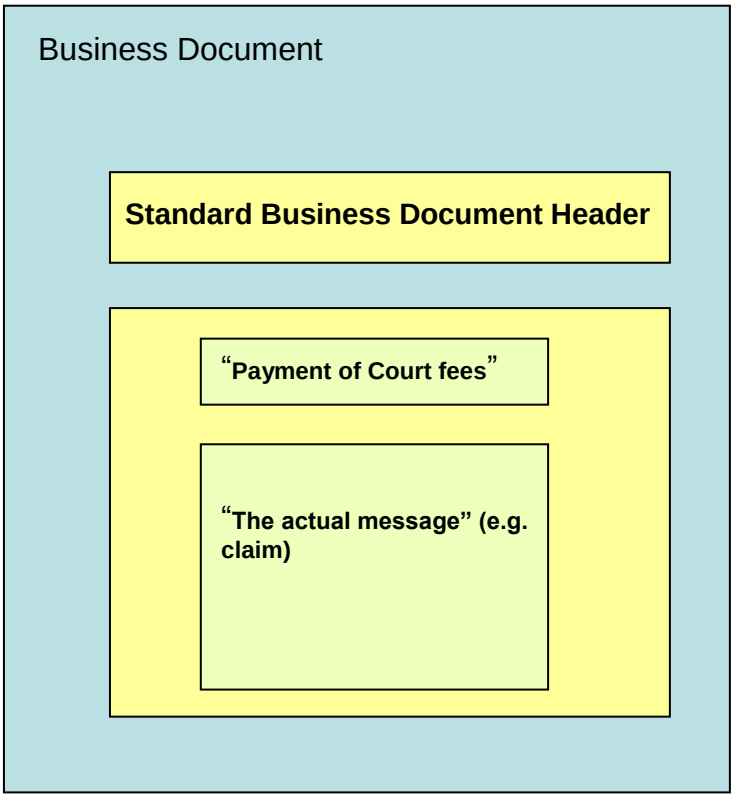

Figure 12: General outline of the template for the payload[32]

The template starts off with an internal header, followed by a structure for the receipt of Court Fees, and the actual message itself. The token that is to be provided by the national solution for verification of identity is taken care of by a SAML assertion, so the token is no concern of WP6. The receipt for payment of the court fees should contain data to identify the case being filed. The payment data does not play a role in criminal justice related messages.

The third part is the actual message, e.g. claim or arrest warrant itself. The metadata describing attachments, which can be pdf or in general any format that has been agreed on, is part of the Business Documents.

Requirement WP6_RQ_F22: The message template should contain parts for an internal header and a part for transactional data, next to the actual payload.

5.10. BB-10 Method to generate unique id's

The concepts, their logical complement and the specifications of document respectively messages are all unique. All these items may have different versions that should be uniquely addressable. To prevent ambiguity they carry a unique id.

[32] from Dutch Judicial Information Agency unpublished documents

Multiple models may incorporate resources with the same name and a slightly different meaning. To prevent name collisions the local name must have in addition a global part that is used for all resources part of the model and known to be unique, the namespace.

The W3C standard Unique Resource Identifier (URI) and its two flavours Unique Resource Name (URN) and Unique Resource Locator (URL) are the foundation for generating unique ID's. All three give structure to the namespace and local name part.

Requirement WP6_RQ_F23: The W3C standard URI is in effect

The local name for a concept, the logical complement and the XML data element equivalent must be recognizable through all the three layers. The concept name stays recognizable.

For this the namespace and local name the concept carries may be slightly rearranged in the logical model according to the selected naming and design rules. The physical model preserves the logical model naming conventions.

For ISO 11179[33] components have a naming convention that makes properties exclusively bound to their aggregate. Aggregates with the same name must belong to different packages. A package is a set of components belonging to a project or domain.

Preferably the packages and the conceptual models share an identical namespace. Because packages are modular in nature the XML schema has a target namespace that corresponds with the package namespace.

The conceptual model

Concepts, part of different conceptual models eventually become components that are part of a message. The concept and the logical complement must have a unique id to prevent ambiguity. The namespace of a RDF/OWL named graph (conceptual model) in combination with the local name make a concept unique. For this a naming convention for the named graph and the local name must be present.

The logical model

CCTS 3.0 NDR (compliant with ISO 111779) is the standard that is selected for the short term. Both the logical model and the XML schema show compliancy with this set of NDR.

The electronic message

The message content which is exchanged shows compliance with a unique XML Schema. A unique id may be attached according to the communication protocol ebMS 3.0. This to prevent confusion with previously sent messages. For WP6 the unique identification concentrates on the content. In analogy with a paper letter where the recipient (front) and sender (back) are written both on the envelope the data put on the envelope is for routing purpose. Once the letter is received the letter is removed

[33] http://metadata-standards.org/11179/ accessed on July 18th 2012

from the envelope. The technical id is lost in this process. A good practice is to repeat unique ids and addresses on the letter itself. In most cases the legal case number given by the competent court in combination with date, claimant name and form type make the content uniquely addressable. Further alignment with WP5 must take place.

Practice

The XML documents are unique and must show compliance with the XML Schema they reference. The data elements being part of a XML document are all unique within the context of the document. Their declaration in the XML schema is made unique through a combination of a unique target namespace and the XML element name. The concept that is the foundation for the data element declaration carries a namespace that contains a substring to be found in the 'targetnamespace'. The concept 'localname' and the logical counterpart start with the same substring. The concept 'localname' stays recognizable.

An example

An ontology is constructed importing a number of named graphs. The named graphs have a unique namespace, name and a date or version.

The named graph name namespace is for instance:
http://www.ecodex.eu/eu/ontology/smallclaim/en/1.0.1.1/

The concept is an expression URN with a local name part, separated by a "#"

To emphasize that a named graph is expressed in the English language in addition the language code 'en' is attached to the namespace.

For a concept the URN becomes:

http://www.ecodex.eu/eu/ontology/smallclaim/en/1.0.1.1/#Person

The logical model inherits a part of the namespace that ensures the uniqueness.

The application and version may differ.

For CCTS the birth date is expressed as 'Person.BirthDate.Date'. Birth date is a property of person and exclusively bound to person. Date is the data type.

The XML schema is the technical manifestation of the logical model and therefore the naming rules are the same.

A CCTS NDR 3.0 URN[34] namespace resembles:

URN: urn:<organization>:<organization hierarchy>[:<organization hierarchy level>]*:<schema type>[:<package>]+:<major>:<status>

[34] UN/CEFACT XML Naming and Design Rules Technical Specification Version 3.0

The XML element becomes for instance
urn:eu:data:smallclaim:0.0.0.1:draft

The XML document that is an instance inherits the target namespace defined in the XML schema. In the CCTS 3.0 NDR XML document the nested declarations may be qualified or unqualified. If unqualified is set the namespace for the local parts is empty.

5.11. BB-11 Repository complete with legal body for maintenance

e-CODEX WP6 aims to provide a sustainable methodology for semantic interoperability supporting the meaningful exchange of information on the domain of Justice. So far D6.3 described the methodology and some aspects of tooling. In order to support wide usage of the e-CODEX approach and to guarantee the sustainability of the semantic interoperability an easily accessible and maintainable repository is needed. All elements of the 3-layer framework to semantic interoperability are ideally part of that repository. The conceptual model, the logical model, and the physical model should be kept in a repository. The repository is available for all those that work on any part of interoperability. The repository not only stores the components, but also supports versioning of elements and messages according to a maintenance procedure. The procedure is to be developed through consultation of the stakeholders.

The repository is not intended to be used in just e-CODEX, but is to be a repository for the justice domain for future projects and LSPs, with future projects adding to the contents the components they find missing. There should be a nonprofit organization taking care of the repository, preferably one that already has some experience in hosting a repository. The repositories should be accessible over the internet for all interested parties to see, and for a moderated community of government workers in data modelling to change. Join-up, the successor to SEMIC, seems be a good candidate for hosting the repository. Also the new IT management organization in Estonia is an option.

5.12. BB-12 Container format

Neither research by the Legal and Security Subgroup nor any other research performed because of e-CODEX has resulted in demand for a container as an asset to achieve semantic interoperability. WP6 therefore labels this building block out of scope. The labelling is not meant as a definite outcome for the long term strategy. However, for the life-time of e-CODEX the container is considered to be out of scope.

5.13. BB-13 Tool to support container life cycle management

The outcome of 5.12 results naturally into labelling life cycle management of a container out of scope as well.

5.14. BB-14 Mapping methodology

For system to system communication the pivot messages to be sent are neutral for both receiving and sending party. Neutral as that the structure is both system and technique independent and for

all EU Member States. The national solutions probably have their own coding schema regarding data elements. Therefore all parties have to map the pivot message and the individual components which are part of the message, to the national specific format. The pivot message specification is the template and guideline about the data that is expected to and that will be exchanged. Mapping decisions on a national level will be based on these facts.

The mapping is a transformation process where data elements may receive a different naming and coding. The data element may have a different label and the data type may follow different conventions for each technical system. For instance the data types date or number may have different separators, field length and fieldnames e.g. '17/sept/2012' and '17-09-12'or '10.9' and '0010,90'. In cases where code list are in use the arrangement of codes may be different for each Member State. The set of codes part of a code list and the individual code value and literal may differ. In addition to transforming the Member States may decide to enrich the message with additional data essential for further processing.

The Member States decide when, if and how messages are transformed from the European level to the national solutions. There are two moments to perform this transformation. First, at the national gateway the message and the envelope are transformed. Second, the transformation may be postponed until the time of receipt at the end point, the competent court. A rather putative option is to leave the message intact.

The way an envelope is structured depends mainly on the communication protocol. For e-CODEX that is ebMS. The individual data elements and their grouping depend on the use case and the modelling decisions. The XML message structure depends on the selected set of naming and design rules (NDR). For the short term NDR of CCTS are chosen for the e-CODEX level. For communication between the national gateway and the competent court or personal mailbox each MS will apply its own set of rules.

For e-CODEX the use cases demand that the forms are presented in the language accepted by the competent court. For this the structure and the data elements are fixed for the forms. The labelling and data values are language depended. The tags or labels from the pivot message are presented in English. The transformation, for which the Member States are responsible, has therefore a major focus on the data format. For this attention must be paid in transforming the tags or labels to the language of preference.

6. Towards XML Schemas

This chapter describes the steps to be taken in order to be able to create the required XML Schemas based on the information requirements articulated in the output of the process analysis activities. It is not about address standards or structures to be used. These have been described sufficiently in the previous chapters. Here a procedure is provided for staff to perform their duties using the standards that have been chosen. The procedure distinguishes between roles, responsibilities and tasks.

The description of the procedure is about what steps have to be taken and not how these steps are executed. The actual execution of these steps is dependent on the choice whether or not a 'standardized approach' for the creation of schemas is chosen and if so, what standard will be applied.

The general steps to be taken are:

Step	1	2
1. Process articulated information requirements with process analyst responsible for the use case description	✓	✓
2. Check if existing building block for a certain (set of) information elements required for the use case are available for reuse in your semantic library	✓	
3. Reuse or extend available building block, according to agreed standards, NDR, etc. Else create new building block	✓	
4. Add definition and source for newly created information entities	✓	✓
5. Add properties for newly created information entities	✓	✓
6. Add code lists	✓	✓
7. Select the use case required building blocks from your library into a XML Schema concept	✓	
8. Delete the information entities contained in the selected building block, but not relevant for the XML Schema	✓	
9. Generate the XML Schema to be implemented by participants	✓	✓

Table 7: Steps to create XML Schemas

1 = scenario where a semantic library is used and maintained for creating XML Schemas

2 = scenario where concepts / information entities and code lists are created specifically for one use case and not necessarily reused in other use cases.

The process of schema creation will raise questions about the sources to be used for data definitions and code lists. Also some kind of order of preference regarding these sources will surface. Clearly there is a need to come up with criteria to assess these sources. Sources already identified are legislation, ISO, EuroVoc, JoinUp or national sources. Of course there are many others. The aim is to provide the administrators of the semantical library supporting e-CODEX to have and to use common guidelines to their disposal, aiding different staff members to create concepts and code lists in a uniform manner.

Creating and maintaining a common semantic library is not easily done with a relatively large staff. Although guidelines will aid the decision making, it still requires human intelligence. Consequently many people will have equally many opinions. A 'best practice' in overcoming extensive discussion on those opinions, is establishing a core-team of data modellers, responsible for creating, editing and extending the concept (core team/concept creators). This limited amount of staff members create the concepts based on the articulated information requirements from the use cases. A created concept is presented to a semantic user council in order to approve a concept for use. The semantic user council is formed by all stakeholders of the semantic library.

Approved concepts can be deployed in schemas to be created. A larger group of people than the core-team is expected to be involved in the creation of the schemas (schema creation group). They will go through this semantic warehouse and put the concepts of their choice in their shopping cart. Out of the contents of their shopping carts, the user will create the schemas. It will be possible to delete information elements form the selected concepts (in the semantic library they will not be deleted of course) and a tailor made schema now can be created.

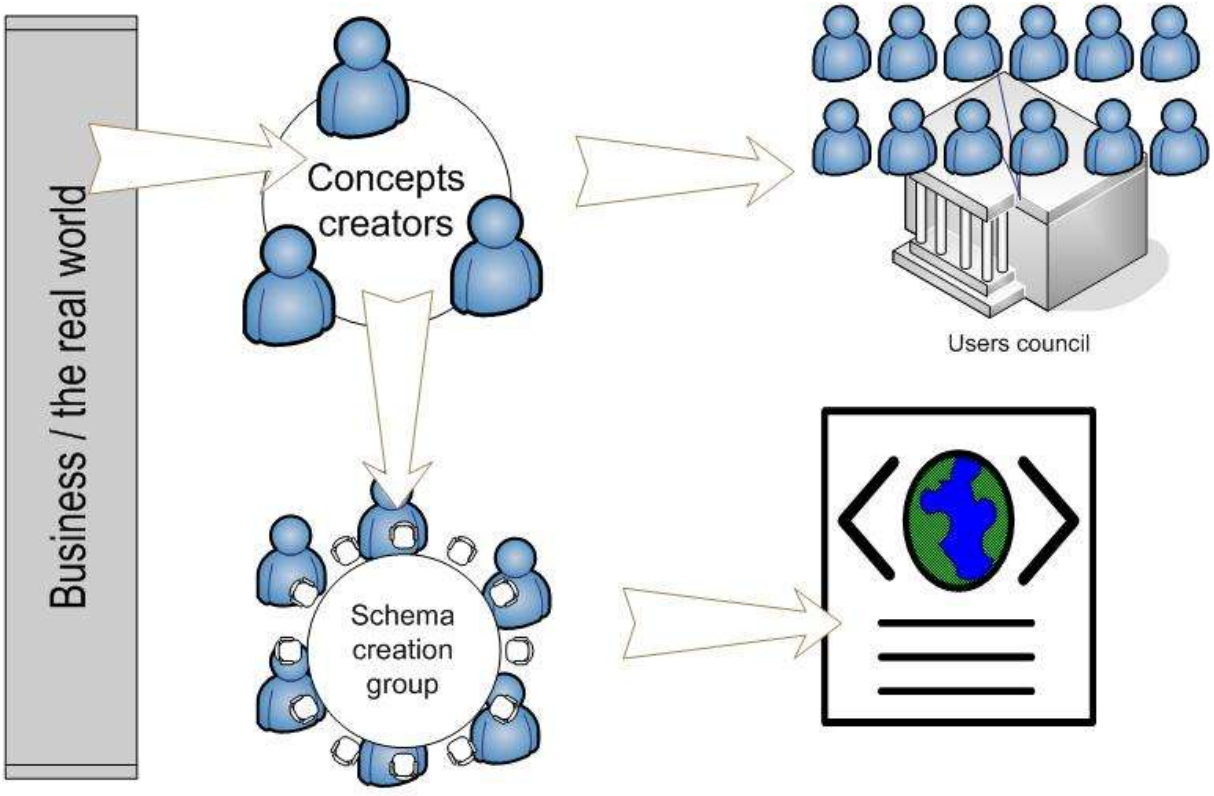

Figure 13: From business to XML Schemas

Conclusions

D6.3 defines specifications to enable meaningful exchange of information between the systems supporting cross border legal procedures by providing semantic interoperability. Because of the demand for operational evidence of the e-CODEX solutions from November 2012 WP6 has to deliver XML Schemas on short notice. However, e-CODEX also aims for long term sustainability demanding for a comprehensive approach. This paradox has been overcome through the agreement on having short and long term strategies as well as on the content of the short term strategy.

The short term strategy for WP6 aims at delivering XML Schemas on time to have e-CODEX support a maximum of 5 use cases as of November 2012. It will be possible to keep oversight no matter the diverging character of the XML Schemas for these 5 use cases. However, the amount of legal procedures to be supported and the wide array of concepts in the domain of Law will eventually lead to the conclusion that a conceptual model, based on a widely accepted standard is inevitable. The current status of e-CODEX allows for the conceptual model to be limited to a piece of paper. Regarding the choice between NIEM, CCTS and other not yet identified methods for data modelling WP6 has chosen CCTS to be applied for the short term implementation.

The conceptual model, the logical model, and the physical model should be kept in a repository. The repository is available for all those that work on any part of semantic interoperability. The repository not only stores the components but also supports versioning of elements and messages according to a maintenance procedure. The versioning regards the different levels of models including the cross referencing between the levels. The procedure is to be developed through consultation of the stakeholders.

The XML Schemas will be created from CCTS through the use of the Metadata Workbench. The initial modelling of CCTS and XML Schemas is done by *WP6 core team*. This core team or concept creators will identify and model the appropriate concepts. These concepts will be reviewed by the *WP6 user council*. The consecutive development of XML Schema for a use case is done by the *schema creation group* based on the input from core team and user council.

D6.1 and D6.2 already mentioned that XML does fulfil all requirements set by WP6. The remaining issue was if XML documents would be legally valid. This issue arose as automated processing of information requires a deeper structure from forms as is available from the available (PDF-) forms. The Working group on e-Law (e-Justice) in their meeting at April 16th supported the idea to use a deeper structure at the transportation level to allow for automatic processing of the legal procedures supported by e-CODEX for the life-time of e-CODEX. As a precondition the working group asked e-CODEX for technical guidelines on how to deal with the issue in the future.

The next paragraph is dedicated to decisions made by WP6 without extensive discussion. The decisions are considered straightforward by WP6 participants. The decisions involve the formal action to apply widely used standards for e-CODEX. The exception is the use of ISA Core Vocabulary Person. The Core Vocabularies as semantic assets however, are focal point of ISA, the EC standardization program. The Large Scale Pilots have been asked to participate in establishing the Core Vocabularies and apply these in the use cases. For the purpose of clear presentation the decisions have been listed underneath. In order to be operational by November 1st the next additional decisions were necessary:

- Use of the UN/CEFACT Standard Business Document Header (SBDH)
- Use of the Dublin Core metadata set
- Use of ISA Core Vocabulary Person

Other decisions are necessary to support the long term sustainability of the semantic assets of e-CODEX. The full impact of these decisions is not yet clear. Therefore additional research into the impact will start after summer 2012. The topics yet identified are listed underneath.

- Adoption of other ISA Core Vocabularies as integral part of the e-CODEX semantic assets
- Assemblance of a conceptual model (RDF, OWL)
- Final decision on standard for data modelling (CCTS/NIEM)
- Choosing an organisation for hosting the repository
- Application of EPOC IV data model for the EAW use case
- SAML token as the token to proof successful verification of the identity of the sender
- Impact on distinction between modelling for a domain and modelling for specific documents
- Composition of unique identifiers

I. References

References are included as footnotes in each chapter.

II. Appendix I

In this Appendix the different requirements and recommendations described in the different chapters of this document are summarized.

Identifier: *WP6_RQ_T1..12*	**Name of Requirement: Conceptual modelling tool requirements**
Description	A set of requirements for a conceptual modelling tool.
Comments	*T1:* Open Source *T2:* Support for the open standard RDF/OWL or RDF/OWL in combination with SKOS *T3:* Multi-language support *T4:* An advanced editing mode, web based and with lexical support *T5:* Import and export ability *T6:* An intuitive graphical presentation, graphs view *T7:* The ability to inference and to check consistency *T8:* The ability to collaborate c.q. multi user mode *T9:* Support for refactoring and transformation, merging *T10:* The ability to design modular *T11:* The presence of an open API *T12:* The ability to connect with a tool independent data repository
Reason / Benefit	These requirements allow the user to specify the conceptual model in a tool.

Identifier: *WP6_RQ_F13*	**Name of Requirement: ISO 11179**
Description	ISO 11179 should be adopted as the standard to construct a logical model.
Comments	
Reason / Benefit	This standard is widely accepted as the best standard for constructing logical models.

Identifier: *WP6_RQ_F14*	**Name of Requirement: Use of code lists**
Description	Code lists should be used for at least Country, Nationality, Currency, City, Gender and to

	denote different types/ kinds of Components, unless these special kinds have additional properties that need to be in the message.
Comments	
Reason / Benefit	The use of Code lists prevents misspellings and restricts the number of components in the model. A smaller model is to be preferred because it can be better understood.

Identifier: *WP6_RQ_F15*	**Name of Requirement: Header component**
Description	The logical model should contain a header component.
Comments	
Reason / Benefit	The header of the transportation layer is often discarded. The header component that is part of the message provides the metadata of the message.

Identifier: *WP6_RQ_NF1*	**Name of Requirement: Open source and offline tooling**
Description	The tooling for maintaining the logical model and creating message specifications should be open source and free of charge
Comments	
Reason / Benefit	Vendor lock-in is to be avoided.

Identifier: *WP6_RQ_NF2*	**Name of Requirement: EIF compliancy for standard adoption**
Description	The standard that is to be adopted should be maintained in a way that is compliant with EIF.
Comments	
Reason / Benefit	Compliance with EIF is mandatory.

Identifier: *WP6_RQ_NF3*	**Name of Requirement: Adoption of modelling standard**
Description	The modelling standard that is to be adopted in e-CODEX should not deviate from choices made previously in European programmes.
Comments	
Reason / Benefit	An adopted standard gives benefits unless there are severe problems with the standard.

Identifier: *WP6_RQ_F16*	**Name of Requirement: Modelling rules in addition to ISO11179**
Description	*Modelling* is to be done according to a set of rules, in addition to the ISO 11179 to ensure consistent and sustainable modelling.
Comments	
Reason / Benefit	ISO 11179 does not provide enough guidance. Both NIEM and CCTS are based on ISO 11179, but they cannot be used together in one model.

Identifier: *WP6_RQ_F17*	**Name of Requirement: Logical modelling tool requirements**
Description	The tool must support: • Specification of components; • specification of attributes and associations of a component with other components complete with the semantics of the attributes and associations; • creation of components with subsets of the attributes and associations of a component; • Specification of data definitions in the repository; • Specification of additional metadata like Dublin core elements, or ASDM (SEMIC – Join-up); • Creation and maintenance of code lists, with code list elements having at least beginning and end dates; • W3C qualified data types (including regular expressions); • A choice of header-component that can be added to the message as part of the payload ; • Specification of business rules; • A user-friendly GUI • Versioning • The tool should be free of charge, maintained by a nonprofit organisation
Comments	
Reason / Benefit	These requirements allow the user to specify the

	logical model in a tool.

Identifier: *WP6_RQ_F18*	**Name of Requirement: Allow specifying root element**
Description	The tool should allow the user to create messages by first specifying a root element and adding as many of the components that were defined in the tool of BB-7 "Tool for logical model composition and maintenance" as needed. The tool should allow importing components that are created in BB-7.
Comments	
Reason / Benefit	Manual composition of messages is error prone.

Identifier: *WP6_RQ_F19*	**Name of Requirement: Support of versioning and cross domain selection**
Description	The tool should support versioning, cross-domain selection of components to make up the business document.
Comments	
Reason / Benefit	Changes to the model are inevitable. The tool should provide the means to deprecate parts of the model. Components must be specified in one domain only, if the meaning is not different. This creates the need to be able to use components that were specified in other domains.

Identifier: *WP6_RQ_F20*	**Name of Requirement: Support according Northern European Subset specification**
Description	The tool should support creation of a subset of a business document in the way of the Northern European Subset specification.
Comments	
Reason / Benefit	A business document built out of standard components tends to be large, while a part of the specification may not be relevant in that specific business document. Creating a subset makes the business document more manageable.

Identifier: *WP6_RQ_F21*	**Name of Requirement: Support of Schematron**
Description	The tool should support Schematron, to impose constraints on the contents of the message not specifiable with XML-Schema.
Comments	
Reason / Benefit	The tool should let the user specify the rules without him having to worry about the schematron syntax.

Identifier: *WP6_RQ_F22*	**Name of Requirement: Support of internal header and transactional data in message template**
Description	The message template should contain parts for an internal header and a part for transactional data, next to the actual payload.
Comments	
Reason / Benefit	These are separate parts that must be put in the message in a standardized way.

Identifier: *WP6_RQ_F23*	**Name of Requirement:URI**
Description	Unique Resource Identifier.
Comments	The W3C standard URI is in effect
Reason / Benefit	A standard for unique referencing resources.