An Ontology-Driven Approach to Electronic Document Structure Design

Denis A. Nikiforov(✉), Alexander B. Korchagin, and Ruslan L. Sivakov

Centre of Information Technology, Ekaterinburg, Russia
{Denis.Nikiforov,Alexander.Korchagin,Ruslan.Sivakov}@centre-it.com

Abstract. Over the course of history, humankind used documents as one of the ways of organization of the data. In the recent decades, electronic documentation became increasingly widespread. To make electronic documents exchange possible, standards regulating transmission protocols, representation formats, and rules for document building are necessary. For some protocols (HTTP, SOAP, etc.) and formats (EDI, XML, JSON, etc.), relatively fixed and generally accepted standards are available. As for the electronic document design, there is an abundance of approaches where a leader could hardly be established; all of them have their benefits and drawbacks. This study explores some of these approaches (UN/CEFACT CCTS, WCO DM, ISO 20022, and NIEM). These approaches have different features but from the conceptual perspective they are intended to describe sets of details of some real-world objects. The paper proposes to describe such objects using an ontology and then, based on this ontology, build conceptual structures of electronic documents that can be converted to platform-independent structures of electronic documents in accordance with one of the standards. The introduced approach allows harmonizing the standards under consideration.

Keywords: Document engineering · Ontology · Model-driven architecture · Platform-independent model

1 Introduction

Electronic Data Interchange (EDI) is among the first standards in electronic data exchange. It regulates protocols, formats and rules for building electronic documents. The Transportation Data Coordinating Committee started to develop this standard in the 1960s and its first version was published in the 1970s. Later, standards like ANSI ASC X12[1], UN/EDIFACT [3], HL7 [1], and many others were developed based on EDI. To this day, those standards are dominant in electronic commerce [2].

In the 1990s–2000s, with Internet expansion, the focus has shifted from protocols and formats of data transmission to structure and semantics of electronic documents. Moreover, structure is described in platform-independent

[1] http://www.x12.org.

© Springer International Publishing AG 2017
D.I. Ignatov et al. (Eds.): AIST 2016, CCIS 661, pp. 3–16, 2017.
DOI: 10.1007/978-3-319-52920-2_1

form, for example, in UML language. Documents themselves can be presented in different platform-depended languages (mainly, EDI or XML). Main standards of this group are UN/CEFACT Core Components Technical Specification (CCTS) [19], World Customs Organization Data Model (WCO DM)[2], ISO 20022 [4], and NIEM [15]. They are key to Government-to-Government (G2G) and Government-to-Business (G2B) interactions.

In all aforementioned approaches, an electronic document is viewed as a hierarchy of data elements. In the 2000s, with growing popularity of Semantic Web, a step change from data hierarchy exchange to fact exchange was made. These approaches are mostly used in rather complicated specialized environments, for example, in industry standards (ISO 15926 [5]).

However, in electronic commerce and G2G/G2B interactions, "traditional" (hierarchical) document exchange is still dominating. It is strongly associated with specifics of these types of interactions, as their entire standard framework is geared towards exchange of (electronic) documents rather than facts. Also, a concept of legal value applies to documents, not facts. Documents have strictly determined hierarchical structure, and they do not need flexibility of data presentation provided by RDF or OWL. Instead, strict control of electronic document content is necessary.

All the abovementioned suggests that in the nearest future, standards oriented on "traditional" exchange of electronic documents rather than facts will be used in electronic commerce and G2G/G2B interactions. However, current standards in this field have at least two drawbacks.

First, there are too many standards and they are not compatible. In this article, we shall present a method unifying approaches described in CCTS, WCO DM, ISO 20022, NIEM specifications. It allows to design electronic documents structures in compliance with any of these standards.

Second, no readily available and convenient tools for designing structures of electronic documents exist. Either general-purpose editors not really convenient for documents design (UML editors, XML Schema editors), or specialized commercial editors (e.g. GEFEG.FX[3]) are usually proposed as tools. In this article, we shall describe our free editor based on open standards and frameworks.

The paper is organized as follows. In Sect. 2, we shall evaluate four approaches to electronic document structure design. In Sect. 3, we shall propose our ontology-driven approach. Finally, we shall conclude this paper in Sect. 4.

2 Overview of Analogues

While standards in question (CCTS, WCO DM, ISO 20022, NIEM) differ in details, their conceptual approach to design of electronic document structures can be summarized in the following chart (Fig. 1).

[2] http://www.wcoomd.org.

[3] http://www.gefeg.com/en/gefeg.fx/fx_descr.htm.

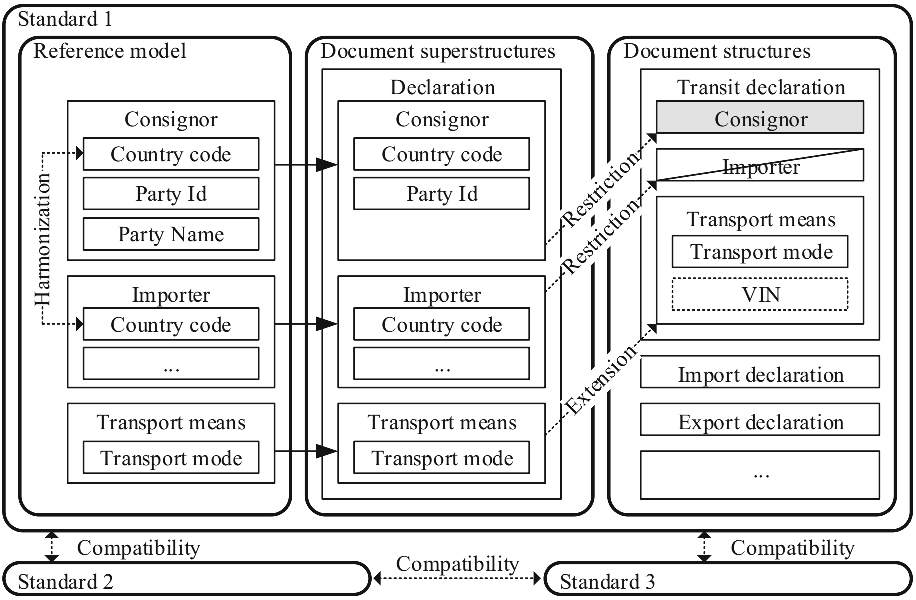


Fig. 1. General design chart of electronic document structures

2.1 Harmonization of Data Elements

All approaches assume creation of a library of data elements or core components. For clarity, let us call it a reference model. This model describes acceptable data elements that can be used in designing electronic document structures.

A reference model does not depend on an application context. In some information exchange, it may be sufficient to indicate a consignor's classification code in a customs declaration while their name may not be necessary. However, if the name of a consignor is necessary in some other document (outside the context of customs control), this element must be defined in a reference model.

The main purpose of a reference model is harmonization of data elements being used in different electronic documents. For example, country code element must have the same description and uncontroversial semantics regardless of the context where it is used.

As for the description of data elements, all standards in question are based on ISO 11179 [6]. It is the framework standard, which defines basic concepts (such as data element, value domain, etc.) and rules of metadata presentation.

As for the rest, specifics of these standards in harmonization are different. CCTS and ISO 20022 allow to harmonize data elements and structures conceptually. However, unlike NIEM, they do not allow to re-use associations, roles and properties of objects. WCO DM gives the least possibilities for harmonization. For example, it describes not a vehicle per se but a number of its characteristics.

2.2 Compatibility with Other Standards

The abovementioned standards are based on metamodels incompatible with each other. For example, CCTS metamodel defines core components and business information entities as basic component blocks for electronic document structures. NIEM metamodel defines object type, role type, association type, etc. ISO 20022 metamodel defines business components and message components. WCO DM metamodel defines classes and attributes. In each of these approaches, structural modelling of electronic documents is performed in different terms.

Differences also exist at the level of reference models. Core Components Library (used in CCTS) and WCO DM are partially compatible due to usage of a common set of unqualified data types, and many data elements are based on ISO 7372 [7]. But ISO 20022 and NIEM use absolutely different sets of data types and data elements, incompatible with other standards.

2.3 Customization of Electronic Document Structures

A reference model is the basis for designing superstructures of electronic documents. In WCO DM, they are also called "base information package". They are subsets of the reference model necessary for data representation in the context of some process.

Structures of electronic documents are designed on the basis of superstructures. As a rule, it amounts to exclusion of excessive data elements from superstructures (e.g. "Importer" in Fig. 1), posing restrictions on mandatory data elements requirements ("Consignor"), and posing restrictions on value domain.

CCTS specification describes two mechanisms of customization: qualification of core components and restriction of context of information entities use. WCO DM recommends to customize information packages at the level of XML schema. NIEM does not assume design of superstructures of electronic documents; new information exchange packages are created by copying and modifying the existing ones.

2.4 Extension of a Reference Model and Electronic Document Superstructures

In many cases, standard may not take into consideration national or other specificities of data exchange. For example, in some countries, a vehicle identification number should be noted in a customs declaration. But this data element is not defined in WCO DM. In this case, a standard must provide a mechanism for extending a reference model and superstructures.

WCO DM allows (but does not recommend) to extend information packages by inclusion of necessary elements into XML schema. ISO 20022 allows to include data with arbitrary XML schema into specific extension points of a message. CCTS and NIEM allow to describe extensions of electronic data structures at a higher (platform-independent) level of abstraction, not directly in XML schema. In CCTS, it is performed by qualification, which allows to describe several derived information entities on the basis of one core component in addition to restricting core components. In NIEM, possibilities of model extension are even wider. It allows to define new entities.

2.5 Design Tools

GEFEG.FX is a basic tool for design of electronic document structures on the basis of CCTS, WCO DM and ISO 20022. It is a closed-source software with proprietary model representation formats. For ISO 20022, Ecore-based metamodel is also accessible. On the basis of the metamodel, an Eclipse plugin for model viewing can be generated. ISO 20022 model is accessible in open XML Metadata Interchange (XMI) format [17]. For NIEM, a whole set of tools for designing information packages is available. However, they cannot be used to design electronic document structures based on other methodologies.

2.6 Comparison Results

We have summarized all the abovementioned and evaluated each of the standards using a five-grade scale in Table 1. All standards in question have limited possibilities of harmonization of data elements, they are not completely compatible with other standards, they have restrictions in customization and extension of electronic document superstructures, and they are not sufficiently equipped with design tools. Hereafter, we shall describe an approach that addresses some of these disadvantages.

Table 1. Evaluation of analogues

Criteria	CCTS	WCO DM	ISO 20022	NIEM
Harmonization	3	1	3	4
Compatibility	3	3	1	1
Customization	4	2	3	1
Extension	3	2	2	4
Tools	1	1	2	3
Total	14	9	11	13

3 Proposed Solution

In all of the examined approaches, electronic document structures are expected to be designed in a form of Platform-Independent Models (PIM). For example, that can be UML models [16] or other Ecore models [4]. After that, Platform-Specific Models (PSM) are generated from PIM, usually in a form of XML schema (Fig. 2).

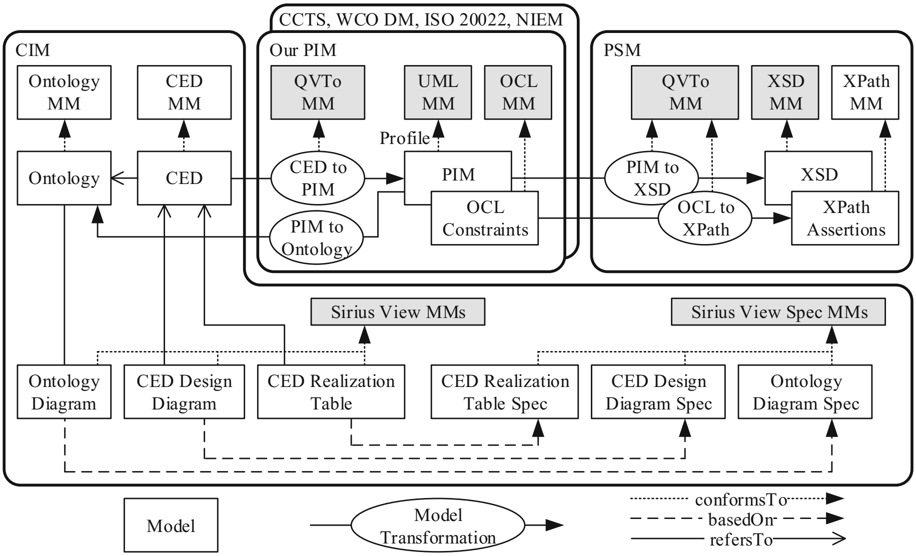


Fig. 2. The proposed approach to design of electronic document structures

A drawback of the existing approaches is a relatively limited mechanism of electronic document structures customization. For example, a transit declaration structure may be required to be built on the basis of some generalized customs declaration (Fig. 1). Herein, excessive data elements must be excluded from the structure, and conversely, other elements must be made mandatory. All studied approaches assume copying data sets available in the model and adding required changes to these copies (removing elements, changing multiplicity, etc.). More complicated restrictions (for example, on the summation value of some data elements or on a value of a data element depending on values of other data elements) are usually described in natural language and then programmers manually implement them in a code.

Our proposal is to replace creating copies with the usage of existing structures accompanied by describing all necessary restrictions in formal Object Constraint Language (OCL) [11]. During implementation of information exchange, these restrictions must be automatically transformed into expressions in some platform-specific language (XPath, SQL, Java, etc.). Examples of OCL constraints and appropriate XPath assertions will be presented in Sects. 3.3 and 3.4. This approach reduces amount of duplicated structures in a model, and minimizes human factor impact on implementation of information exchange [8].

We have been successfully using our approach to support cross-border information exchange between authorities of several countries [10]. However, integration of our information system with other systems based on different standards (CCTS, WCO DM, ISO 20022, and NIEM) is necessary in many cases. In those cases, it is usually proposed to create mappings between document structures.

Instead of mapping, we propose to unify all studied approaches to design of electronic document structures by adding one more level of abstraction, Computation Independent Model (CIM). This level is designated for describing Conceptual structures of Electronic Documents (CED), which are not dependent on any reference model (libraries of data elements, libraries of core components), and they are not dependent on the standard planned to be used during implementation of information exchange.

CED must be designed based on a uniform ontology. This ontology is a generalization of different reference models, but it does not describe data elements or core components used for design of electronic document structures. It describes all real-world objects, which details could be transmitted in electronic documents. It also describes all possible properties and relations of these objects. Our approach is conceptually different from the studied approaches as it clearly separates the ontological level (Sect. 3.1) and the level of data elements and data sets (Sect. 3.2).

To work with ontology and conceptual structures of electronic documents, we propose a tool [9], based on open standards (Meta Object Facility (MOF) [14], XMI [17]) and free frameworks (Eclipse Modeling Framework (EMF) [18], Sirius [20]). Using this tool, a developer can build a conceptual structure of an electronic document, choose a required standard, and automatically generate an electronic document structure compliant to the selected standard using MOF Query/View/Transformation (QVT) [13]. At the moment, only our standard is supported. First, it is a free alternative to commercial tools for designing structures of electronic documents (GEFEG.FX). Second, the need for mapping electronic documents structures is eliminated because they are based on a uniform ontology.

Designing a structure of an electronic documents based on a defined ontology is rather easy. The main challenge of this approach is creation and actualization of an ontology. We have developed a QVT transformation, which helps a developer to create an initial version of ontology based on an already existing reference model [9]. This ontology may contain many duplicated entities; some defined entities may result from conceptual errors. After automatic generation, the ontology must be harmonized manually.

3.1 Ontology

Metamodel. Like any model, an ontology must conform to some metamodel. We decided not to use RDF or OWL [12] and developed our own metamodel (Fig. 3). Being less universal than RDF, it allows to only describe well-defined types of facts but it is easier to work with. If necessary, the ontology can be automatically transformed into RDF representation. Our ontology is very similar to an Entity-Relationship (ER) model, but does not replicate it. The main difference is the ability to reuse properties and roles of objects.

Table 2 describes correlation of structures used for building our ontological model, and those described in other standards.

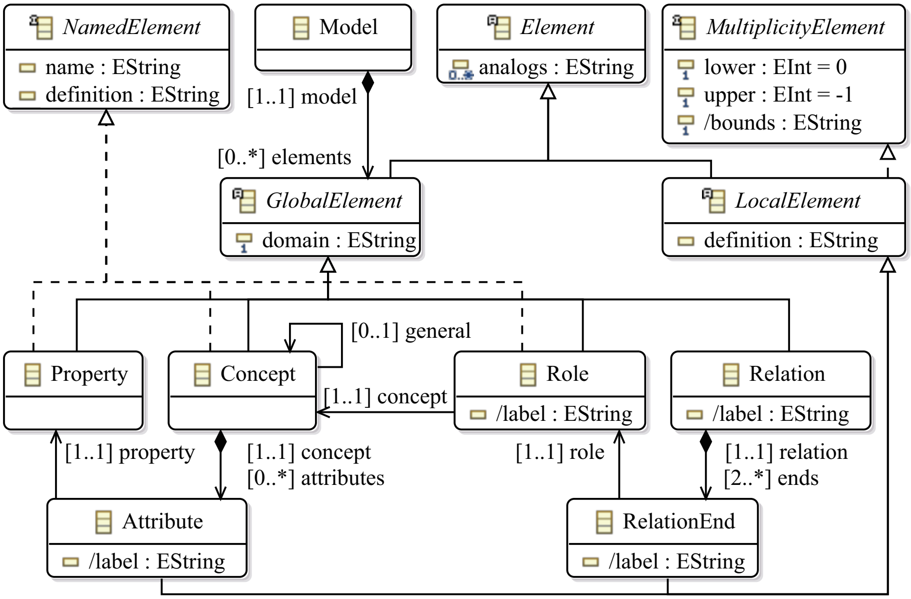


Fig. 3. The proposed ontology metamodel

Table 2. Correlation of structures defined in various metamodels

Ontology	ISO 11179	Our PIM [10]	CCTS	ISO 20022	NIEM	WCO DM
Element	Administered item	—	Core component	Business concept	—	—
Concept	Concept	Complex type	Aggregate core component	Business component	Object type	Class
Property	Property	Simple element	Basic core component property	—	Property holder	Property term
Attribute	Data element concept	Component	Basic core component	Business attribute	Property	Attribute
Relation	Concept relationship	—	—	—	Association type	—
Role	—	Complex element	Association core component property	—	Role type	—
Relation end	—	Component	Association core component	Business association end	—	—

Sample Model. Figure 4 presents a small fragment of our ontology. For simplicity, repeatedly used properties and roles are not depicted. For example, country code attributes (in a business entity and in a subject address) are both based on one global property. Consignor and consignee roles can also be reused. There are different notations for ontologies (RDF graphs, EXPRESS-G, VOWL), but we consider this simplified notation as the most convenient at the moment.

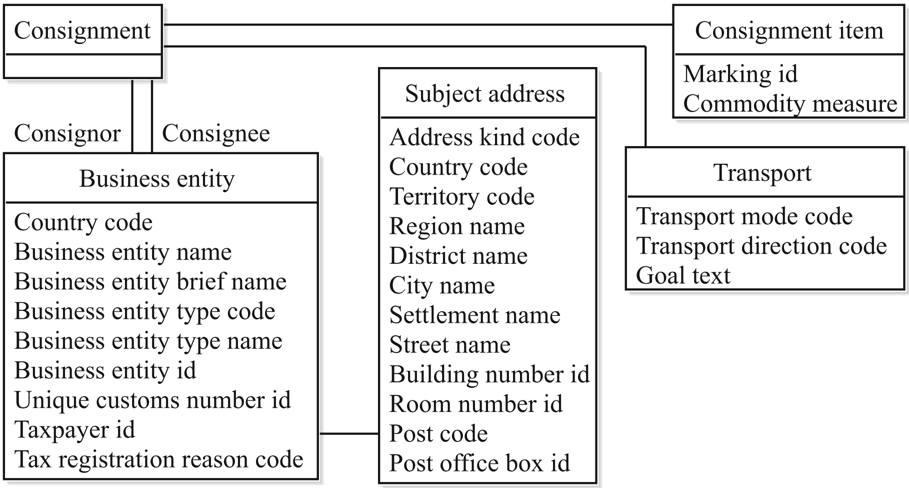


Fig. 4. An ontology fragment

3.2 Conceptual Structure of Electronic Document

Metamodel. An electronic document structure consists of data elements and data sets (Fig. 5). Data sets can include data elements and other data sets.

A data set must be based on some object role as defined in the ontology. Nesting other data sets based on roles connected with the basic role of the first data set is acceptable. It is also acceptable to use data elements based on properties of the object, which role defines this data set. In these cases, a structure of an electronic document complies with the ontology.

If objects, roles, relations or properties are not yet defined in the ontology, and their details must be transmitted in electronic document, then a developer can define new data elements or data sets. In this case, an electronic document structure will not comply with the ontology. The latter must be actualized afterwards.

Sample Model. Figure 6 presents an example of a conceptual structure of an electronic document. In the design view, a developer determines data sets and data elements necessary for this document based on a defined ontology (Fig. 4). Then, in the implementation view, he can (1) specify multiplicity of components; (2) specify their definitions; (3) indicate, which PIM objects they must be implemented with.

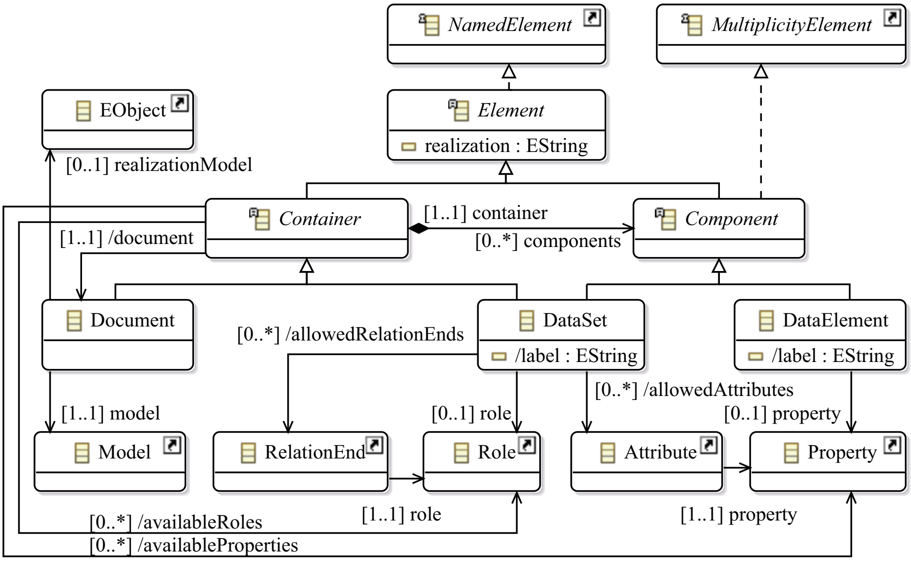
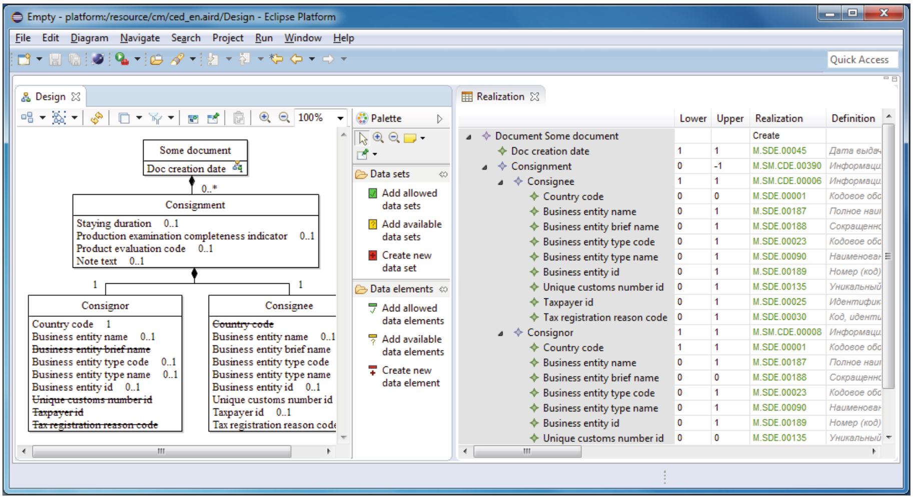


Fig. 5. The metamodel of conceptual structures of electronic documents


Empty - platform:/resource/cm/ced_en.aird/Design - Eclipse Platform
File Edit Diagram Navigate Search Project Run Window Help
Quick Access
Design
100%
Palette
Data sets
Add allowed data sets
Add available data sets
Create new data set
Data elements
Add allowed data elements
Add available data elements
Create new data element
Some document
Doc creation date
0..*
Consignment
Staying duration 0..1
Production examination completeness indicator 0..1
Product evaluation code 0..1
Note text 0..1
1
1
Consignor
Country code 1
Business entity name 0..1
~~Business entity brief name~~
Business entity type code 0..1
Business entity type name 0..1
Business entity id 0..1
~~Unique customs number id~~
~~Taxpayer id~~
~~Tax registration reason code~~
Consignee
~~Country code~~
Business entity name 0..1
Business entity brief name
Business entity type code
Business entity type name
Business entity id 0..1
Unique customs number id
Taxpayer id 0..1
Tax registration reason code
Realization

	Lower	Upper	Realization	Definition
Document Some document			Create	
Doc creation date	1	1	M.SDE.00045	Дата выдач
Consignment	0	-1	M.SM.CDE.00390	Информаци
Consignee	1	1	M.SM.CDE.00006	Информаци
Country code	0	0	M.SDE.00001	Кодовое обс
Business entity name	0	1	M.SDE.00187	Полное наи
Business entity brief name	0	1	M.SDE.00188	Сокращенн
Business entity type code	0	1	M.SDE.00023	Кодовое обс
Business entity type name	0	1	M.SDE.00090	Наименова
Business entity id	0	1	M.SDE.00189	Номер (код)
Unique customs number id	0	1	M.SDE.00135	Уникальный
Taxpayer id	0	1	M.SDE.00025	Идентифик
Tax registration reason code	0	1	M.SDE.00030	Код, иденти
Consignor	1	1	M.SM.CDE.00008	Информаци
Country code	1	1	M.SDE.00001	Кодовое обс
Business entity name	0	1	M.SDE.00187	Полное наи
Business entity brief name	0	0	M.SDE.00188	Сокращенн
Business entity type code	0	1	M.SDE.00023	Кодовое обс
Business entity type name	0	1	M.SDE.00090	Наименова
Business entity id	0	1	M.SDE.00189	Номер (код)
Unique customs number id	0	0	M.SDE.00135	Уникальный

Fig. 6. The example of design and implementation views of an electronic document conceptual structure

3.3 Platform-Independent Conceptual Structure of Electronic Document

Metamodel. Once an analyst has designed a conceptual structure of an electronic document and has determined ways of implementation of its components, it only takes starting the QVT transformation and generating a structure of electronic document based on one of the standards. In the future, it will be possible to generate structures with different methodology (CCTS, ISO 20022, WCO DM, etc.), but for now, only our methodology is supported. To comply with it, an electronic document structure must be a UML model based on UML profile, which is described in [10].

Sample Model. The QVT transformation results in a UML model similar to the one below (Fig. 7).

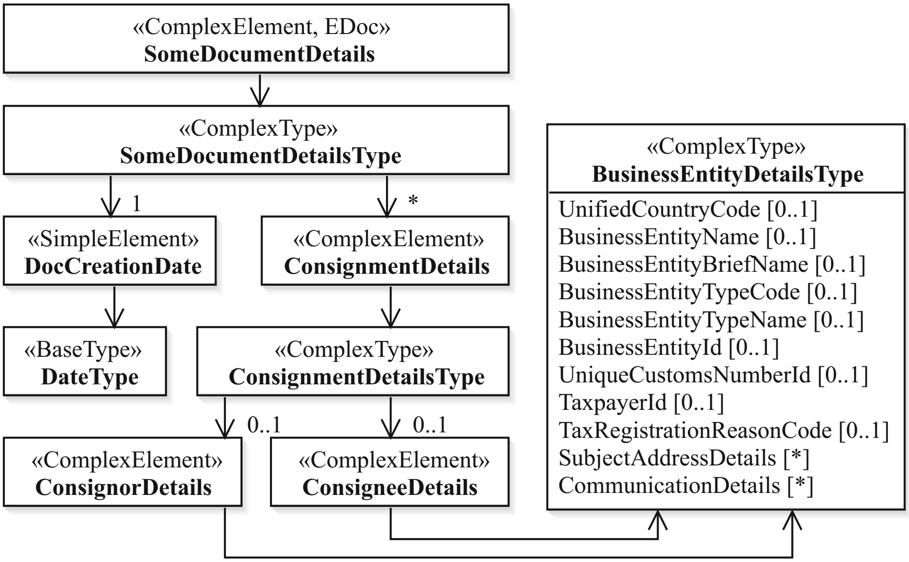


Fig. 7. The example of a platform-independent model of an electronic document

When designing a conceptual structure of the electronic document, an analyst defined required and prohibited details of consignor and consignee (Fig. 6). However, it is not reflected in PIM in Fig. 7. These requirements can be accommodated in one of two ways: (1) by creating separate composite data types for consignor and consignee, where appropriate multiplicity of data elements is specified; (2) by using common type for consignor and consignee while describing additional restrictions in some formal language. We use the latter approach and describe additional requirements in OCL. Examples of these restrictions in natural language and in OCL are presented below.

Element ConsignmentDetails/ConsignorDetails is required:

```
ConsignmentDetails.value.ConsignorDetails.value->notEmpty()
```

Element ConsignmentDetails/ConsigneeDetails/CountryCode is prohibited:

```
ConsignmentDetails.value.ConsigneeDetails.value
.CountryCode.value->isEmpty()
```

Constraints can be complicated; their detailed analysis is out of scope of this article.

3.4 Platform-Specific Model

The next stage is generation of PSM (XML schema, ER model, etc.) on the basis of PIM. With that, OCL expressions are translated in some platform-specific language (XPath, SQL, Java, etc.).

We transform UML models with OCL constraints into XML schemas 1.1 with XPath assertions:

```
<xs:complexType name="SomeDocumentDetailsType">
  <xs:sequence>
    <xs:element ref="DocCreationDate" />
    <xs:element ref="ConsignmentDetails" />
  </xs:sequence>
  <xs:assert test="ConsignmentDetails/ConsignorDetails" />
  <xs:assert test="fn:not(ConsignmentDetails/ConsigneeDetails/
                           CountryCode)" />
</xs:complexType>
```

4 Conclusion

This article studies four standards for design of electronic document structures (CCTS, WCO DM, ISO 20022, and NIEM). The main contributions of this article are as follows.

First, we have presented our approach, proposing to describe constraints of electronic documents in OCL and then to translate these constraints into some platform-specific language, for example, XPath [8].

Second, we have proposed to use ontology for description of real-world objects, which details can be transmitted in electronic documents. Based on this ontology, a developer must design conceptual structures of electronic documents, and then he must transform them into structures conforming to one of the studied standards.

Third, we have proposed a free tool [9] to work with ontologies and conceptual structures of electronic documents based on open standards (MOF [14], XMI [17]) and frameworks (EMF [18], Sirius [20]).

We plan to enhance our approach and tool in the following directions.

First, we plan to introduce additional relation types (for example, mereological) and concept types (subjects, objects, events, etc.) into the ontology metamodel.

Second, we plan to refine the tool so that it will allow to generate platform-independent models based on different standards (CCTS, ISO 20022, WCO DM, NIEM, etc.).

Third, we plan to develop several QVT transformations, which will add new entities to an ontology from different reference models (Core Components Library, WCO DM, etc.).

The presented approach can be applied to design of electronic document structures being used in electronic commerce, in G2G or G2B interactions.

References

1. ANSI: Health Level Seven Standard Version 2.6 – An Application Protocol for Electronic Data Exchange in Healthcare Environments. ANSI/HL7 V 2.6, American National Standards Institute (2007)
2. Glushko, R., McGrath, T.: Document Engineering: Analyzing and Designing Documents for Business Informatics & Web Services. MIT Press, Cambridge (2005)
3. ISO: Electronic data interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules. ISO 9735:2002, International Organization for Standardization (2002)
4. ISO: Financial services – Universal financial industry message scheme – Part 1: Metamodel. ISO 20022–1:2003, International Organization for Standardization (2003)
5. ISO: Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 1: Overview and fundamental principles. ISO 15926–1:2004, International Organization for Standardization (2004)
6. ISO: Information technology – Metadata registries – Part 1: Framework. ISO/IEC 11179–1:2004, International Organization for Standardization (2004)
7. ISO: Trade data interchange – Trade data elements directory. ISO 7372:2005, International Organization for Standardization (2005)
8. Nikiforov, D.A.: UML to XML Schema 1.1 Transformation, November 2013. http://dx.doi.org/10.5281/zenodo.16151
9. Nikiforov, D.A.: Conceptual Electronic Document Editor, February 2016. http://dx.doi.org/10.5281/zenodo.46610
10. Nikiforov, D.A., Lisikh, I.G., Sivakov, R.L.: An approach to multi-domain data model development based on the model-driven architecture and ontologies. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Delhibabu, R., Spirin, N., Labunets, V.G. (eds.) Supplementary Proceedings of the 4th International Conference on Analysis of Images, Social Networks and Texts (AIST 202015), Yekaterinburg, Russia, 9–11 April 2015. CEUR Workshop Proceedings, vol. 1452, pp. 106–117. CEUR-WS.org (2015)
11. OMG: Object Constraint Language (OCL), version 2.4. Specification, Object Management Group (2014)
12. OMG: Ontology Definition Metamodel (ODM), version 1.1. Specification, Object Management Group (2014)
13. OMG: Meta Object Facility (MOF) 2.0 Query/View/Transformation, version 1.2. Specification, Object Management Group (2015)
14. OMG: Meta Object Facility (MOF), version 2.5. Specification, Object Management Group (2015)
15. OMG: UML Profile for National Information Exchange Model (NIEM), version 3.0. Specification, Object Management Group (2015)

16. OMG: Unified Modeling Language (UML), version 2.5. Specification, Object Management Group (2015)
17. OMG: XML Metadata Interchange (XMI), version 2.5.1. Specification, Object Management Group (2015)
18. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework 2.0, 2nd edn. Addison-Wesley Professional, Amsterdam (2009)
19. UN/CEFACT: Core Components Technical Specification, version 3.0. Specification, United Nations Centre for Trade Facilitation and Electronic Business (2009)
20. Viyović, V., Maksimović, M., Perisić, B.: Sirius: a rapid development of DSM-graphical editor. In: 2014 18th International Conference on Intelligent Engineering Systems (INES), pp. 233–238, July 2014