A Two-Tiered Database Design Based on Core Components Methodology

Elena Jelisic[1](✉), Nenad Ivezic[2], Boonserm Kulvatunyou[2], Marija Jankovic[3], and Zoran Marjanovic[1]

[1] Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia
{elena.jelisic,zoran.marjanovic}@fon.bg.ac.rs
[2] National Institute of Standards and Technology, Gaithersburg, MD, USA
{nivezic,serm}@nist.gov
[3] Centre for Research and Technology Hellas, Thessaloniki, Greece
jankovicm@iti.gr

Abstract. The number of Industry 4.0, Internet of Things, and cloud service implementations are growing rapidly. In the resulting, emerging, cross-industry, cooperative environments, a common understanding of message standards will be necessary to enable better semantic interoperability among both traditional enterprise applications and business-to-business applications. In this paper, we first discuss issues with the current state of interoperability, which is based on industry sector-based message standards. We then propose a new database design, which incorporates the Core Components Technical Specification (CCTS) methodology, to resolve those issues. Finally, we analyze benefits that come with the new database design and identify new challenges that should be considered through future research.

Keywords: Database design · Business context · Enterprise application · Integration · Global standard

1 Introduction

Evolving fabrication and information technologies are enabling greater cross-industry collaborations. For example, an enterprise specializing in additive manufacturing can provide parts to both aerospace and medical-device sectors. In another example, a vendor of data-analytics can provide those same services to many different business and industry sectors. Today, and for the foreseeable future, such collaborations are, and will continue to be, fueled by the implementations of three concepts: Industry 4.0, Internet of Things, and cloud services. To a large extent, the success of such cross-industry collaboration will depend on solving the semantic interoperability problems associated with those implementations.

Semantic interoperability is not a new problem. It has existed, at least within each vertical, industry sector, for decades. Many existing message standards[1]

[1] Message standards are also referred to as document standards and content standards.

T. Welzer et al. (Eds.): ADBIS 2019, CCIS 1064, pp. 350–361, 2019.
https://doi.org/10.1007/978-3-030-30278-8_36

support semantic interoperability within each industry sector. Those sector-based standards, however, are incompatible. They cannot provide the common reference model needed to achieve cross-industry semantic interoperability. Neither individually nor collectively. Recent attempts to develop a common reference model are based on transforming existing message standards to a global message standard. This transformation has been supported by the Core Component Technical Specification (CCTS) methodology [1]. CCTS is an implementation-neutral standardization method that offers two types of data modeling components – Core Components (CCs) and Business Information Entities (BIEs). Together, these two components can capture both the structure and the contents of information exchange models [2].

Inspired by these transformation attempts, we are intrigued by the opportunity to introduce the CCTS **itself** as the basis for a new database. A database that is designed specifically to facilitate cross-industry, semantic interoperability. The core idea is that if the CCs are stored in a common registry, it would be possible to create a **universal**, conceptual, data model. A data model that could be contextualized easily, giving rise to BIEs, to create logical and physical data models. This idea will be elaborated throughout this paper, which documents a preliminary analysis of both the concept and the benefits of creating such a CCTS-based database. The analysis has been performed from two perspectives: using the database to facilitate cross-industry, semantic interoperability and comparing the database to existing approaches that adopt the existing message standards. In this analysis, we particularly focus on the same complex mapping processes that cause semantic interoperability failures between partners within the same industry sector. In performing that analysis, we made four assumptions (1) all considered business partners are using message standards that have adopted the CCTS methodology; (2) semantic interoperability between these standards is achieved by their transformation into a global standard; (3) all Core Components are stored in a common, open-standards based registry that is accessible to all [3]; and (4) a new information system is considered with CCTS-based database design.

The rest of the paper is organized as follows. Section 2 gives a background information about concepts used in the paper. Section 3 describes the considered integration solutions for new information systems and shows a use case that will be used for the analysis of proposed solutions. Section 4 provides discussion of the presented approach and proposes next research steps. Section 5 gives conclusions of the paper.

2 Background

2.1 Industry Sector Efforts

There have been some attempts to create common, reference, data models that could assure semantic interoperability between business partners within a vertical industry sector. In this paper we will mention only two reference models in the healthcare industry. One of them is the Clinical Information System (CIS), which provides a basis for Electronic Health Record (EHR) systems. CIS is a computer-based system that is designed for collecting, storing, manipulating and communicating available clinical information important to the healthcare delivery process [2]. Another is the Health informatics – HL7 v3 Reference Information Model (RIM) [4]. These standards, even though they are from the same industry sector, are far from being completely harmonized and adopted.

Other attempts to create a common reference model have tried using industry-independent, data-modeling languages. In [5], for example, the authors stated that UML is an industry-independent standard that has been used by Standards Development Organizations (SDOs) as a language to formally represent the semantics. Unfortunately, UML, just like all other standardized data-modeling languages, has not been consistently implemented in any industry where it has been adopted.

We draw two conclusions from these examples. First, semantic-interoperability in a given industry sector is still an unsolved problem. Second, neither industry-specific nor industry-independent data modeling languages cannot be the basis for the common reference model needed for cross-industry semantic interoperability.

2.2 Cross-Industry Efforts

In the past few years, cross-industry efforts have focused on transforming sector-specific message standards into global message standards. There are multiple ways to achieve such a transformation. In this paper, we use the Core Component Technical Specification (CCTS) methodology [1] developed by the UN Centre for Trade Facilitation and Electronic Business (UN/CEFACT). CCTS is an implementation-neutral standardization approach that offers two types of data modeling components – Core Components (CCs) and Business Information Entities (BIEs), which capture the structure and contents of information exchange models [7]. CCs are used for creating conceptual data models, and they are context-free. BIEs are context-specific and they are used for creating logical data models. BIEs restrict the underlying CCs for a specified business context.

UN/CEFACT provides a list of available CCs that can be used for description of exchanged content. This list is called the Core Components Library (CCL). It contains more than 7,000 business entities that can be reused in many scenarios [8]. Example CCs from that list include Document, Contact, Contract, and Person. In our view, the CCL can be the foundation for that missing, common reference model.

Business context (BC) is used to capture the intent of a message. UN/CEFACT defines a business context to be a set of the context values associated with their corresponding context categories [1]. UN/CEFACT provides eight business context categories that can be used for business context description. BIEs are obtained by specifying values or constraints on the values for each of the selected eight business context categories.

SDOs, such as Chemical Industry Data eXchange (CIDX), the Open Applications Group Incorporated (OAGi), Automotive Industry Action Group (AIAG), Universal Business Language (UBL), and RosettaNet, have already incorporated CCTS methodology or are in the process of adopting the CCTS methodology and standards stack [3]. This transition means that, for example, all components of the Open Application Group Integration Specification (OAGIS) standard are mapped to CCs [6].

From these examples we came to the following conclusions. First, global standard can provide cross-industry semantic interoperability, but this holds only for those message standards that have adopted a common data model, like CCTS, in its core. Second, global standard cannot solve all problems in the traditional approach, like mapping problems, which we discuss below.

3 A Foundation for CCTS-Based Database Design

Inspired by transformations of message standards to global standards, our paper presents a new approach in database design that would incorporate multiple promising techniques towards achieving overall semantic interoperability. First, this section describes a simple use case that will be used as a basis for analysis of a new database design. Then, by analyzing integration requirements, we distinguish two alternatives for achieving semantic interoperability (1) a specific message standard selection, and (2) adoption of CCTS methodology in database design. We consider both alternatives, and then focus our discussion on the latter one. Accordingly, we propose a new foundation for the considered alternative realization.

In performing that analysis, we will use three of the eight categories for BC definition: Geo-political, Activity (Business process) and Industry. When BC is applied on some of the provided CCs, it gives it a necessary semantics in a specific integration scenario. For example, if BC is defined as *Invoicing business process in chemical industry in Serbia*, and is applied on the *Document CC*, we can say that this abstract, implementation-neutral concept *Document* in this specific BC represents an Invoice business document for that business context of the *invoice processing in the chemical industry in Serbia*.

3.1 A Use Case

Company A, a logistics enterprise, wants to provide logistics services to two different kinds of enterprises that currently operate in different European countries, called B and C. Enterprise B is a chemical company that has adopted the CiDX messages standards for their business processes. Enterprise C is an automotive manufacturing company that has adopted the OAGIS message standards for theirs. A System Context Diagram (SCD) that captures the business processes that define the relationships between Company A and Companies B and C is presented in Fig. 1. The rectangles represent the two business partners, B and C. Each business partner is described by its name and the industry sector to which it belongs (in brackets). The oval represents Company A that has its own business process, which differs from both B's and C's business processes.

The directed arcs in the diagram present messages exchanged between Company A and its business partners. Each message is labeled; there are two output *Invoice* messages from Company A; and two, input *PurchaseOrder* messages to Company A. This means that Company A must have two sub-processes: *Receiving a Purchase Order and Sending an Invoice.* Businesses B and C will have similar sub-processes for *Sending a Purchase Order* and *Receiving an Invoice.* Our case study focuses on the *Sending an Invoice* sub-process in Company A.

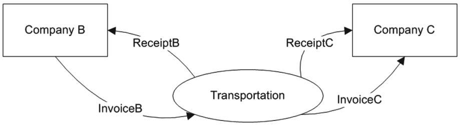

Fig. 1. Company A - Transportation business process SCD.

The problem with implementing this simple diagram is that each company represents *Invoices* (and *PurchaseOrders*) in completely different modeling languages. Company A creates an *Invoice* business document, after any transportation activity is successfully completed, to charge for its services. The structure of the message containing that *Invoice* should be compliant with, and easily converted into, the two semantic interoperability standards used by the business partners: CiDX and OAGIS. We investigated two alternatives for achieving the required cross-industry, semantic interoperability.

1. Company A should adopt one of the two vertical standards either CiDX or OAGIS.
2. Company A should adopt neither vertical standard. Instead, it should incorporate our proposed CCTS-based database.

3.2 Alternative 1: Select One Vertical Standard

This selection means that the existing database design and data modeling language used by Company A to create the structure of an *Invoice* message remain the same. The structure and content of this message do not have to necessarily be compatible with the structure and content of either CiDX or OAGIS. Nevertheless, both CiDX and OAGIS have already incorporated CCTS into their core. This means that, theoretically at least, these industry-specific standards have the basis for a transformation into a single, global standard. If successful, this global standard would further imply that any message created in one of the two standards should be **correctly interpretable** in the other one. Defining the necessary real-world mappings needed for "correctly interpretable" still requires additional, and sometimes difficult, work. So, generally speaking, if Company A were to select one of these vertical standards to represent its *Invoice* message, semantic interoperability should be achievable at a much-reduced cost and risk.

The constraint of this approach, of course, is the difficulty in defining those mappings. This undertaking brings in many issues that can have a negative influence on interoperability. In other words, for each local concept in the local data model, an appropriate counterpart should be found in the chosen message standard. Otherwise, failure to find this match could cause semantic interoperability problems between partners. Also, local data might be lost if there is no adequate counterpart in the message standard structure. These issues can be addressed by adopting a new, database design based on CCTS, as described next.

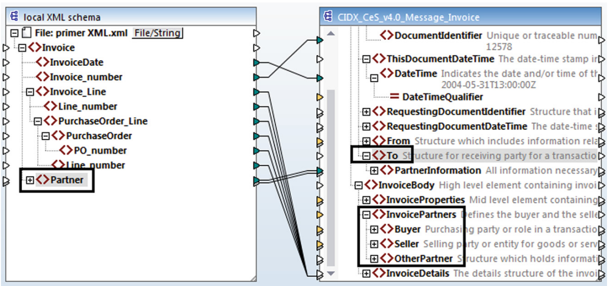

Fig. 2. Partner details mapping.

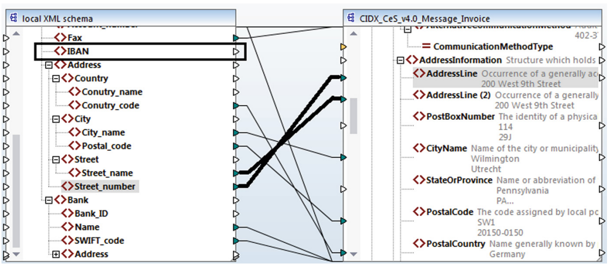

Fig. 3. Address and IBAN mapping.

Figure 2 presents the first constraint of this approach - finding an appropriate counterpart. XML Schema that presents the structure of *Invoice* business document from the local database is on the left side, and on the right side is *CIDX Invoice message schema*. This figure shows that Partner details can be mapped to multiple elements in *CIDX Invoice message schema - To, Buyer, Seller,* and *OtherPartner*. Which one is going to be used depends on user preferences. In addition, this mapping can be implemented differently each time. In other words, there is no consistency in mapping. Figure 3 presents two elements from a local XML schema (Street and IBAN) that do not have adequate counterpart. Actually, *Street name* and *Street number* have the only one fitting element in *CiDX Invoice message schema*, and that is the *AddressLine*. By default, there is only one *AddressLine* element, but in order to map both street details from local schema we had to duplicate *AddressLine* element, otherwise some data from local schema would be lost. The other element that does not have an appropriate counterpart is IBAN. This figure depicts the second issue when local data might be lost if there is no adequate counterpart in the message standard structure.

3.3 CCTS-Based Database Design

The second alternative that Company A considers is a new database design that would make use of the emerging CC environment. This environment (1) provides a collection of CCs and BIEs stored in a common registry accessible to all and (2) enables the creation of a universal, conceptual, data model [3]. This paper proposes a foundation, based on this environment, for designing such a database. That foundation has two tiers: a conceptual, data-model tier and a business-context (BC) tier.

The First Tier - Integral Invoicing Data Model
To develop the integral data model that represents the concept *Invoice*, we propose a three-step process.

1. Examine the CiDX data model of the concept *Invoice*
2. Examine the OAGIS data model of the concept *Invoice*
3. Create a new data model using CCs

Step 1. By analyzing schemas provided by Chem eStandards 4.0 [9] we have decided to use CIDX_CeS_v4.0_Message_Invoice schema. Figure 4 shows the structure of Invoice message presented using UML class diagrams. Those diagrams contain a CiDX-specific, CC list of entities. Those entities describe the various properties and their relationships associated with the concept *Invoice*. Those properties include Invoice Number, Ship Date, Language Code, Issue Date, and Invoice Date, to name a few. The resulting model gives an *Invoice* message structure that is defined inside CiDX Chem eStandards. In other words, this is Invoice structure that is specific for chemical industry. As we stated in Sect. 2, each of the entities presented in Fig. 4 has some predefined CC as its basis.

Step 2. By analyzing schemas provided by OAGIS 10.5 Enterprise Edition [10] we have decided to use the GetInvoice schema. Figure 5 presents the structure of this schema using UML class diagram. This diagram presents OAGIS - specific Invoice message structure. An Invoice is described using Invoice Line and Invoice Header. Invoice Header holds general data, like Tax, Total Amount, details about Supplier, Customer, Billing Party and so on. Invoice Line presents Item-specific data - Quantity, Unit Price, Amount Discount to name a few. As before, each of the entities presented in Fig. 5 has some predefined CC in its basis.

Step 3. The previously described data models, their concepts, and their underlying CCs[2], provide the basis for creating a CCTS-based, conceptual, data model.

[2] The list of all CCs is available in [8].

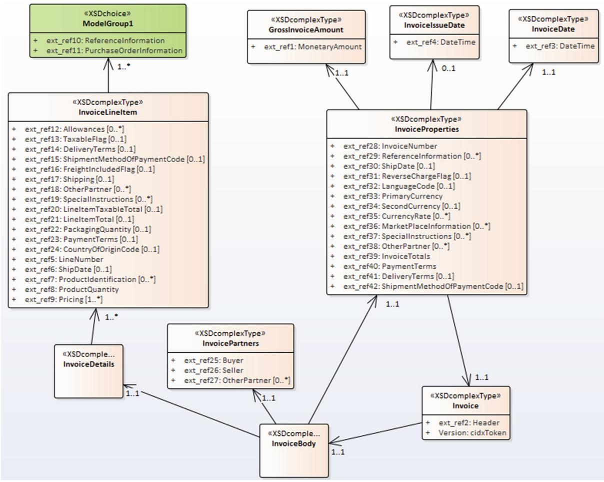

Fig. 4. CiDX Chem eStandards Invoice message UML class diagram.

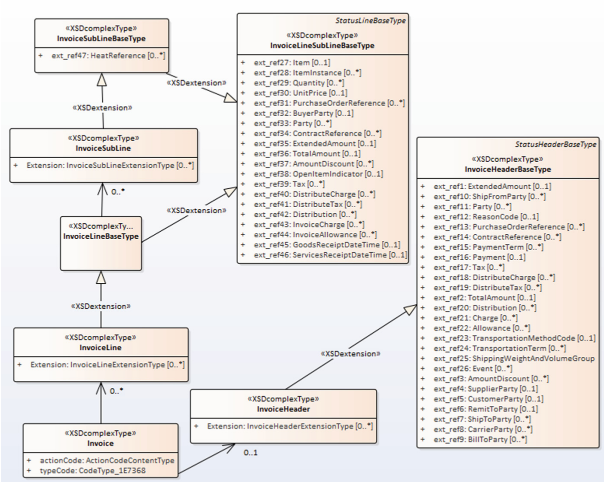

Fig. 5. OAGIS Invoice message UML class diagram.

By manually analyzing the entire list of CCs, we have concluded that the two individual *Invoice* structures can be interpreted using (1) *Document* Aggregate Core Components (ACC) that has a list of Basic Core Components (BCCs) and (2) Association Core Components (ASCCs) used for association with other ACCs. For easier reading, we will call the BCCs as "fields", the ACCs as "components", and ASCCs will be referred to as "associations". Since the resulting CCTS-based document is supposed to support the structures of all, Invoicerelated, business document types, its data model is quite complex. Thus, this paper presents only its representative parts without any "fields" specifications.

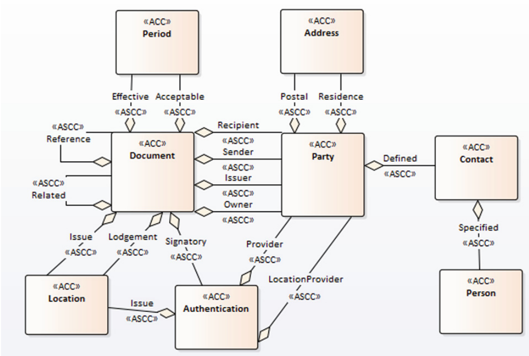

Fig. 6. Document ACC - Conceptual data model.

The resulting conceptual data model is presented using UML class diagrams in Fig. 6. In this model we can see that *Document* has associations with several components including *Party, Period, Location* and *Authentication*. With each of these components, multiple associations can be created. For example, for *Document* we can identify details about its sender party, its recipient, its owner and its issuer. Further, *Party* has details about its *Address* and *Contact Person*. So, using the conceptual data model presented in Fig. 6, complete CiDX or OAGIS Invoice message structures can be interpreted. Finally, this tier presents the database structure that would be implemented by Company A. Using the conceptual data model presented in Fig. 6, Company A will create its own physical data model, specific for a selected Database Management System (DBMS). The main idea is that there will be database tables as presented in conceptual data model and they will be used to store any type of business document used by Company A. In our case, those documents include the *PurchaseOrder* and the *Invoice*.

The Second Tier - Business Context (BC) Definitions

The second tier involves two functions. First, it provides information about all BCs in which the target enterprise operates. Second, it holds definitions for the structure of each business document type. These definitions will be stored in a BC repository that can be implemented inside the individual database schemas.

In Fig. 7, a data model that supports business context definitions is presented as a UML class diagram. In this model, we can see that each BC is described by a set of BC categories. Each BC category has its list of available values. There is also an association class named List of Values that further has an association with BC Category Value. This association is used to denote which BC Category's specific value is applied in a specific BC. Through List of Values association class values for each BC Category are defined to describe some BC.

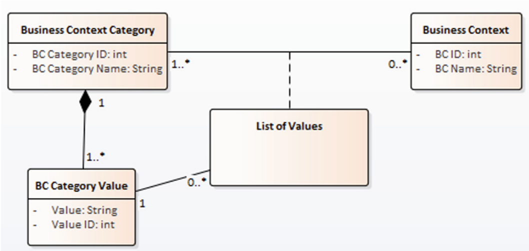

Fig. 7. Data model for Business context tier.

As we stated, the second tier will also hold definitions for each business document type. This function will be implemented through a list of SQL views for each identified business-document type. For example, an SQL view for Invoice would select a list of fields and components (from conceptual data model in Fig. 4) that are needed to describe this type of message. In addition, the same SQL view will reference the BC in which it is supposed to be used. This resulting SQL view will use concepts introduced in the conceptual data model (*Document, Party, Location* etc.) and their names will not be message standard-specific. The assumption is that a business partner will be able to interpret any message defined in this way since it has adopted CCTS-based message standard.

For our use case, we have defined BC_B and BC_C, that denote business environment in which Company A cooperates with its business partners (Companies B and C respectively). These BCs are presented in Table 1. Since our business partners are from Europe, the relevant BC value is defined in the list of values for the Geo-political BC category. Other category values could be countries from all around the world. Using International Standard Industrial Classification of All Economic Activities (ISIC) [11] we have defined the list of values for the Industry BC category. For these BC categories we have assigned values of chemical and automotive, since these are the ones in which Company A's business partners operate. The list of values for the Activity BC category is defined by the business processes of Company A. In our case, this category has two values: *Receiving a Purchase Order* and *Sending an Invoice.*

In Fig. 8 an overall architecture for proposed approach in database design is presented.

Table 1. Business contexts B and C.

BC category	Business contexts	
	BC_B	BC_C
Geo-political	Europe	Europe
Activity	Receiving a Purchase Order	Sending an Invoice
Industry	Chemical industry	Automotive industry

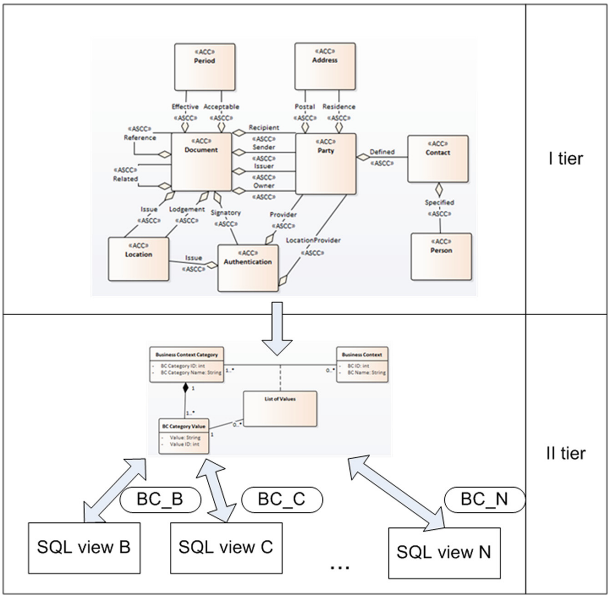

Fig. 8. Database design - an overall architecture.

4 Discussion and Next Steps

In this paper we have seen possible alternatives for achieving cross-industry semantic interoperability. One presented alternative is to adopt a message standard that some of business partners have already incorporated in their business. In this case, local data models need to be mapped to a message structure of the chosen message standard. The paper named some issues that arise in this mapping process. These issues have negative effects on achieving cross-industry, semantic interoperability.

We have also presented another alternative that does not require choosing any specific message standard. In this approach, semantic interoperability is achieved through an integral, conceptual, data model that is based on CCTS. The result is that collaboration can be achieved with any message standard that has adopted CCTS at its core. In addition, a CCTS-based design bypasses mapping process, thus eliminating the identified issues.

This paper also opens three questions that need to be addressed through future research. First, the presented solution is applicable only to information systems being designed from scratch. The future work will consider applying this approach to existing information systems. Second, currently, the unifying conceptual data model can only be created manually and, thus, is error prone. Further research needs to focus on the possibility of automated conceptual data model creation. Third, we can see that the new approach does not provide information about the message standard that is adopted by a business partner. Thus, future research may consider including one additional BC category that would denote message standard that the business partner is using.

5 Conclusion

In this paper, we analyzed alternative integration solutions that are available for new information systems. Each solution is considered in turn and potential problems were identified for each. In particular, this paper presented a foundation for a CCTS-based environment that could support newly proposed approach to information systems database designs. This foundation could contribute to significant improvements in cross-industry semantic interoperability. The paper identifies also a new collection of research questions that need to be addressed to realize CCTS-based database designs.

Disclaimer

Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.

References

1. UN/CEFACT Core Components Technical Specification CCTS, version 3.0. https://www.unece.org. Accessed 10 May 2019
2. Clinical Information System. Biohealthmatics. http://www.biohealthmatics.com/. Accessed 12 June 2019
3. CiDX Core Component White Peper. Open Applications Group Inc. https://oagi.org/. Accessed 14 May 2019
4. HL7 Reference Information Model. HL7 International. http://www.hl7.org/. Accessed 12 June 2019
5. Lopez, D.M., Blobel, B.G.M.E.: Enhanced semantic interoperability by profiling health informatics standards. Methods Inf. Med. **2**(48), 170–177 (2009)
6. Context-Driven Message Profiling. https://oagi.org/. Accessed 10 June 2019
7. Novakovic, D.: Business Context Aware Core Components Modeling, 2nd edn. Publikationsdatenbank der Technischen Universität Wien (2014). https://publik.tuwien.ac.at/
8. UN CCL. The United Nations Economic Commission for Europe. http://tfig.unece.org/. Accessed 10 May 2019
9. Chem eStandards 4.0. Open Applications Group Inc. https://oagi.org/. Accessed 15 May 2019
10. OAGIS 10.5 Enterprise Edition documentation. https://oagi.org/. Accessed 15 May 2019
11. International Standard Industrial Classification of All Economic Activities (ISIC), Rev. 4. https://oagi.org/10.18356/8722852c-en