

Model-based Analysis of HL7 CDA R2 Conformance and Requirements Coverage

Abderrazek Boufahja¹, Eric Poiseau², Guillaume Thomazon¹, Anne-Gaëlle Bergé¹

¹Gazelle Team, Kereval / IHE-Europe, Rennes, France

²Gazelle Team, Inria / IHE-Europe, Rennes, France

Abstract

Numerous national and regional projects around the world [23] are developing specifications for sharing electronic medical records. Many of them are basing their specifications on the HL7 CDA standard, extending it in order to meet the local requirements or medical practice. Many of these projects are illustrating the specifications with sample CDA documents and provide in addition tools [13] [14] [24] to check the conformance of CDA documents with their extensions. In this paper we provide the outcome of an evaluation of both the samples and the tools provided by these projects.

We looked at the conformance of the provided samples with the basic HL7 CDA requirements as specified within the "Clinical Document Architecture, R2 Normative Edition", and we looked at the capability of the tools provided to check those requirements. The outcome of the study shows that a large portion of the requirements specified by the standard are neither tested nor respected by the provided validation tools and samples.

Keywords

CDA, conformance, coverage, requirements, model-based, validation

Correspondence to:

Abderrazek Boufahja

Kereval

Address: 4, rue Hélène Boucher

Z.A. Bellevue, 35235 Thorigné Fouillard

E-mail: abderrazek.boufahja@ihe-europe.net

EJBI 2015; 11(2):en41–en50

received: October 31, 2014

accepted: January 6, 2015

published: January 20, 2015

1 Introduction

Since the publication of the HL7 CDA R2 specifications [1], conformance checking of CDA documents has been a source of inspiration for multiple companies expert on healthcare IT standards. Since the CDA schema does not cover all the basic CDA requirements, several methodologies were developed and experimented in order to validate the content and the structure of the CDA documents [5] [14] [15]. The purpose of this paper is double. In a first part we have revisited the content of the CDA specifications in order to clearly identify and extract the requirements that are specified in the CDA standard but not covered by the CDA schema. Then, we evaluated the conformance of CDA documents published by various projects in Europe, with the extracted basic CDA requirements. And finally we checked the coverage of these basic requirements by different CDA validation tools.

2 State Of The Art

2.1 Introduction

The activity diagram described in Figure 1 summarizes the process used by most of the CDA validation tools.

The validation steps are as follows:

- The first step of the activity diagram checks that the CDA document is a well-formed XML Document [20]. This step checks that the syntax of the document is correct: the root element is present, all elements have a closing tag, elements are properly nested, attributes values are correctly quoted, etc.
- The second step is used to verify that the document is valid against the CDA schema [21].
- The third and final step is commonly used to verify the business rules related to the CDA standard and which are not expressed in the CDA schema.

Numerous tools are used to perform the last step of this validation process; this section describes the most used

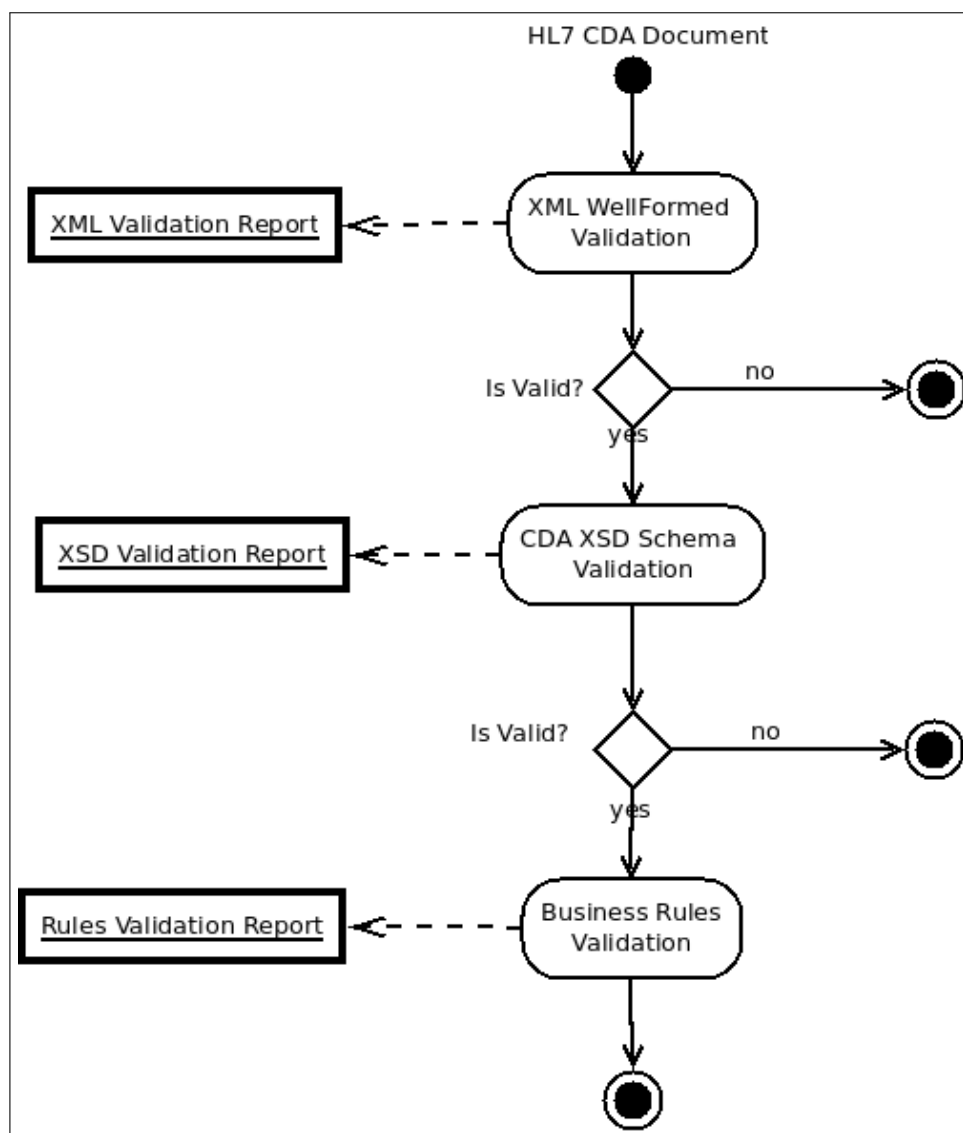


Figure 1: Validation process of CDA documents.

ones. Also, many studies were done to describe basic requirements in CDA standard; this section provides these analyses.

2.2 HL7 CDA R2 Validation Tools

This paragraph describes the most commonly used CDA validation tools.

Trifolia Workbench: Trifolia Workbench is developed by Lantana Consulting Group. It is a web-based application for standard development work, and it supports the generation of schematrons in order to test the conformance of CDA documents, based on requirements written within the tool [13].

MDHT: Model-Driven Health Tools (MDHT) is an open source tool developed and maintained by Open Health Tools[14]; the purpose of which is to validate CDA documents. It provides also a validation of basic CDA documents.

Eclipse Instance Editor (EIE): Eclipse Instance Editor is a tool used to create and edit CDA documents, and also to validate according basic CDA requirements. As described in [19], this tool is based on MIF and R-MIM description.

Art-Decor: Art-Decor is a tool developed by the ART-DECOR expert group [15]. It is a web-based application to record HL7 templates and reusable artifacts as value sets and templates. This tool allows the definition and generation of specification documentations. It also allows the generation of schematrons for checking the conformance to these specifications. This tool is largely used in Europe (ELGA from Austria, HL7 Deutschland, HL7 Norway, etc).

NIST validation tool: The NIST validation tool is released by the National Institute of Standards and Technology (NIST), an agency of the U.S. Department of Commerce, USA. This tool is a web-based application allowing the validation of CDA documents based on schematrons

developed by the NIST for the validation HL7 Continuity of Care Document (CCD) and IHE Patient Care coordination domain. The tool offers a web service to validate CDA documents. It is largely used within the IHE community.

Gazelle ObjectsChecker tool: The Gazelle ObjectsChecker tool is developed and maintained by IHE-Europe and it is part of the Gazelle project [11]. The tool provides a model-based validation of CDA documents, based on UML description of CDA requirements [5], including the CDA basic validation described in this paper.

Validation Tools Properties

The Table 1 summarizes the properties of each validation tool. From this table, we can remark that there are two kinds of tools: tools oriented to specifications generation, and tools oriented to requirements validation. The tools oriented to specifications generation are MDHT, Tri-fovia and Art-Decor.

2.3 National and Regional Validation Materials

Many regional and national healthcare CDA-based specifications provide material to validate the conformity of CDA documents according to the national constraints; ASIP Santé (France) provides schematrons [25], eHealth Swiss provides an online validation tool based on schematrons [26], ELGA (Austria) provides an online validation tool and schematron resources used within the validation, KELA (Finland) provides online validation tools based on schematrons and the EVSClient tool[24]. Also, many US implementations provide schematrons to validate the conformity of CDA documents.

Many of these validation materials claim to be able to verify the conformity of the CDA documents according to their national or regional specifications; however multiple requirements related to the CDA standard could not be verified by these tools (see the requirement coverage paragraph for more information), and so these implementation materials may validate the requirements related to their specifications, but ignore an important number of basic requirements related to the CDA standard. This problem is due to the fact that there is no formal description of all the CDA basic requirements that the validation tools could refer to.

2.4 HL7 CDA R2 Requirement Studies

Previous studies were performed in order to describe the common requirements of CDA standard, which are not verified by the CDA schema [17] [18]. An important study is the one referenced [17], a white paper written by Rene Spronk and Grahame Grieve about the common issues found in the implementations of CDA, a description of the most common recurrent errors produced by imple-

menters. This white paper was an important reference for this one, even if the subjects of those two papers are not the same.

3 CDA Basic Requirement Specifications

3.1 Introduction

The HL7 CDA R2 standard is based on:

- four specification documents [1] [2] [3] [4]
- a list of HL7 value sets
- the hierarchical descriptor of CDA standard
- the CDA Schema

All CDA documents SHALL conform to these specifications. Most of the requirements specified in the standard are expressed in the XSD schema, but not all of them. The paragraph below summarizes the requirements that are not expressed within the schema.

The complete list of the requirements missing from the schema was extracted and compiled by our team and published in the document ‘HL7 CDA R2 Basic Requirements’ [12]. The latter contains the interpretation of the requirements expressed in the CDA standard, which are formally listed and uniquely identified. More than 150 requirements not expressed in the schema were extracted, and the following sections describe these requirements.

3.2 HL7 CDA R2 RIM Requirements

Multiple requirements coming from the RIM model are not covered by the CDA schema, and are expressed in [2]. These requirements are especially related to the use of CNE value set (as it is the case for statusCode elements), and the use of the SET<T> data types.

3.3 HL7 CDA R2 R-MIM Requirements

There are multiple requirements missing in the CDA schema, and expressed in [1]. These missing requirements are extracted in [12], and they are mostly related to the following kinds of requirements:

- Requirements related to the use of the SET<T> data types (see paragraph 3.1, [3])
 - the SET<T> shall not contain null elements, when there are other elements which are not null
 - the SET<T> shall not contain equal elements
- Requirements related to the use of CNE value set on coded data type elements (as it is the case for statusCode, languageCode and interpretationCode elements)

Table 1: CDA validation tools properties.

	MDHT	Trifolia	Art-Decor	Gazelle Objects Checker	EIE	Nist Validation
Open source	yes	no	yes	yes	yes	yes
Rules editions GUI	yes	yes	yes	no	n/a	no
Generation of schematrons	no	yes	yes	no	n/a	n/a
Code generation	yes	yes	yes	yes	yes	n/a
Binding to value set repository	no	yes*	yes*	yes	yes*	yes*
Online validation service	yes	yes	unknown	yes	no	yes
Specifications generation	yes	yes	yes	no	no	no
Specifications output format	PDF DITA XHTML	Word XML HTML	HTML docbook XML	no	n/a	n/a

* The use of value sets is not from a repository but from file(s) containing all the value sets

- Requirements related to the behavior of the CDA elements, like the relationship between the narrative text and the statements, etc.

3.4 HL7 CDA R2 Data Type Requirements

The data type requirements in CDA standard which are not expressed in the CDA schema come from documents [3] and [4]. Multiple kinds of requirements are missing in the schema:

- requirements related to the use of nullFlavor with data types
- requirements related to the structure of the data type, like the structure of the telecom data, the UUID, the email, etc.
- requirements related to the use of fixed value set, like the use of UCUM for units.
- requirements related to the use of intervals; like for example the IVL<TS>, where the low value SHALL be lower or equal to the high value
- requirements related to the XML implementation: come especially from the data type ITS specification; like requirements related to the XML header definition

3.5 HL7 CDA R2 Narrative Block Requirements

Some requirements related to the narrative block description were expressed in the specification [1]. These requirements are especially related to the relationship between the narrative text and the coded elements, like the IDs references.

3.6 CDA Requirement Types

All the requirements missing in the CDA schema have been divided into two types, inspired from the RFC 2119 [22]. The ones that express an absolute requirement or prohibition belong to type 1, not respecting them will result in an error. The requirements that express a recommendation belong to type 2 and will raise a warning when not respected.

- Type1: the requirement is strong, and if the CDA document fails to implement it, then the document shall not be considered as a valid CDA document. The outcome of the validation raises an ERROR
- Type2: the requirement is not as strong as those of type1, and if the CDA document fails to implement it, then the document is still a valid CDA document. The outcome of validation against such requirement is a WARNING.

4 HL7 CDA R2 Conformance Analysis

To verify the conformance of the CDA documents, validation tools execute a list of checks in addition to the validation against the CDA schema. The outcome is a Boolean: the document does or does not conform to the CDA standard, with a list of positive and negative checks. From our point of view, the conformance of a CDA document could not be only described by a Boolean value, but with an indicator which states how conformable a CDA document is. As a consequence, we define the indicator of CDA validity.

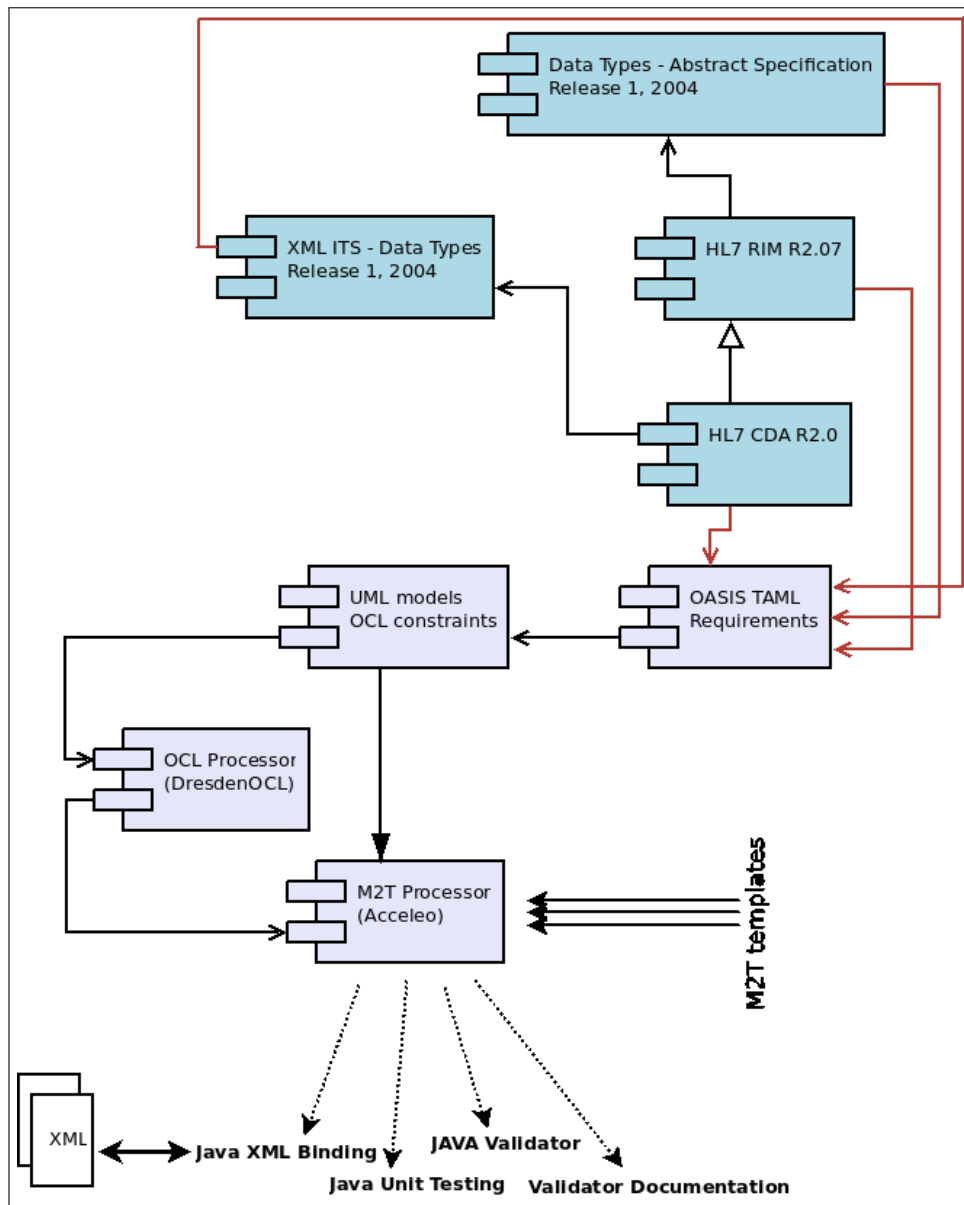


Figure 2: Principle of Gazelle ObjectsChecker.

4.1 CDA Validity

The CDA validity responds to the question: is the document valid according to the CDA standard? The indicator that describes the CDA document validity (I_v) is the number of errors found when checking the document against the CDA requirements.

$$I_v = N_{err} \tag{1}$$

Where N_{err} : Number of errors found

Another indicator, which is more significant from tool perspective, is the absolute indicator of validity (I_v^a), which describes only the number of different kinds of error found, and not the total number of errors found in the CDA document. Consequently, we remove the duplicated errors.

$$I_v^a = N_{err}^a \tag{2}$$

Where N_{err}^a : Number of kinds of error found

In addition to describing the document validity, this indicator mostly describes the degree of validity of the tool that has generated the CDA document.

5 Gazelle ObjectsChecker Methodology

Within the Gazelle Test Bed [11], we have developed a methodology for the conformance checking of XML content [5]. Figure 2 presents the principles of the method which is based on the UML description of the XML structure of the document, in our case the structure of CDA

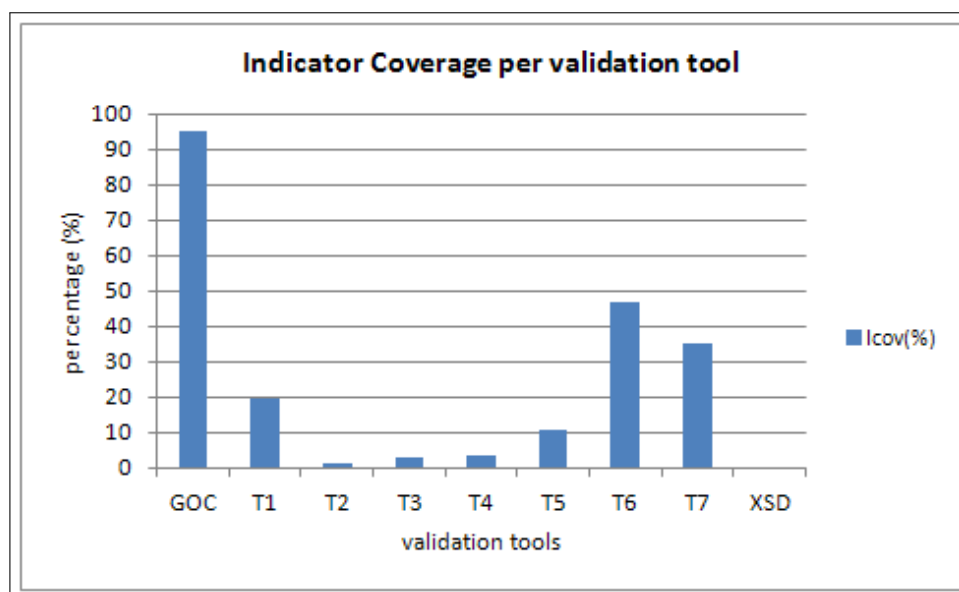


Figure 3: CDA basic requirements coverage indicator by CDA validation tool.

documents, and the description of the requirements based on the OCL language, which allows expressing formal requirements between UML classes and elements [6]. Once the model and the constraints populated, it is processed using a model to text (M2T) processor (acceleo)[7] and an OCL Processor (DresdenOCL) [9]. The outcome of the processing results in:

- First, Java code is generated for editing and modifying the CDA documents. This Java code allows the binding of Java instances to XML elements using JAXB API [8].
- The second output is a set of Java classes for validating CDA documents. This Java classes contain the OCL constraints processed and transformed into Java using the OCL processor.
- The third output is a set of HTML pages which documents the requirements written into the UML model.
- The last output is a set of unitary tests written in Java and based on the OCL constraints.

To manage requirements in a formal structure, we use OASIS TAML standard [10]. This technology allows to uniquely identifying each requirement from the specifications, which allows correlating each OCL constraint with a set of requirements.

For basic CDA validation, the OASIS TAML requirement structure is fulfilled using the requirements extracted from the normative description of CDA R2, RIM V2.07, and data types specifications, and described in [12].

All the UML requirements are hand written directly into the UML models using the OCL language and are tested and verified using unit tests.

The Java code generated using the Gazelle Objects Checker tool can be used for edition and for validation of CDA documents. In the same way, the generation and processing of unit tests result in a database of samples. This set of CDA documents is used for testing the requirements coverage of other CDA validation tools.

6 CDA Basic Requirements Coverage Analysis

6.1 Methodology

Requirement Coverage Indicator Specification

The requirement coverage is an indicator for validation tools. It describes the percentage of requirements covered and tested by the tool, compared to the total number of requirements.

$$I_{cov} = \frac{N_{cr}}{N_{tr}} \quad (3)$$

Where N_{cr} is the number of covered requirements, and N_{tr} is the total number of requirements related to the standard, in occurrence the CDA basic requirements, extracted and described in section 3.

This indicator provides information about the strength of the validation tool. When a document is validated against a tool with a low I_{cov} , the validation result is not relevant and the reliability of the tool decreases.

The indicator of validity I_v describes the validity of a document regarding a validation tool, and the indicator of coverage describes the validation tool itself; it is a constant value related to the tool and does not change per document validated. There are no direct link between the I_v and the I_{cov} except the fact that the couple (I_v, I_{cov})

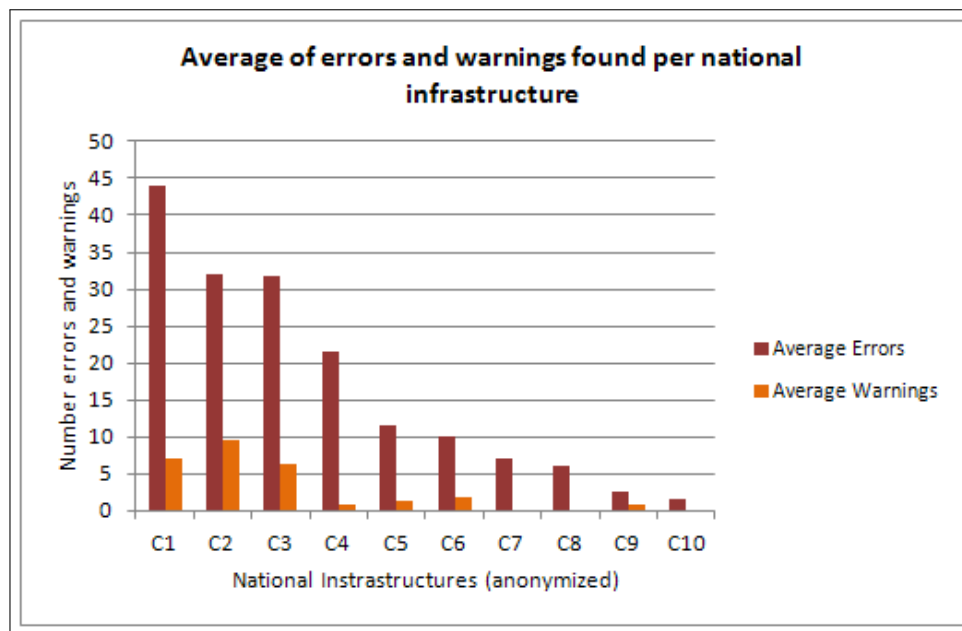


Figure 4: Validation of national specifications samples from Europe and North America.

describes the consistency of the CDA document regarding the basic CDA requirements.

Validation Tools: Requirements Coverage Indicator Calculation

To compute the requirements coverage of the different CDA validation tools, we developed a set of unit tests for each requirement. Each unit test consists of at least two CDA documents: one document that complies with the tested requirement (result will be ‘passed’, OK test), and one that does not (result will be ‘failed’, KO test); when the validation tool fails finding an error in a KO test (‘passed’ is returned whereas ‘failed’ was expected), we assume that the tool does not cover the requirement.

6.2 Results

More than 600 CDA were created to test the requirements expressed in [12].

Based on this methodology, we succeeded in generating a requirement coverage diagram (Figure 3). This diagram contains the requirement coverage for 7 tools, in addition to Gazelle ObjectsChecker validation service. The tools tested are especially those which provide basic validation of CDA documents, which are MDHT basic CDA validation, Eclipse Instance Editor, NIST web service validation for basic CDA documents, and some schematrons which are ART-DECOR basic validation, ASIP santé schematron for common validation, XSD-SD schematron and XD-LAB schematron from IHE. The weakness of this methodology is the fact that we are not able to apply it on schematrons based on templates, without modifying the original schematrons, and so the I_{cov} for the schematrons tested indicates the number of requirements detected and

covered by the tool, but it does not indicate the number of requirements uncovered.

The purpose of this paper is not to compare the tools so we anonymized the results of the study.

None of the tools does cover 100% of the basic requirements and apart from Gazelle ObjectsChecker, all are ignoring more than 50% of the requirements that are not expressed within the CDA Schema.

7 Implementation

7.1 Application 1: National Infrastructures Samples Studies

Most of the national and regional infrastructures which restrict the use of the CDA standard to their national and regional use cases provide samples documents. Their purpose is to help the implementers of IT systems with creating and/or parsing the CDA documents. Multiple European and North American national healthcare IT do it, and most of them provide tools to validate the CDA documents according to the national specifications (some of them are described in the state of the art section). The samples provided with the national healthcare IT specifications are generally valid against the provided tools. As an application of the CDA basic validation using the Gazelle ObjectsChecker methodology, this paragraph studies the conformance to the CDA standard of the samples provided by some European and North American national entities; they are: ASIP Santé (France)[25], ELGA (Austria)[27], Kanta (Finland), HL7 Germany, HL7 Switzerland, region Emilia-Romagna (Italia), NICTIZ (Netherlands), e-MS (Canada), HL7 Czech Republic[28], and some USA CDA imple-

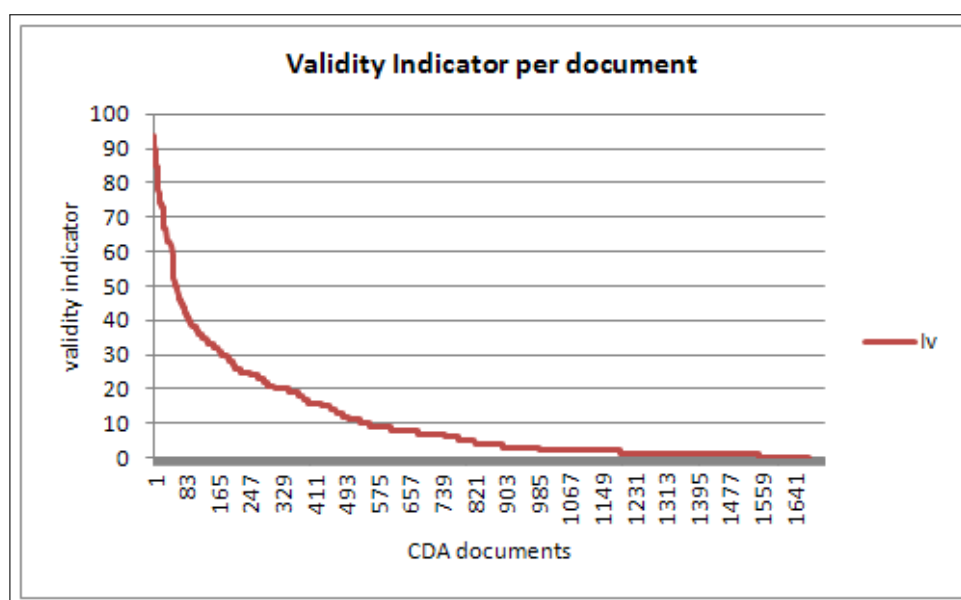


Figure 5: Basic CDA validation of already valid CDA documents according to IHE schematrons.

mentation specifications: CCD, C-CDA R1.1, CRS, and HIPAA. And, for security issue, we anonymized the results of validation, in Figure 4.

153 documents were used to generate this graphic. The average of errors indicator used in this graph describes the average of the validity indicator I_v per document validated for each national infrastructure, and the average of warnings indicator describes the average of the number of warnings found per document. We can conclude from this diagram that a huge number of samples provided by national specifications are not conform to the basic specification of the CDA standard. We found an average of 14.97 errors per CDA document, which is an alarming number as it means that 2200 errors were found in official national and regional samples. The difference of the average of errors between the different national infrastructures may be explained by the difference of the level of complexity between their standards, and also could be explained by the quality of the national tool used to validate these samples.

7.2 Application 2: IHE Schematrons Validation Studies

The Figure 5 describes the number of errors (I_v) found using Gazelle ObjectsChecker for the conformance checking of basic CDA requirements, in 1700 CDA documents, which are valid according to the IHE schematrons. The samples were sorted by the number of errors found.

The first remark that we can do is about the number of errors found, which varies from 0 to 100, with 50 different kinds of error not detected. We conclude that only a few basic CDA requirements are validated by the schematrons. The average of errors found is 11.08 per document, which proves how important is the validation of basic CDA requirements.

By executing the basic CDA validation on 20,000 CDA documents coming from multiples sources (especially epSOS[16] and IHE), we found errors for only 60 out of the 160 requirements identified. This could be related to the fact that multiple requirements are rarely encountered; and this could explain why multiple validation tools do not check more than 50% of the requirements.

7.3 Application 3: Most Frequent Errors in CDA Documents

Based on the analysis done on the national CDA samples, and the IHE valid samples, we extracted the most frequent errors found in the CDA documents. The Figure 6 describes the frequency of errors found. We classified all the errors found in the validation of the documents by requirements, and the indicator used describes the percentage of errors found related to each requirement, regarding the total number of errors found.

50% of errors found are related to only five requirements, and 85% of errors found are related to only 10 requirements. These requirements are:

1. DTITS-007: the use of reference element under an ED data type (27% of errors found)
2. CDADT008 /CDADT-006: the use of attributes related to CD data type (nullFlavor, code, displayName, etc)
3. CDADT-011: the use of UUID structure
4. RMIM-078: the use of scope and player elements under a CDA role element
5. CDADT-013/CDADT-014: the specification of URL references

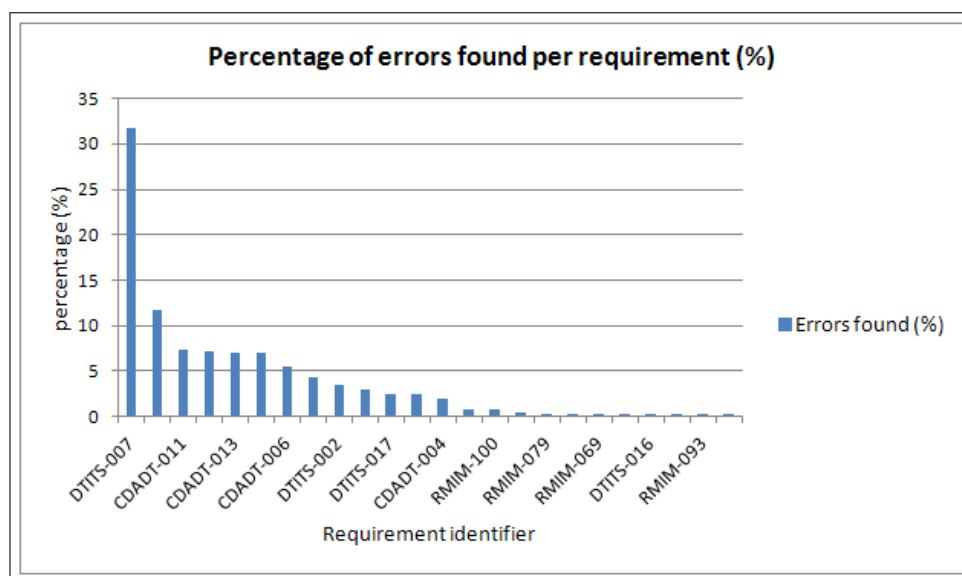


Figure 6: Most frequent CDA basic errors.

6. DTITS-017: bad declaration of a timestamp under TS data type

Other kinds of errors were also largely detected in the CDA documents, like the telecom structure, the use of schemaLocation attribute, and the misuse of CNE codes, like in Observation.interpretationCode element.

The errors found describe an inconsistency between the documents validated and the HL7 CDA R2 standard. However, these errors do not always deteriorate the clinical information included in the documents. Some of the errors found can create problems of interpretation of the clinical information, like the errors in the structure of the timestamp, or the misuse of codes from CNE value sets; on the other hand, a big number of the errors found does not create inconsistencies in the clinical information, but may create troubles for the software that will treat the document, like the structure of UUID used, the misuse of the reference element, or the use of a schemaLocation attribute.

8 Conclusion

Validation of CDA requirements presents a real challenge for national, regional and even cross enterprise infrastructures. In this paper, we have shown that error free CDA documents are rare and that there is a clear need for better validation tooling. The Gazelle ObjectsChecker based on UML model and OCL presents a good coverage of the basic CDA requirements and could be used as a common tool. Its methodology has proven to be highly effective in detecting non-conformity in the tested documents. The validation of national infrastructure material samples using this methodology detected hundreds of inconsistencies, which proves that the basic requirements are rarely respected. The analysis of the pool of IHE CDA documents showed that schematrons lack validation

rules and that combining schematrons business rules to the Gazelle ObjectsChecker tool might be necessary. Finally, the analysis of coverage of the different validation tools has proven the fact that most of them are far from covering 100% of the requirements. This raises the question of coverage reporting. When a tool provides an evaluation of the conformance of a document to some specifications, it is a good practice to provide information about the coverage of the specification.

Several extensions to this paper could be done, like a possible conjunction between the Gazelle ObjectsChecker tool and some CDA graphical specifications editors like Trifolia or ART-DECOR, to allow the automatic generation of validation materials. This methodology could also be extended to other healthcare XML based technologies, like for example for the HL7V3 messages or FHIR resources, which could improve the interoperability between healthcare systems, and avoid inconsistencies.

References

- [1] HL7, HL7 Clinical Document Architecture, Release 2.0
- [2] HL7, HL7 Reference Information Model, Release 2.07
- [3] HL7, Data Types - Abstract Specification, Release 1, 2004
- [4] HL7, XML Implementation Technology Specification - Data Types, Release 1, 2004
- [5] Abderrazek Boufahja and Eric Poiseau, Model-based Validation of XML Requirements, Applied in Healthcare IT Standards, HEALTHINF 2014
- [6] OMG, Object Constraint Language specification (OCL), v2.3.1, 2011
- [7] OMG, MOF Model to Text Transformation Language. V1.0, 2012
- [8] McLaughlin, Java And Xml Data Binding, O'Reilly, 2002

- [9] Birgit Demuth, Model and object verification by using DresdenOCL, Proceedings of the Russian-German Workshop Innovation Information Technologies: Theory and Practice, 2009
- [10] OASIS, Test Assertions Markup Language (TAML), v1.0, 2011
- [11] Eric Poiseau, Testing Tools Strategy, Healthcare Interoperability Testing and Conformance Harmonisation (HITCH), 2010
- [12] Abderrazek Boufahja, HL7 CDA R2 Basic Requirements, gazelle team /IHE-Europe, V0.1, April 14, 2014. available from: <http://gazelle.ihe.net/cda/cda-basic-req.pdf>
- [13] Trifolia Workbench, available from <https://www.lantanagroup.com/resources/trifolia/>
- [14] Sondra Renly, Rita Altamore, L. N. A new model for collaboration: Building CDA documents in MDHT. In AMIA Annual Symposium Proceeding, November, 2012
- [15] Kai U. Heitmann, ART-DECOR: a tool bridging the chasm between clinicians and health IT, in HL7 EUROPE NEWS, May 2014
- [16] epSOS Architecture and Design Interoperability Specification. Smart Open Services for European Patients, January 03, 2013
- [17] Rene Spronk and Grahame Grieve, Common issues found in implementations of the HL7 Clinical Document Architecture, 2008, http://www.ringholm.com/docs/03020_en_HL7_CDA_common_issues_error.htm
- [18] Keith W. Boone, Validating CDA Documents, April 8, 2010, <http://motorcycleguy.blogspot.fr/2010/04/validating-cda-documents.html>
- [19] Eclipse Instance Editor, http://hl7book.net/index.php?title=Eclipse_Instance_Editor
- [20] Well-Formed XML Documents Specification, [http://www.w3.org/TR/xml/\\$\sharp\\$sec-well-formed](http://www.w3.org/TR/xml/\sharpsec-well-formed)
- [21] R. Allen Wyke, Andrew Watt, XML Schema Essentials, John Wiley & Sons, 15 oct. 2002
- [22] Scott Bradner, RFC 2119: Keywords for use in RFCs to Indicate Requirement Levels, Harvard University, March 1997
- [23] Keith W. Boone, Where in the World is XDS, <http://motorcycleguy.blogspot.com/2010/01/where-in-world-is-xds.html>
- [24] Gazelle project, EVSClient tool, <http://gazelle.ihe.net/content/EVSClient>
- [25] ASIP, Cadre d'interopérabilité des SIS, Document chapeau. Agence des systemes d'information partages de sante, April 25, 2012
- [26] eHealth Suisse Online-CDA-Validator, <http://www.e-health-suisse.ch/umsetzung/00252/00273/index.html>
- [27] Georg Duftschmid, Wolf gang Dorda, The elga initiative: A plan for implementing nationwide electronic health records system in austria, Medical University of Vienna, 2009
- [28] Slavetinsky Vaclav, Stanovska Iva, Hrib Zdenek, Application integration in healthcare domain - Electronic prescription in Czech Republic using HL7 standards, 2010