German Medical Data Sciences: Bringing Data to Life
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI210048

ODM Clinical Data Generator: Syntactically Correct Clinical Data Based on Metadata Definition

Tobias J. BRIX[a,1], Ludger BECKER[b], Timm HARBICH[b], Johannes OEHM[a], Maximilian FECHNER[a], Martin DUGAS[a] and Michael STORCK[a]
[a] *Institute of Medical Informatics, University of Münster, Germany*
[b] *Institute of Computer Science, University of Münster, Germany*

Abstract. The Operational Data Model (ODM) is a data standard for interchanging clinical trial data. ODM contains the metadata definition of a study, i.e., case report forms, as well as the clinical data, i.e., the answers of the participants. The portal of medical data models is an infrastructure for creation, exchange, and analysis of medical metadata models. There, over 23000 metadata definitions can be downloaded in ODM format. Due to data protection law and privacy issues, clinical data is not contained in these files. Access to exemplary clinical test data in the desired metadata definition is necessary in order to evaluate systems claiming to support ODM or to evaluate if a planned statistical analysis can be performed with the defined data types. In this work, we present a web application, which generates syntactically correct clinical data in ODM format based on an uploaded ODM metadata definition. Data types and range constraints are taken into account. Data for up to one million participants can be generated in a reasonable amount of time. Thus, in combination with the portal of medical data models, a large number of ODM files including metadata definition and clinical data can be provided for testing of any ODM supporting system. The current version of the application can be tested at https://cdgen.uni-muenster.de and source code is available, under MIT license, at https://imigitlab.uni-muenster.de/published/odm-clinical-data-generator.

Keywords. Operational data model, data generation, test dataset

1. Introduction

The Operational Data Model (ODM) by the Clinical Data Interchange Standards Consortium (CDISC) is an XML-based standard used for exchanging clinical trial data. ODM is supported by a vast variety of Electronic Data Capture (EDC) vendors and applications [1]. An ODM file may include metadata, describing properties of the gathered data elements, and clinical data, i.e., answers of the participants. The portal of medical data models (MDM Portal) is Europe's largest academic infrastructure for creation, exchange, and analysis of medical metadata models [2]. Currently, over 23000 metadata definitions are provided in ODM format. These medical data models range from clinical trials to routine documentation. Due to data protection, no clinical data is provided along with the metadata. For testing EDC systems and planning statistical analyses, exemplary clinical data according to the metadata definition is necessary. It can be important to double-check with statisticians if the planned analyses can be performed

[1] Corresponding Author, Tobias J. Brix, Institute of Medical Informatics, University of Münster, Albert-Schweitzer-Campus 1, Building A11, 48149 Münster, Germany; E-mail: tobias.brix@uni-muenster.de.

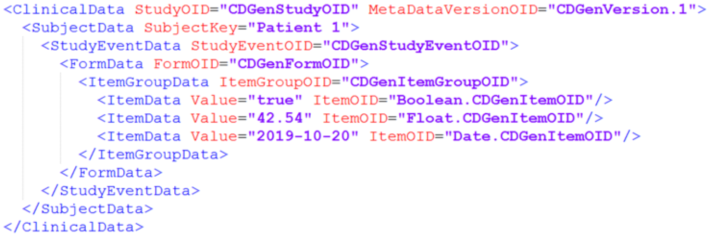
with the type of data (nominal, ordinal, etc.) or to debug and evaluate ODM-based tools like ODM Data Analysis [3].

Aim of this work is to develop a web application, which can read any syntactically correct ODM file and generate syntactically correct clinical data in ODM format. The application should support common ODM data types, range checks, and allow the configuration of certain data distributions. The system should be capable of generating clinical data for any number of subjects in a reasonable amount of time.

2. Methods

2.1. Structure of the Operational Data Model (ODM)

Only relevant parts for our application of ODM will be described here. More details can be found on the CDISC website. The ODM XML format is based on the structure of clinical trials. The metadata consists of a hierarchical structure beginning with a study element-tag, where multiple studies are supported in a single file. Inside a study, multiple study events can be defined. A study event represents the baseline and follow-up visits of a clinical trial. Each study event can contain multiple forms, which represent physical questionnaires of documentation. A form itself can have multiple item groups and items, which represent the data elements to be completed. The clinical data part of ODM follows the metadata structure, which describes study events, forms and items and references each hierarchical level by its unique ID. Figure 1 shows an example of clinical data representing a form with three data items. Most important for our application is the metadata definition of items. Each item consists of a data type and optional range checks to limit the range of answers. Another way to limit the answers is by using code lists, i.e., value sets of predefined answers. Furthermore, repeat keys are allowed, which indicate if a form or item group can be completed multiple times for a subject. This is used if the exact number of repetitions is unknown in advance, e.g., a form for the documentation of adverse events.



```
<ClinicalData StudyOID="CDGenStudyOID" MetaDataVersionOID="CDGenVersion.1">
  <SubjectData SubjectKey="Patient 1">
    <StudyEventData StudyEventOID="CDGenStudyEventOID">
      <FormData FormOID="CDGenFormOID">
        <ItemGroupData ItemGroupOID="CDGenItemGroupOID">
          <ItemData Value="true" ItemOID="Boolean.CDGenItemOID"/>
          <ItemData Value="42.54" ItemOID="Float.CDGenItemOID"/>
          <ItemData Value="2019-10-20" ItemOID="Date.CDGenItemOID"/>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
```

Figure 1. Exemplary clinical data part of an ODM file.

2.2. Used software and evaluation

We chose to develop a web application, since new versions can be provided to the community by updating the application server and no local installations are required. For web development, we chose Java in combination with the Spring framework.

For parsing ODM files, we use the library Java Architecture for XML Bindings (JAXB). JAXB allows for the automated generation of Java classes based on XML

Schema Definition (XSD) files. Since CDISC provides an XSD for ODM, JAXB is a natural choice. Besides generating the Java classes, JAXB also allows for transformation from XML files into Java objects of the pre-defined classes and vice versa [4].

To evaluate the technical feasibility of our software, we chose three ODM files from the MDM Portal with different numbers of forms and items to represent smaller and larger trial case report forms (CRFs). Table 1 shows all used metadata definitions with their quantity of forms and items. For each definition two subject counts, 1000 and 100000, have been generated. Thus, the data generation of clinical data of realistically large studies have been simulated. The corresponding generation times and resulting ODM file sizes have been recorded. The evaluation was performed on the same sever, which URL is linked in the results.

Table 1. Used metadata definitions of the evaluation with their contained metadata quantities.

ODM file	#Forms	#Items	Repeat Keys
WHO-Five Well-being index [5]	1	5	no
DIVI core data set intensive care [6]	5	108	no
Craniocerebral trauma register [7]	16	658	yes

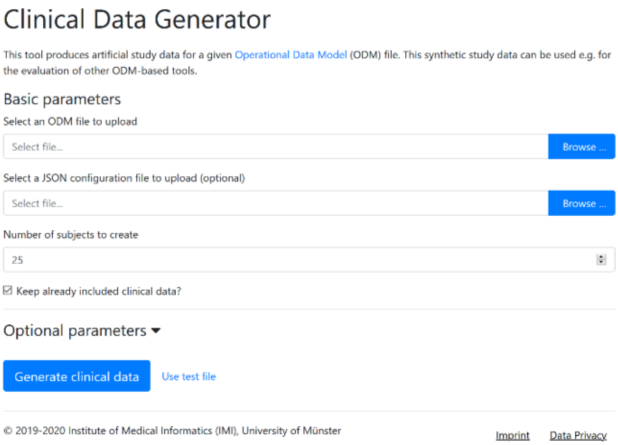
3. Results

The final application is available at https://cdgen.uni-muenster.de. A predefined example ODM file is downloadable to test the application. A screenshot can be seen in Figure 2. Furthermore, the source code can be obtained from https://imigitlab.uni-muenster.de/published/odm-clinical-data-generator under MIT license.

After selecting an ODM file and an optional configuration file for upload, the generation process can begin. Required configurations, e.g., the number of subjects to be generated, can be specified on the website itself without the requirement of a configuration file. During the upload process, the syntactical correctness of the given XML file is validated against the XSD of ODM versions 1.3.0 to 1.3.2. In case of errors, the process is aborted and all validation errors are displayed to facilitate the correction of the input file. The same applies for syntactical errors in the configuration file. If all syntax checks are passed, the clinical data generation process is started. With JAXB, the ODM file is parsed into Java objects and clinical data is generated. The final ODM file containing the metadata and clinical data is provided as downloadable Zip file. For reproducibility, this file also contains the original ODM and the used configuration file. Runtime errors during the generation do not stop the process and a list of occurred exceptions is provided in the Zip as generation notes. Runtime errors may occur due to invalid data types of XML attributes, which cannot be validated by the XSD file.

3.1. Application features

Important for scientific research is reproducibility. Therefore, the randomly generated data is based on a single seed, which can be defined in the configuration file. By using the same configuration, the same clinical data will be generated. If no seed is specified, a random seed will be generated and stored in the output configuration file. Thus, the output Zip contains all required information to reproduce the data generation.



Clinical Data Generator

This tool produces artificial study data for a given Operational Data Model (ODM) file. This synthetic study data can be used e.g. for the evaluation of other ODM-based tools.

Basic parameters

Select an ODM file to upload

Select file... | Browse ...

Select a JSON configuration file to upload (optional)

Select file... | Browse ...

Number of subjects to create

25

☑ Keep already included clinical data?

Optional parameters ▾

© 2019-2020 Institute of Medical Informatics (IMI), University of Münster

Figure 2. Screenshot of the final application available at https://cdgen.uni-muenster.de.

Currently, only the most frequently used data types of metadata models within the MDM Portal are supported. ODM supports further data types like partial dates, which are rarely used in clinical trials. For each data type, default value domains can be specified independent of local configurations in the ODM file. We support three types of item distributions. A uniform distribution is used by default. Alternatively, a normal distribution can be applied. In this case, the mean and standard deviation must be specified for each item. This distribution is not supported for items with discrete answer options like Boolean or code lists. For those items, fixed probabilities for each answer option can be specified. Further distributions, e.g. Poisson and log-normal, can be added by implementing provided Java interfaces and will be added in future versions. In addition, it is possible to configure the percentage of generated missing values, which increases the authenticity of the generated data towards real datasets. This parameter can be applied globally or at item level.

By writing the generated file directly into the response stream, no generated data has to be stored in main memory of the server. In addition, the Java objects can be directly deleted after writing, preventing the Java heap from overflow. Therefore, the generation of files of any size is possible.

3.2. Evaluation results

Table 2. Creation times of the evaluation. The item count and subject count are the relevant properties for the generation times, while the file size is used as indicator of successful generation.

ODM file	#Items	Times [s] / File size [mb]	
		(N=1000)	(N=100.000)
WHO-Five Well-being index	5	0.8 / 0.5	17.2 / 35.2
DIVI core data set intensive care	108	1.5 / 3.1	97.2 / 298.3
Craniocerebral trauma register	658	56.6 / 193.5	5421.0 / 2432.9

The evaluation results can be seen in Table 2. The software was able to generate clinical data for the ODM metadata files with different numbers of items and subject counts. All files are available from MDM Portal. The average generation time and file size of five runs were recorded. The generation times were between 1 second and 2 hours. The craniocerebral trauma register needed significantly more generation time, since it featured repeat keys generating the 658 items up to eight times for each subject.

4. Discussion

All defined goals of this work have been achieved. Clinical data can be generated without a natural limitation of file size. During the evaluation, dataset sizes, exceeding the usual number of subjects of clinical trials, were generated. In addition, most applications supporting ODM input may not be able to load files of this size. The generation times were reasonable for all evaluation branches.

The idea of generating generic datasets for evaluation tasks is not new. Benjamin Keen has proposed an open source tool for the generation of arbitrary datasets in multiple file formats [8]. After defining data items, e.g., phone numbers or country name, an export format like CSV or XML can be selected and the data generation starts. According to his website, the generation of up-to 5000 records at a time is supported. Support of ODM or any other medical data standard is not provided.

An application using ODM is the CDISC ODM Generator by XML4Pharma [9]. This tool can generate ODM files from given CSV data files. The missing metadata information is estimated based on the provided data structure. A combination of both tools can be used to generate output files similar to our application. The major benefit of our approach is the direct support of the ODM standard, allowing a quick generation of clinical data for all medical data models within the MDM Portal.

4.1. Limitations

The current version only supports the most commonly used data types, which are Boolean, String/Text, Integer, Float, Double, Date, Time and Datetime. Further ODM data types like PartialDate or HexBinary will be addressed in future versions, but their missing support should not be an issue in most use cases. Another unsupported feature of ODM is metadata versions. Items can be defined in different versions to reflect changes in the CRFs during study conduction. Currently, only the first version of metadata is used for generation. To the best of our knowledge, most vendors do not support metadata versions and should be therefore no limitation in practical use.

Furthermore, our application only generates syntactically correct clinical data. Ranges and distributions of items can be configured, ensuring medically consistent results for single items, but dependencies between items are not considered. Tools like Synthea achieve for certain diseases medical correctness by using predefined state transition machines based on empirical values [10]. However, the supported items and diseases are predefined models and Synthea cannot be used to generate data not defined in such a model. Although our approach lacks the medical correctness, it can be used for any ODM file. As mentioned in the introduction, the goal was not meant to generate medically plausible data, but large datasets to evaluate ODM based applications in a reasonable amount of time.

Finally, we only evaluated the technical feasibility of our application. It was shown, that up to 2 GB of clinical data can be generated in a reasonable amount of time. Still missing is the evaluation of practical feasibility to test and evaluate ODM supporting systems. While the generated data sets have been internally used to test different applications, no systematical evaluation has been performed.

5. Conclusion

In this work, we presented our tool for generating syntactically correct clinical data from a given ODM file containing metadata definition. The application is available as web application for easy usage. It supports the generation of over one hundred thousand subjects in a reasonable amount of time. Thus, using the MDM Portal, over 23.000 syntactically correct ODM files consisting of metadata definition and clinical data can be used for evaluation and validation purposes of any ODM supporting system.

Currently, our development is focused on ODM files. In future work, the generation will be extended to more data standards like openEHR or Fast Healthcare Interoperability Resources (FHIR) as in- and output format [11]. Furthermore, due to annotations with Unified Medical Language System (UMLS) codes within the MDM Portal, distributions of items could be estimated based on the underlying medical concept. Finally, the application is yet evaluated on pure technical application. It would be beneficial to evaluate its value in practical research settings, which also could be driven by the community using the provided tool.

6. Acknowledgements

Supported by BMBF grant No. 01ZZ1802V (HiGHmed/Münster)

References

[1] Hume S, Aerts J, Sarnikar S, Huser V. Current applications and future directions for the CDISC Operational Data Model standard: a methodological review. J Biomed Inform. 2016;60:352-62.

[2] Dugas M, Neuhaus P, Meidt A, Doods J, Storck M, Bruland P, Varghese J. Portal of medical data models: information infrastructure for medical research and healthcare. Database (Oxford). 2016;11.

[3] Brix TJ, Bruland P, Sarfraz S, Ernsting J, Neuhaus P, Storck M, Doods J, Ständer S, Dugas M, ODM Data Analysis - a tool for the automatic validation, monitoring and generation of generic descriptive statistics of patient data. PloS one. 2018;13(6).

[4] Soto-Rey I, Neuhaus P, Bruland P, Geßner S, Varghese J, Hegselmann S, Brix T, Dugas M, Storck M. Standardising the Development of ODM Converters: The ODMToolBox. Stud Health Technol Inform. 2018;247:231–235.

[5] Topp CW, Østergaard SD, Søndergaard S, Bech P. The WHO-5 Well-Being Index: a systematic review of the literature. Psychother Psychosom. 2015;84(3):167-76.

[6] Waydhas, C. Kerndatensatz Intensivmedizin 2010 der DIVI und DGAI. Anästh Intensivmed. 2010;*51:* 801-08.

[7] Rödiger M, Linden T, Althaus J, Debus O, Dugas M, Fiedler B, Petershofer A, Schulte C, Storck M, Teetz K, Völzke V, Wietholt G, Omran H. It Is All in the Head: Clinical Register for Patients with Traumatic Brain Injury: TBI Register. Neuropediatrics 2014;45.

[8] generatedata.com [Internet]. Benjamin Keen; [cited 2020 Mar 15]. Available from: https://github.com/benkeen/generatedata

[9] The XML4Pharma CDISC ODM Generator [Internet]. Thal: XML4Pharma, Jozef Aerts; [cited 2020 Mar 15]. Available from: http://www.xml4pharma.com/CDISC_ODM_Generator/index.html

[10] Walonoski J, Kramer M, Nichols J, Quina A, Moesel C, Hall D, Duffett C, Dube K, Gallagher T, McLachlan S. Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. J Am Med Inform Assoc. 2018;25(3):230-8.

[11] openEHR - a semantically enabled, vendor-independent health computing platform [Internet]. Atalag K, Beale T, Chen R, Gornik T, Heard S, McNicoll I; c2016 [cited 2020 Mar 15]. Available from: https://www.openehr.org/resources/white_paper_docs/openEHR_vendor_independent_platform.pdf