International Journal on Digital Libraries (2021) 22:365–383
https://doi.org/10.1007/s00799-021-00311-0

Improving data quality in large-scale repositories through conflict resolution

Artur Kulmukhametov[1] · Andreas Rauber[1] · Christoph Becker[2]

Received: 27 January 2021 / Revised: 20 September 2021 / Accepted: 20 September 2021 / Published online: 21 October 2021

Abstract
Digital repositories rely on technical metadata to manage their objects. The output of characterization tools is aggregated and analyzed through content profiling. The accuracy and correctness of characterization tools vary; they frequently produce contradicting outputs, resulting in metadata conflicts. The resulting metadata conflicts limit scalable preservation risk assessment and repository management. This article presents and evaluates a rule-based approach to improving data quality in this scenario through expert-conducted conflict resolution. We characterize the data quality challenges and present a method for developing conflict resolution rules to improve data quality. We evaluate the method and the resulting data quality improvements in an experiment on a publicly available document collection. The results demonstrate that our approach enables the effective resolution of conflicts by producing rules that reduce the number of conflicts in the data set from 17 to 3%. This replicable method for presents a significant improvement in content profiling technology for digital repositories, since the enhanced data quality can improve risk assessment and preservation management in digital repository systems.

Keywords Data quality · Technical metadata · Digital curation · Conflict resolution · Content profiling

1 Introduction

In increasingly large-scale digital repositories with preservation responsibility, a multitude of technical properties and file formats co-exist. Technical metadata about objects in large-scale digital repositories are a key component of effective repository management and long-term stewardship. At the same time, tasks such as format identification, characterization, and conformance validation have been key concerns in digital libraries for years [1,15].

The knowledge of technical properties exhibited by the assets contained in a repository comes from the output of characterization tools that are aggregated and analyzed through content profiling. The resulting profiles are used for such key tasks as repository management, reporting, and preservation risk assessment [6,10,14]. They enable informed decisions such as the choices of the best tools to use for access and management, the right metadata schemas to apply, or what risks to expect [1].

However, the accuracy and correctness of the tools vary [19,20], and they frequently disagree on such questions as *which format is this object encoded in?*. As a consequence, they produce contradicting outputs [23,24]. Another source of contradiction is evolving metadata [25] when information and metadata standards change over time. The resulting metadata conflicts raise risks for repository management and preservation.

Existing work has either ignored these conflicts or avoided them. The most common form this takes is to select one tool to use, deploy it in the ingest process of a repository system, and rely solely on its output. Recent tools and platforms provide a baseline architecture that makes these issues visible. For example, the File Information Toolset (*FITS*)[1] fuses multiple tool outputs and marks any contradictions between the tools in the combined output it produces. The content profiling platform C3PO exposes these conflicts so that they cannot be missed, but does not resolve them [7].

✉ Artur Kulmukhametov
artur.kulmukhametov@tuwien.ac.at

Andreas Rauber
rauber@ifs.tuwien.ac.at

Christoph Becker
christoph.becker@utoronto.ca

[1] Vienna University of Technology, Vienna, Austria

[2] University of Toronto, Toronto, Canada

[1] http://projects.iq.harvard.edu/fits/home.

However, these conflicts are significant: they affect substantial portions of typical repositories' content holdings and prevent repository managers from adequately selecting appropriate actions [24]. The resulting metadata quality issues can cause misunderstood distributions of what is in the repository, misapplied techniques, for example, feature extraction, or preservation actions, or inadequate metadata schemas and indexing methods. Simple majority voting will usually not produce correct results. As a consequence, significant research efforts are currently dedicated to rigorous testing and benchmarking approaches with the aim of producing more accurate and reliably correct metadata [2,13,15]. Yet, even with well-tested tools, conflicts will inevitably remain. For example, OpenOffice documents are valid Zip files as well. Tools reporting such documents as Zip files are correct as well.

While manual assessment is typically effective in identifying the correct file format [24], it is extremely expensive, and in large-scale environments, the investigation of individual objects is no longer possible.

Our contribution is a systematic rule-based approach to improving data quality in this scenario through an *expert-guided conflict resolution process*, a *stratified sampling method*, and *context-sensitive views*. It allows the user to measure the data quality, analyze and address metadata conflicts in large-scale repositories. The conflict resolution process guides the user in identifying the most frequently occurring groups of conflicts. The stratified sampling method supports the analysis of the patterns that arise to identify clusters of objects that can be treated by common rules. The context-sensitive views are built on top of a content profiling platform to enable users to focus on the most relevant property distributions depending on the content type.

In the following, we discuss related work on technical metadata quality in digital repositories and classify the conflicting format identification problem according to a framework in data quality research. We highlight key characteristics of this particular domain-specific conflict resolution problem and describe how our approach applies data quality principles and conflict resolution techniques. Section 3 describes our method step by step and highlights the requirements it poses on the content profiling platform. Section 4 describes the use cases of conflict resolution using the introduced approach. How we run the conflict resolution using scalable solutions is described in Sect. 5. In Sect. 6, we evaluate the method by applying it to the whole *Govdocs1* test corpus. We provide details on the experiment results, discuss the findings and their implications, and highlight open issues and opportunities for future work. Section 7 summarizes the paper and provides conclusions.

2 Background

2.1 Technical metadata quality in digital repositories

As Thibodeau writes, "the variety and complexity of digital information objects engender a basic criterion for evaluating possible digital preservation methods, namely, they must address this variety and complexity" [32]. The first step towards this is an understanding of the scope and technical diversity of content.

In digital preservation, *characterization* typically means a process of extraction of technical metadata from digital objects. Characterization processes aim to accurately describe the technical feature space of the representations of digital objects held in a repository. To do so, they analyze files and produce measures for each file according to a range of properties of interest, such as page counts for documents, the presence of embedded fonts in PDF, or whether a JPEG file is well-formed. For example, the characterization tool JHove considers a JPEG file to be well-formed based on three criteria: one such criterion is that the first three bytes of the file must be 0xFF, 0xD8, 0xFF.[2]

Characterization is one of the family of three content analysis processes in DP, the other two being *format identification*, which produces one or several identifiers, and *conformance validation*, which evaluates the degree to which the logical structure of the file corresponds to known constraints in a format specification. In contrast to these analysis processes, *rendering* aims to enact a *performance* [18] of the objects and may evaluate them for their fidelity [11].

Format identification, file characterization, and file format conformance validation have been identified as key tasks and concerns in digital stewardship [1,15]. Arguably, the identification of which format a file corresponds to is central to identifying its content type, deciding which tools to use to extract further information, classifying it, and validating, whether it is well formed and valid.

As part of digital preservation efforts, multiple specific format characterization tools were developed. Apart from the UNIX file(1) command, the first prominent identification tool was DROID, developed jointly with the PRONOM technical registry. The resulting format profiles have been widely shared to enable comparisons across repositories [10].

Subsequently, a variety of other identification and characterization tools have been developed for both broad and narrow domain-specific purposes. Increasingly, repositories are also making use of general-purpose tools developed by broader and larger communities, such as Apache Tika. Table 1 lists popular format identification tools and key characteristics. Of these tools, *FITS* is of particular inter-

[2] http://jhove.openpreservation.org/modules/jpeg/.

Table 1 Popular file format identification and characterization tools

Full name	Type	Description
DROID Digital Repository Object Identification	Identification	DROID is used in the UK National Archives. DROID was developed jointly with PRONOM, the file format registry, for digital preservation purposes, and uses its signature database [20,21]
FFIDENT Java metadata extraction/file format identification library	Identification	The tool extracts file format using magic number signatures. The tool is no longer maintained [21]
JHove JSTOR/Harvard Object Validation Environment	Identification, validation, and characterization	The tool is integrated in the workflows of major international preservation institutions. Jhove consists of modules for various format families: ASCII, PDF, JPEG, XML, ZIP, etc. [21]
NLNZ National Library of New Zealand Metadata Extractor	Characterization	It is a domain-specific tool designed for preservation activities at the National Library of New Zealand. The tool extracts metadata from images, documents, audio, video, markup languages and internet files and produces an XML for preservation purposes [21]
EXIFtool	Characterization	Exiftool read, writes and edits metadata information in files and supports 168 formats [21]
file UNIX file utility	Identification	A standard UNIX command-line program for recognizing file format [21]
FITS File Information Toolset	Aggregator	FITS is a part of Archivematica, a tool suite for digital preservation. A Java-based aggregator, *FITS* executes all of the above and maps their result into a common structured XML output schema [21]
JHove2 JSTOR/Harvard Object Validation Environment	Identification, validation, and characterization	The tool is integrated in the workflows of major international preservation institutions. The tool solves issues found in the first version of the tool, JHove [21]
FIDO Format Identification for Digital Objects	Identification	The tool was created by Open Preservation Foundation for simple integration into automated preservation workflows. An open-source command line application produces PRONOM-compliant results, PUIDs [21]
TIKA Apache Tika	Characterization	Developed by the Apache Software Foundation, TIKA is a general-purpose tool that supports over a thousand different formats and has become frequently used in digital preservation [21]

est because instead of characterizing files itself, it runs other tools and combines their output into a standard XML structure. As part of this, it maps the individual characteristics extracted by multiple tools and described in their terminology onto a unified terminology. In doing so, it also flags all properties with conflicting values. In this work, we use FITS outputs as a source of technical metadata on data collections.

However, characterization remains a complex process whose correctness cannot normally be proven [4]. Instead, systematic testing and benchmarking are required to establish reasonable confidence in the correctness of such tools [15]. The absence of robust data sets means that our ability to establish firm confidence in such tools' correctness remains limited [5].

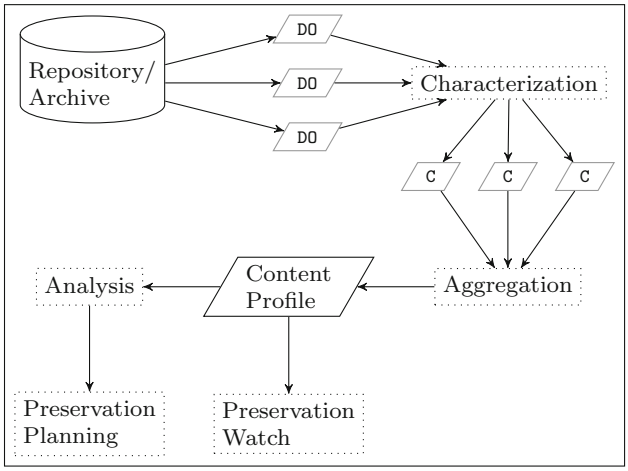

Fig. 1 Content profiling process: characterization tools decribe digital objects (DO) and produce characterization results (C), which are aggregated in the form of a content profile (CP). This information can be used in digital preservation activities, e.g., preservation watch, and/or further analyzed in preservation planning [6]

Several experiments have evaluated different tools [20,21]. Their results emphasized that misidentification is common [19] and highlighted the contradictory and potentially unreliable results obtained by different tools [24], and even by one tool in different versions over time [31]. This underlines the argument for the need to develop more evidence-based approaches [28], as presented in this paper.

2.2 Data quality and conflict resolution

Characterization results as created by these tools for sets of individual objects are aggregated and analyzed in a process sometimes referred to as *content profiling* [6,23]. The overall process is illustrated in Fig. 1. The process aggregates individual metadata records by merging, or fusing, multiple data points from individual sources. This allows the user to gain knowledge about the collection by generating property distributions, filtering, and sampling. The output of the process, called *content profile*, in the form of a document report is consumed by other DP processes, such as preservation planning (PP). In PP, information from the content profile helps to identify actions to meet organizational policies and mitigate risks in DP [6]. For example, the sample objects from the content profile allow the user to run experiments on finding the most reliable format migration path not using the original collection.

In this process of *data fusion*, a "data conflict" arises when "for the same real-world object (e.g., a student), semantically equivalent attributes, from one or more sources, do not agree on its attribute value (e.g., source 1 reporting '23' as the student's age, source 2 reporting '25')" [9, 1:7].

The conflicts may also occur because of the evolving metadata [25]. For example, one source may contain more recent information about the digital object than the other sources, e.g., a change of the name of the author or of the conference proceedings. The authors of [25] address the question of quick identification of evolving metadata in large-scale digital libraries followed by a conflict resolution using techniques from Information Retrieval.

Examples of data conflicts are presented in Table 2. The examples are taken from the Govdocs1 dataset, a corpus of about 1 million objects from the government domain [17]. It is openly available and accessible and regularly used for experiments in digital curation [20] and digital forensics [16]. We use this dataset throughout this work.

As the examples show, "Instance-level heterogeneities are caused by different, conflicting data values provided by distinct sources for the same objects. This type of heterogeneity can be caused by quality errors, such as accuracy, completeness, currency, and consistency errors; such errors may result, for instance, from independent processes that feed the different data sources" [3]. In this example, the various characterization and identification tools use different techniques, such as signatures or pattern matching, to identify the format of given files and create labels for various identifiers. Typically, these include a format name and version, MIME type, and for some tools a PRONOM unique identifier. Outputs of the tools partially overlap.

Bleiholder and Naumann [9] distinguish three types of strategies or attitudes toward data conflicts.

1. *Conflict ignorance* is a state of unawareness, in which the output of one tool is relied on. This stage describes the broad state of practice in digital repositories reasonably well, although practitioners conducting research in their repositories have increasingly taken notice of the existence of conflicts and quality issues [19–21,24].
2. *Conflict avoidance* aims to postpone or avoid the challenge of directly addressing the conflicts by following an approach similar to the above. However, in this case, there is awareness of potential quality issues, and the tool may be carefully selected based on testing results. Many advanced repositories follow this approach.
3. *Conflict resolution* requires the ability to address contradictions either individually (on the basis of the data instances) or indirectly (on the basis of other sources of information such as trust in specific information sources). In the case of instance-level conflict resolution, a further choice is between *mediating* contradicting values (such as using arithmetic operations to calculate averages for numeric values), or *deciding* on one of the values to choose [9]. The latter can be automated through voting

mechanisms, but our experiments will show that this can cause problematic errors.

Further differences arise between situations of *uncertainty* and situations of *contradiction*. In the first case, some tools report null values, whereas others provide valid measures. In the second, tools provide measures that differ [12]. In the former case, content profiling typically uses whatever value is available; the number of tools that agree on a value is recorded and can be used to evaluate confidence. In the latter case, conflict resolution mechanisms are required.

Conflicts are resolved by defining a *conflict resolution function*, which should be declaratively specified [27]. As discussed by Müller et al. [26], "conflicts between contradicting sources are often systematic, caused by some characteristic of the different sources". The goal for experts is to identify these characteristics and use their domain knowledge to resolve conflicts. As such, conflict resolution is always an extremely domain-specific process [9].

The earlier work [23] showed that in this scenario, reducing conflicts with a rule engine can yield promising results. However, the rules required programming skills to code them manually, so that the approach is not applicable and not scalable in addressing real-world data quality issues in digital repositories.

To resolve conflicts efficiently, experts require effective tool support that allows them to explore the data and understand its characteristics, then formulate resolution strategies and apply them to the data set. In doing so, the ability to "exploit different data sources that contain information about the same set of objects" is key [26].

2.3 Profiling platforms and issues

The need to support repository managers in large-scale digital stewardship has motivated the development of systems that support curators in gaining an up-to-date overview of the collections they manage, understand the diversity and different subsets, and identify particular subsets out of large collections. These systems aggregate and visualize specific characteristics of a set of files and support interactive analysis of visual information for purposes such as exploratory data analysis [8], partitioning, and sampling. To varying degrees, these systems support the well-known information visualization tasks of overview, zooming, filtering, details-on-demand, relating, history, and extracting [30] across the categorical and numerical data types encountered in technical metadata.

1. *Overview* features typically provide a high-level aggregate feature distribution according to such properties of interest as the content type, formats, time ranges, or format conformance.
2. *Zooming* allows visualization of greater levels of detail on sets of objects for instance-level analysis.
3. *Filtering* allows curators to select subsets of objects such as those in a particular format, or considered invalid.
4. *Details-on-demand* enables the isolation of a group of items in the collection on which a detailed investigation can be run. This task is of particular importance for content profiling since it allows drilling down into the collection. The user can get specific information available only for a subset of data, such as a distribution of word counts, used font styles, or copyright in Microsoft Word documents.
5. *Relating* allows viewing and exploration of relationships between digital objects and their properties. This task enables the identification of all objects with similar properties. For example, after examining the properties of an object, relating helps to detect all objects with the same set of values. On an aggregate level, correlations across properties may be of interest to identify answers to questions such as: Which formats have the highest rate of ill-formed objects?
6. *History* operations make the process of data analysis easier by allowing the user to backtrack, for example, to remove filters and return to prior views.
7. *Extracting* specific data or visuals allows them to be kept and shared. The extracted results of content profiling may be used in preservation processes, such as ingest and planning.

Xu et al. [33–35] developed an interactive visual analytics application and accompanying requirements based on user studies with archivists. The application uses technical metadata, which is automatically extracted from digital objects using DROID and then aggregated. For visualization, a treemap shows folders and their content as rectangles, the color, and size of which indicate the type and size of the content they represent. To enable data analysis, the tool calculates statistics, allows filtering queries, and provides means to interact with the data.

The content profiling tool *C3PO* [6] provides a configurable view in which the user has full control over the set of properties to be visualized. Figure 2 shows part of the overview in which a user sees distributions of property values for a set of default properties. The distributions are rendered in the form of a histogram with the most common property values on the x-axis and their counts on the y-axis. The long tail of property values is replaced by the value 'Other'. Characterization results that do not contain the given property are put in the corresponding category 'Unknown'. The visualizations are limited to single properties, and the user can analyze multidimensional distributions. The tool supports scalable processing via a MongoDB database in the backend and can be integrated with repository systems through a standardized interface. Due to its focus on data integration for repository

Fig. 2 C3PO gives a high-level overview of the entire Govdocs1 collection. Note the file format identification conflict rate of 17.6%; there are that many files with conflicts in any of the identification properties: format, format version, and mimetype

management and stewardship decisions and its extensible platform architecture, C3PO provides the basis for the work described here.

To enable such visualization and analysis, data integration systems for technical metadata are needed. For example, C3PO extracts metadata from sources such as *FITS* XML files, transforms it to a column store (persisted in MongoDB), and loads it for analysis. During this process, instance-level heterogeneities arise from conflicting data values provided by distinct sources for the same objects. These are often caused by quality issues such as those discussed above.

C3PO is designed so that these conflicts cannot be ignored: since there is no obvious way to assign categories for such objects, conflicting values are flagged and visualized, and the corresponding data entries are treated as separate "conflicted" categories.

3 Conflict resolution engine

3.1 Objectives and requirements

Experts' knowledge is often situated in and built from experience with individual collections and technologies. This work aims to enable them to deploy this knowledge on large-scale data sets where they can no longer inspect a majority of objects. For a large-scale feature space, sampling is typically used to support exploration and navigation. However, this is not applicable in this scenario because the individual clusters and categories and their exact distribution are unknown prior to inspection and conflict resolution. Instead, the entire data set must be loaded. Therefore, we aim to address the following requirements.

1. *Interactive visualization of the entire data set* is required for browsing, exploration, and filtering. The loading time for visualizations must not significantly interfere with the sequence of cognitive steps throughout the visualization tasks outlined above.
2. *Sampling* should allow the expert to select a partition of the collection and specify criteria over which a representative sample should be computed.
3. *Consistency and transparency* need to be ensured: the resolution of a conflict should not potentially result in new conflicts.

We introduce a systematic method, called conflict resolution engine (CRE), to support instance-level conflict resolution through a definition of conflict resolution functions in rule form. We implement CRE as an extension to the existing software tool and demonstrate it on a publicly available dataset. Preliminary steps toward the work described here have been presented in [23], which demonstrated the potential of rules to significantly increase data quality.

CRE enables the user to identify clusters of conflicts, iteratively analyze the conflicts and their sources, and develop triggering conditions and actions for a rule that is then applied to the data set. The method uses a new stratified sampling algorithm, which considers all viable combinations of property values that lead to conflicts.

Additionally, we extended the software tool with the context-sensitive views, which allow the users to focus on the most relevant statistical information about the data subsets with conflicts.

To have an overview of the approach, the following example demonstrates how all parts of the approach perform together. After the characterization metadata is ingested in the content profiling platform, the user decides on the data quality. If the data quality is low, the platform produces a list of conflicts and allows the user to select a conflict group

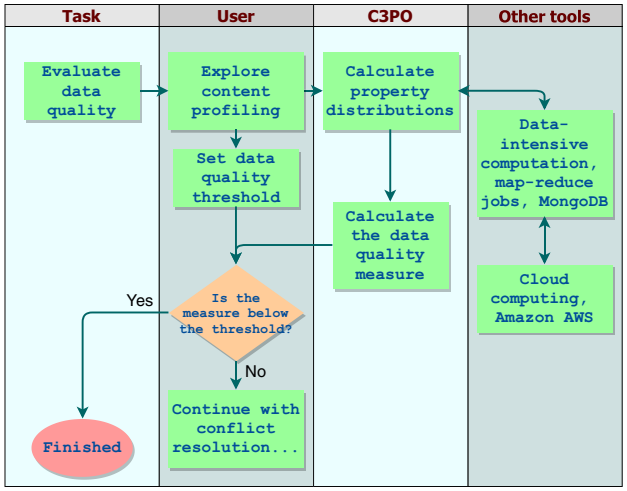

Fig. 3 The user decides if conflict resolution is necessary in Step 1

that presents the most efficient return on effort. On each iteration, the sampling algorithm picks representative objects and the user is provided with detailed information on the conflict using the context-sensitive views. This is followed by decision making on how to resolve a conflict and expressing this knowledge in the form of a conflict resolution rule. Executing the rules resolves the conflicts in the data set.

Later in this section, we describe CRE and its parts in detail.

3.2 Conflict resolution workflow

The workflow consists of four steps: data quality evaluation, conflict identification, conflict analysis, and conflict resolution. The steps provide the expert with a guide along the conflict resolution process.

3.2.1 Data quality evaluation

At the first step, the user decides if the conflict resolution is necessary for the given collection (Fig. 3). Besides content exploration, C3PO automatically calculates a *data quality measure*, which is defined as a percentage of digital objects with mismatching values in properties 'format', 'format version', and 'mimetype'.

The decision to continue the conflict resolution is to be made based on a user-defined threshold.

3.2.2 Conflict identification

Next, the user generates a *conflict overview table* (COT), which contains details on all conflict groups found (Fig. 4). COT is structurally similar to Table 2, except that the conflict overview table has additional information about each conflict: the number of occurrences of the conflict in the

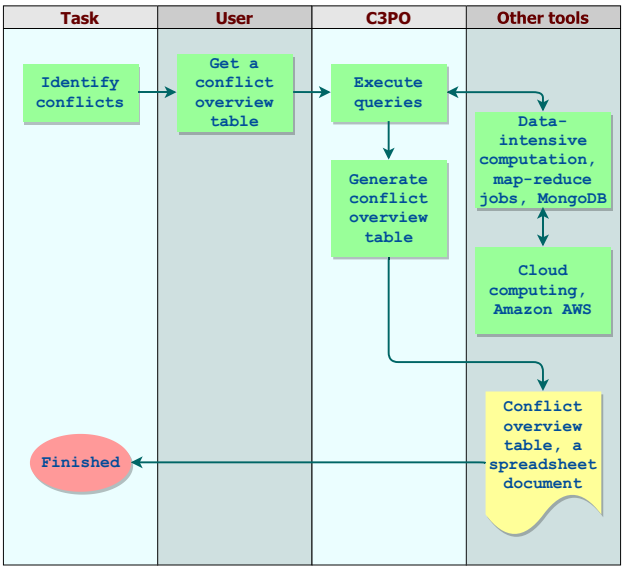

Fig. 4 The outcome of Step 2 is the conflict overview table

data, property values producing the conflict, and a query to filter the affected objects. The latter helps the user to navigate directly to the problematic subset.

Conflicts in the table are sorted in decreasing order by the number of occurrences. This allows the user to address conflicts in the table iteratively, starting from the most frequent.

When working through COT, the user has an opportunity to learn more about the conflicts. For example, it is possible to estimate how many conflicts (the number of rows in the table) must be resolved to achieve the desired threshold value for the data quality measure.

3.2.3 Conflict analysis

On the next step, the user *studies the conflicts* from the conflict overview table (Fig. 5). C3PO allows the user to drill down into the collection and filter the characterization results affected by the conflict. The templates help to focus on property distributions relevant to the currently chosen content type. Sampling picks representative examples, reducing the human effort to the manual examination of the digital objects. These capabilities of C3PO assist the user in decision making during conflict resolution. The knowledge will help to design and justify rules.

3.2.4 Conflict resolution

Conflict resolution is the last step in the workflow (Fig. 6). Here the user devises a strategy to resolve a particular conflict and expresses the strategy in the form of the conflict resolution rule.

A conflict resolution rule consists of *a trigger* and *an action*. The trigger describes a condition when the action must be executed. The trigger is represented as a set of triplets of type property, property value, source. An example of the trigger for Conflict 2 from Table 2 is a set of 3 triplets—format, Comma Separated Values, DROID:3.0, format, Plain Text, JHove:1.5, and format, Plain Text, File utility:5.03. The trigger activates the action only on characterization results of digital objects with 'format' property values "Comma Separated Values", "Plain Text", and "Plain Text" and produced by tools Droid v3.0, JHove v1.5, and File Utility v5.03, correspondingly.

The action part holds information on what changes must be done on the stored metadata of the digital objects. In the previous example, a possible action identified after inspecting several sample objects is to run an UPDATE query which removes the property value "Plain Text" produced by JHove v1.5 and File Utility v5.03, as DROID 3.0 turned out to provide the better fitting, more precise format specification (i.e., a CSV file, which obviously, on a more general level, is also a plain text file). Such a rule resolves the conflict.

The produced rules are self-descriptive and can be executed independently of each other. Thus, the order of a rule execution is not relevant.

C3PO assists the user during the creation of resolution rules. Section 4 provides in-depth examples of conflicts, affected digital objects, and rules to address the conflicts.

3.3 Stratified sampling

When analyzing digital objects in a repository, sampling helps to reduce the number of digital objects to process without losing the characteristics of the collection. A small representative subset of the data collection is helpful in content profiling tasks mentioned previously. It may speed up analysis of conflict groups followed by informed decisions.

Choosing samples from a collection can be done in different ways, e.g., based on size statistics, or a format distribution. The goal is to get samples that represent the original collection with respect to given properties. However, we are challenged by the problem of considering all viable combinations of property values, which lead to conflicts. Thus, a representative subset of digital objects forming the conflict group needs to be identified.

We apply *stratified sampling*, i.e., we select representatives of strata from the population. In our use case, a stratum can be used to differentiate digital objects with respect to conflicts. The stratum contains all objects that have the same property values for the given property/-ies leading to the conflict.

For example, consider Conflict 7 from Table 2. The conflict occurs in property 'format version' because of the values "2 0 0" and "3.1 type EPS Level 2" and is found in files with

Table 2 Exemplary format identification conflicts in Govdocs1

ID	Property	Characterization tools						Sample files	Conflicts count
		Exiftool v.7.74	DROID v.3.0	ffident v.0.2	JHove v.1.5	file utility v.5.03	NLNZ Extractor v.3.4GA		
1	format version	–	4.01	–	HTML 4.01	–	–	893232.html, 893860.html, 893416.html	22,889
2	format	–	Comma Separated Values	–	Plain text	Plain text	–	991446.csv, 991738.csv, 991355.csv	14,055
3	format	PPT	Microsoft Powerpoint Presentation	Microsoft Excel Format	–	–	–	992495.ppt, 042681.ppt, 042321.ppt	2483
4	format	MPEG 1/2 Audio Layer 3	GZIP Format	GZIP Format	GZIP Format	–	–	991336.gz, 085349.ps, 631134.eps	1998
5	format	Hypertext Markup Language	Extensible Markup Language	–	Extensible Markup Language	–	Hypertext Markup Language	998088.html, 993511.html, 935613.html	1769
6	MIME type	–	message /rfc822	–	text/plain	text/plain	–	880215.txt, 005631.txt, 000084.txt	688
7	format version	2.0 0	–	–	–	3.1 type EPS Level 2	–	063587.eps, 660383.eps, 712009.ps	21
8	MIME type	–	text/plain	image/ bmp	text/plain	text/plain	image/ bmp	306221.txt, 421159.txt, 418253.txt	4

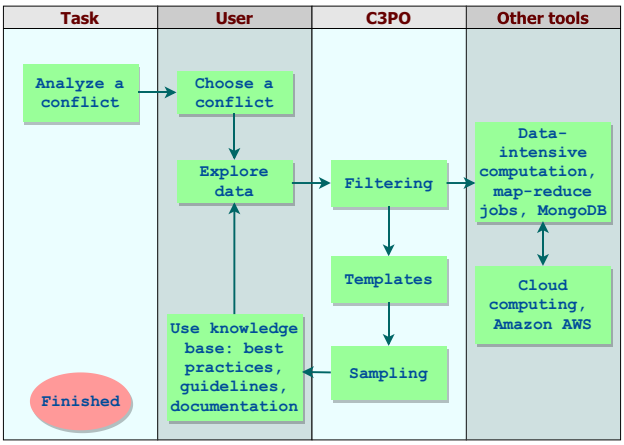

Fig. 5 Information about the conflict is gathered in Step 3

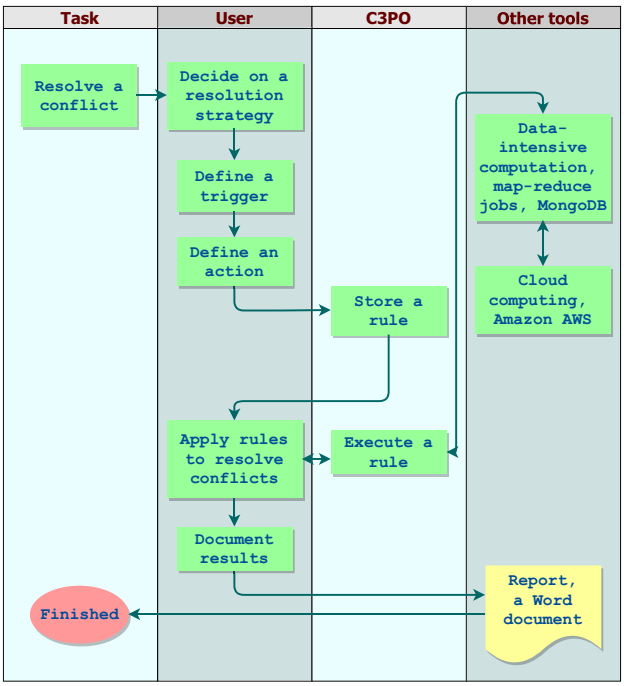

Fig. 6 The actual resolution of a conflict is done in Step 4

different file extensions, namely ".ps" and ".eps". The proposed sampling method puts objects in 4 strata; they are a result of the Cartesian product of all values for the properties: ("2 0 0", ".ps"), ("2 0 0", ".eps"), ("3.1 type EPS Level 2", ".ps"), and ("3.1 type EPS Level 2", ".eps"). Each stratum yields samples that are studied during conflict resolution.

More formally, an n-tuple of property values represents a subset of the collection that is sampled by n properties of interest. For example, if we consider two properties 'format' and 'mimetype', a tuple is a combination of any possible values for these properties e.g. (PDF, application/pdf), (html, text/html), etc. Each combination represents a subset of homogeneous objects from the collection. The subsets can be empty.

Let function *tuple(properties P, Object o)* get an n-tuple of values corresponding to the list of properties P from the object o. For example, tuple({'format', 'creation-date'}, paper.pdf) produces the tuple (PDF, 31.05.2021) from the file 'paper.pdf' created on 31.05.2021.

Let function *tuples(properties P, collection C)* produce a set of n-tuples based on properties P for all objects in the collection C.

Finally, let function *count(tuple t, collection C)* count objects from collection C, properties of which have exactly the values of the tuple t.

Based on these definitions, we introduce two measures. The first one, $pCoverage$, measures a property coverage of tuples in the collection. This can be expressed as:

$$pCoverage(P, C, t) = \frac{|\{o | o \in C, tuple(P, o) = t\}|}{|C|} \quad (1)$$

where C—is a collection set, P—a set of properties, t—an n-tuple.

In other words, the measure is used to calculate a number of objects matching the tuple relative to the size of the collection. It allows us to sort all strata by their size and to identify the biggest stratum. In case of conflict resolution, we calculate the measure for every unique conflict. It tells us what percentage of characterization results have the conflict.

Measure $tCoverage$ is defined as a ratio of the number of distinct tuples in the sample set to the number of distinct tuples in the collection:

$$tCoverage(P, S, C) = \frac{|tuples(P, S)|}{|tuples(P, C)|} \quad (2)$$

where C—is a collection set, S—a sample subset, P—a set of properties.

tCoverage tells us how many tuples were found in the samples relative to the total number of tuples in the collection for the given properties. In conflict resolution, the measure tells us what part of unique tuples is covered by the samples.

An example of calculating tCoverage and pCoverage is given in Fig. 7.

The pseudocode of the algorithm is presented in Algorithm 1. The user needs to define input parameters: a threshold for pCoverage, a threshold for tCoverage, and a set of properties of interest. The algorithm runs until the accumulated values of pCoverage and tCoverage reach the corresponding thresholds. Both thresholds take values from 0 to 1. The algorithm selects more samples when thresholds are set closer to 1. By increasing the thresholds, sampling picks more objects with less frequent conflicts (the long tail).

The algorithm extracts tuples based on the selected properties. The tuples are sorted in descending order of the number

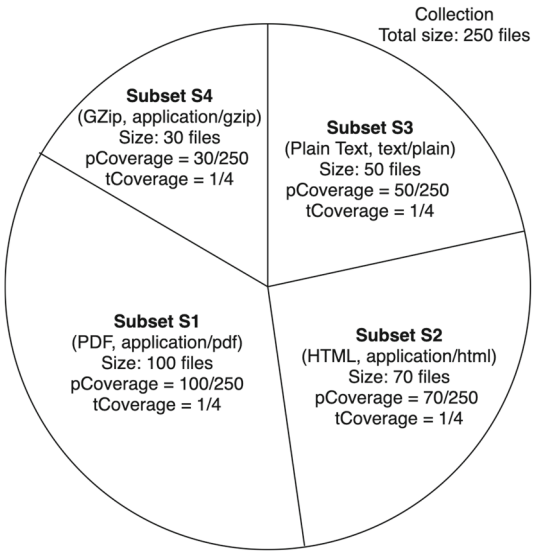

Fig. 7 $pCoverage$ and $tCoverage$ calculated for each subset of digital objects grouped by properties 'format' and 'mimetype'. The collection contains 4 tuples; therefore, $tCoverage$ of each subset equals 1/4

Algorithm 1: Stratified sampling for content profiling

```
pCoverage threshold → Tp;
tCoverage threshold → Tt;
Sample size threshold → Ts;
Collection size → Nc;
Select properties of interest → Properties;
Get all tuples for Properties → Tuples;
foreach tuple T in Tuples do
    Get all objects with T → Objects;
    Size of Objects → Nt ;
    Sample size for Objects = Ts*Nt/Nc → N;
    Choose N samples from Objects → Samples;
    Calculate pCoverage and tCoverage for Samples;
    pCoverage + pCoverageAcc → pCoverageAcc;
    tCoverage + tCoverageAcc → tCoverageAcc;
    N + NAcc → NAcc;
    if pCoverageAcc >= Tp or tCoverageAcc >= Tt or NAcc
    >= Ns then Exit
end
```

of digital objects with such tuples. The most frequent tuple comes first. Next, the algorithm counts the number of samples to extract from the collection for the given tuple and extracts the samples. The mentioned measures, pCoverage and tCoverage, and the number of samples are accumulated and compared against respective thresholds. If either of the conditions fulfills, the algorithm halts.

The output of the algorithm is a set of samples and statistics on the final accumulated values of $pCoverage$ and $tCoverage$, sample size, processing time, and a table with information on the measures for different sample sizes (see Table 3). The latter helps the user to decide on what thresholds for the measures and the sample size to use. As the algorithm iterates through the tuples, pCoverage and tCoverage approach their thresholds and the number of samples grows. If any of the thresholds are set to a value less than 1 or the sample size is reached, then some smaller subsets of homogeneous objects (from the long tail) will not be considered by the algorithm. Thus, the samples will not contain such objects. Given Table 3 as an example, the user needs to set the sample size to at least 6 in order to reach the threshold for pCoverage of 0.8.

Table 3 pCoverage, tCoverage, and different sample sizes for the Govdocs1

Sample size	pCoverage	tCoverage
2	0.244	0.024
3	0.416	0.048
4	0.587	0.071
5	0.754	0.095
6	0.835	0.119
7	0.932	0.143
8	0.951	0.167
9	0.963	0.190
10	0.973	0.214
11	0.977	0.238

3.4 Context-sensitive views

We implemented CRE as an extension to C3PO.[3] The extension adds the conflict resolution workflow, the stratified sampling, and context-sensitive views.

While some properties of files are general—such as the format or file creation date—other aspects of interest are genre- or format-specific. For example, the absence of embedded fonts in PDF formats can be a possible risk factor; the camera model is a relevant distinctor within raw photography. C3PO by default visualizes only generic properties, and there is generally a long tail of properties. (For the Govdocs1 test corpus, a total of 117 distinct properties are identified by the various tools.) It is difficult even for an expert to identify at which point in browsing and exploration which properties should be rendered.

Context-sensitive views support the user in browsing by associating specific filter expressions with sets of attributes to be visualized. A declarative mapping between filter conditions and sets of properties and visualization parameters provides an extensible mechanism that can be used to define views for specific domains, content types, or conflict patterns. This extensible definition of property sets to be added to the visualization according to the current filter set facilitates efficient exploratory data analysis.

[3] The tool is freely available via the project's Github page https://github.com/datascience/c3po.

For example, if the user wants to study document objects of the Govdocs1 collection, C3PO visualizes and focuses the user's attention on 14 properties[4] instead of 117. The properties are document-specific, therefore using them in an overview of other types of objects, e.g., images, will not result in useful information.

4 Use cases

We walk through three in-depth use cases of format identification conflicts from the Govdocs1 dataset and reason on a possible conflict resolution in each use case. We run C3PO on a computing cluster with sharded MongoDB; the results and the performance of such a setup are described in the next section.

After the characterization results of the Govdocs1 are ingested in C3PO, we open the C3PO main page in the web browser and start analyzing the dataset. The functions of C3PO are spread in different views, such as Overview, Object view, Samples, Conflicts, and Export. Loading of the collection overview takes on average 40 s. The overview contains the information on the conflict rate, and we decide that conflict resolution is necessary. We proceed to the conflict view and request the conflict overview table. C3PO produces the conflict overview table in 23 min.

4.1 Single rule

The first conflict occurs in property 'format' of 14.055 objects (Conflict 2 from Table 2). We know from the conflict overview table that two tools report "Plain text" as the property value and one tool reports "Comma Separated Values". By following a hyperlink from the conflict overview table, we get to the overview of the 14.055 objects in C3PO.

The overview consists of histograms, the number of which is controlled by the predefined context-sensitive views in C3PO. This helps to focus on histograms that contain conflicts in the property values of the subset.

It is impossible for us to go through all 14.055 objects and to study their properties to resolve the conflict. Instead, we go to the sampling view and use the stratified sampling method to get representative objects. The method enables C3PO to identify tuples, calculate the measures, and select samples based on the threshold values. We set the following values to the thresholds: 0.8—for pCoverage and tCoverage, 11—for sampling size. The values are high enough to produce a representative majority of the objects and reduce the manual effort to assess samples later. After we start the process, C3PO runs the calculations and outputs the samples in 12 s.

The output contains 11 objects, which we study in detail using the object view as well as accessing the original files. All samples have conflicts in properties 'format', 'mimetype' and 'puid' (see Table 4 rows 1–5). Property 'puid' contains value "x-fmt/18", which corresponds to "Comma Separated Values" format according to the PRONOM registry. Property 'filename' has value with the extension "csv", which is a common abbreviation of the format. The files contain data structured in columns.

We decide that the correct property value is "Comma Separated Values," and the conflict is resolved by setting this property value. All the samples are homogeneous; they have the same property values for the properties of interest, and characterization results contain the same conflict. A rule resolving the conflict in one of the objects can be applied to the whole subset.

C3PO allows us to create the rule using the object view (see Fig. 8). The rule can be expressed as follows:

```
Rule "Format CSV 14055" {
  trigger = [
    (format, "Plain text", "file utility:5.03"),
    (format, "Plain text", "JHove:1.5"),
    (format, "Comma Separated Values", "Droid 3.0")],
  action = REMOVE values [
    (format, "Plain text", "file utility:5.03"),
    (format, "Plain text", "JHove:1.5")]
}
```

We decide to only remove parts of the records from the database, because removal operations leave the original records without overwriting.

We give the rule a name and save it. On the conflict page, we select the rule and execute it. C3PO automatically finds all objects matching the trigger and updates the records by removing the incorrect data. The rule solves the conflict in characterization results of all the 14.055 objects. The reduced conflict rate is visible on the overview page.

4.2 Multiple rules

The second conflict occurs in property "format" of 2.483 objects (Conflict 3 from Table 2). According to the conflict overview table, three tools report 3 different property values: "PPT", "Microsoft Excel Format", and "Microsoft PowerPoint Presentation". Opening the overview in C3PO, we see two values on histograms of properties "puid" and "format version". It means that there are 2 sub-subsets of objects with conflicts that might be caused by separate reasons. Objects of the first subset have value "fmt/126" for property "puid" and value "97-2002" for property "format version"; objects of the second subset have values "fmt/125" and "95" (see Table 4 rows 6–13).

[4] The properties are 'format', 'format version', 'mimetype', 'file extension', 'page count', 'word count', 'character count', 'char set', 'has annotations', 'has forms', 'has outline', 'is protected', 'is right managed', 'is tagged', 'line break', 'markup basis', and 'word size'.

Table 4 Exemplary characterization results of Govdocs1 (highlighted property values require separate conflict resolution rules)

ID	File	Property	Status	Values (as reported by tools)
1	991446.csv	Format	Conflict	Plain text (File Utility:5.03, JHove:1.5), CSV (DROID:3.0)
2	–	MIME Type	Conflict	text/plain (File Utility:5.03, JHove:1.5), text/csv (DROID:3.0)
3	–	PUID	OK	x-fmt/18 (DROID:3.0)
4	–	Valid	OK	True (JHove:1.5)
5	–	Line Break	OK	CR/LF (JHove:1.5)
6	991980.ppt	Format	Conflict	MS Excel (ffident:0.2), PPT (Exiftool:7.74), MS Powerpoint (DROID:3.0)
7	–	MIME Type	Conflict	application/vnd.ms-excel (ffident:0.2), application/vnd.ms-powerpoint (Exiftool:7.74, DROID:3.0)
8	–	Format Version	Conflict	97-2002 (DROID:3.0)
9	–	PUID	Conflict	fmt/126 (DROID:3.0)
10	042284.ppt	Format	Conflict	MS Excel (ffident:0.2), PPT (Exiftool:7.74), MS Powerpoint (DROID:3.0)
11	–	MIME Type	Conflict	application/vnd.ms-excel (ffident:0.2), application/vnd.ms-powerpoint (Exiftool:7.74, DROID:3.0)
12	–	Format Version	Conflict	95 (DROID:3.0)
13	–	PUID	Conflict	fmt/125 (DROID:3.0)
14	991336.gz	Format	Conflict	GZIP (ffident:0.2, File Utility:5.03, DROID:3.0), MPEG 1/2 Audio Layer (Exiftool:7.74)
15	–	MIME Type	Conflict	application/x-gzip (ffident:0.2, File Utility:5.03, DROID:3.0), audio/mpeg (Exiftool:7.74)
16	–	File Extension	OK	.gz (OIS File Information:0.1)
17	–	PUID	Conflict	fmt/266 (DROID:3.0)
18	085349.ps	Format	Conflict	GZIP (ffident:0.2, File Utility:5.03, DROID:3.0), MPEG 1/2 Audio Layer (Exiftool:7.74)
19	–	MIME Type	Conflict	application/x-gzip (ffident:0.2, File Utility:5.03, DROID:3.0), audio/mpeg (Exiftool:7.74)
20	–	File Extension	OK	.ps (OIS File Information:0.1)
21	–	PUID	Conflict	fmt/266 (DROID:3.0)
22	631134.eps	Format	Conflict	GZIP (ffident:0.2, File Utility:5.03, DROID:3.0), MPEG 1/2 Audio Layer (Exiftool:7.74)
23	–	MIME Type	Conflict	application/x-gzip (ffident:0.2, File Utility:5.03, DROID:3.0), audio/mpeg (Exiftool:7.74)
24	–	File Extension	OK	.eps (OIS File Information:0.1)
25	–	PUID	Conflict	fmt/266 (DROID:3.0)

Fig. 8 All characterization results on a digital object are accessible on the object view. The view also allows a user to create conflict resolution rules. To do so, we first mark the property, for example 'format', as a trigger using the checkbox. Then, we check the boxes of incorrect or redundant property values to be removed, for example, 'Plain text'

As in the previous use case, it is not possible to study all the objects and we rely on sampling. We use the stratified sampling algorithm with the same threshold values as before. The sampling algorithm based on properties "puid" and "format version" produces samples that represent the identified subsets.

We study the original files of the samples and the characterization results using the object view in C3PO. The objects of the two subsets have different format versions; thus, each subset requires a separate resolution rule. We decide that the correct property value is "Microsoft PowerPoint Presentation". We select properties "format" and "puid" as a trigger and mark the incorrect property values for removal.

Based on our instruction, C3PO creates the following two rules:

```
Rule "Format PPT 2483 fmt/126" {
  trigger = [
    (format, "MS Excel", "ffident:0.2"),
    (format, "PPT", "Exiftool:7.74"),
    (format, "MS Powerpoint Presentation", "Droid 3.0"),
    (puid, "fmt/126", "Droid 3.0")],
  action = REMOVE values [
    (format, "MS Excel", "ffident:0.2"),
    (format, "PPT", "Exiftool:7.74")]
}
```

We save the rules and execute them. C3PO runs queries, finds matching records in the database, and updates them. The rules resolve the conflict in characterization results of the 2.483 objects.

4.3 Single rule for multiple subsets

Another conflict is caused by values of the property 'format' in characterization results of 1.998 objects (Conflict 4 from Table 2). One tool reports the value "MPEG 1/2 Audio Layer 3", while the other three tools report "GZIP Format". The overview in C3PO shows a histogram on the property "file extension" with property values ".gz", ".ps", and ".eps" (see Table 4 rows 14–25).

We use the stratified sampling algorithm with the same threshold values as before and the additional property "file extension". It produces samples with all the identified file extensions.

During the manual examination, we find out that the format of all objects is "GZIP Format" and they are valid archive files. Therefore, we do not add an additional condition on the property "file extension". One resolution rule is sufficient to cover conflicts from the three identified subsets. We create the following rule in C3PO:

```
Rule "Format PPT 2483 fmt/125" {
  trigger = [
    (format, "MS Excel", "ffident:0.2"),
    (format, "PPT", "Exiftool:7.74"),
    (format, "MS Powerpoint Presentation", "Droid 3.0"),
    (puid, "fmt/125", "Droid 3.0")],
  action = REMOVE values [
    (format, "MS Excel", "ffident:0.2"),
    (format, "PPT", "Exiftool:7.74")]
}
```

We save and execute the rule. The rule resolves the conflict in 1.998 objects after execution.

The three use cases demonstrate different approaches to rule generation. The samples in the first use case are homogeneous; one trigger with conditions on the target property enables the user to find all such objects and to resolve the conflict within them. The conflict from the second use case can be resolved using two rules. Each subset of samples requires a separate rule with additional conditions to the rule trigger. Finally, in the third use case sampling identifies the three subsets of objects. Nonetheless, one rule is capable of addressing the conflicts in all three subsets.

Regardless of the conflict and the resolution rule complexity, C3PO is capable to identify conflicts, helping the user to devise a resolution, and to express the knowledge in the form of a rule.

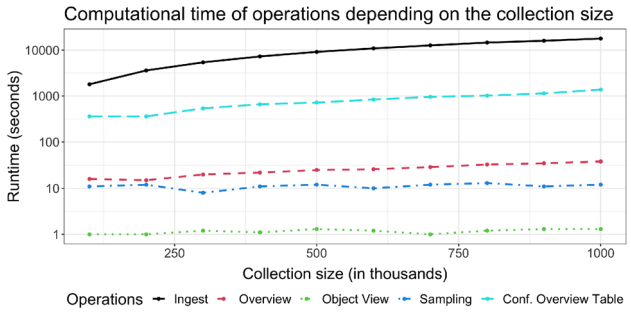

Fig. 9 Runtimes of the Ingest process and the Conflict Overview Table Generation increase linearly as the collection size grows. Note logarithmic scale on *Y*-axis

5 Scalability

Running processes on data at scale, such as data quality improvement, is challenging. A conflict resolution in large-scale collections is impossible if the user needs to resolve all conflicts individually, even when using scalable software solutions, e.g., databases.

CRE provides an answer to this problem. It eliminated the need to visit all objects of the collection. Instead, we generate the conflict overview table and get samples for each unique conflict. We create and, then, apply the resolution rules based on samples. A single rule execution can resolve thousands of conflicts, as we show in Sect. 6. The CRE is split into smaller tasks, which can be separately run at scale. Thus, our method enables the user to apply and benefit from the existing scalable solutions.

When conflict resolution supports scalability, improving the processing performance can be achieved by increasing available computational power, e.g., allocating additional CPU resources.

C3PO uses MongoDB as storage and queries the data using map-reduce jobs. While this can in principle provide highly scalable analytics, previous research on a very large test data set highlighted that the platform was not sufficiently scalable for large data sets [7,29].

This problem is addressed by integrating C3PO with Amazon Web Services. It allows us to increase the throughput of map-reduce jobs. Ten AWS nodes of type T2.Medium serve an instance of C3PO and a MongoDB sharded cluster. Sharding enables data replication among the nodes so that the same data can be used in parallel. Statistics on the runtime of processes involved in the conflict resolution are provided in Fig. 9. The runtimes are measured for portions of the Govdocs1 ingested in C3PO: first after ingesting 100.000 objects, then after 200.000 objects, and so on until the whole collection is ingested. This information is helpful to plan on the required performance of operations and adjust the configuration of the database.

CRE makes conflict resolution in large-scale collections using C3PO feasible and allows the user to improve the performance of the operations by well-known methods, such as adding computational nodes to a cluster.

6 Evaluation

CRE and its instantiation in C3PO require evaluation to support our claims. We split the evaluation as follows. First, we state questions for the evaluation. Then, we define the evaluation process. After that we apply the conflict resolution process to the Govdocs1, collect statistical data, and manually assess the rules. In the end, we discuss the strong and weak points of CRE and list possible improvement steps.

6.1 What to evaluate and how?

The complexity of the evaluation comes from the fact that the approach consists of a number of processes: the conflict resolution workflow, the sampling, map-reduce queries, measure calculations, navigation using UI, the conflict overview table generation, and rules generation. Each activity requires an independent evaluation. However, these will not give an overall picture of the performance of the approach. We want to evaluate the characteristics of CRE with respect to the number of resolved conflicts. This is of interest to practitioners seeking a method applicable in a real-world scenario. Thus, we are interested in whether CRE is:

- *effective* in improving data quality, i.e., whether it successfully removes conflicts without introducing errors;
- *efficient* in terms of user effort; and
- **efficient** in terms of processing resources; and
- *generalizable*, i.e., the rules created to resolve conflict for one data collection are also applicable to another collection.

We conduct a quantitative evaluation and measure the performance of C3PO in the context of conflict resolution. During the evaluation, we apply CRE to resolve conflicts in the dataset until the desired data quality threshold is reached.

6.2 Before the resolution process

We use the Govdocs1 collection in this evaluation. The collection contains 986.278 files with a total size of 488 MB. The file format distribution is presented in Fig. 10. 160.372 conflicts occurred during format identification. Initially, 17% of the records of Govdocs1 contain conflicts in the format identification properties.

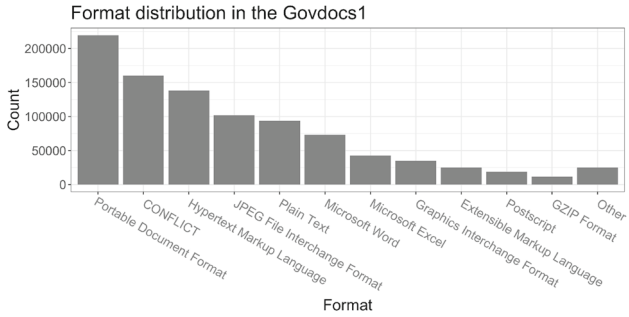

Fig. 10 This figure shows the distribution of files of different formats in Govdocs1. Conflicted metadata make up 17% of the collection

Fig. 11 The drop in the conflict rate of the Govdocs1 collection after applying the proposed method. The conflict rate is reduced to 3% after applying 40 conflict resolution rules

The most common formats in the collection before conflict resolution are PDF (219.502 files), HTML (138.139 files), and JPEG (102.033 files).

We set the target conflict rate to 3% to demonstrate the amount of effort required to resolve conflicts in a real-world scenario.

6.3 Reaching the target threshold

We follow the conflict resolution workflow to achieve the target threshold.

We start with the overview tab of C3PO containing histograms of property values of the Govdocs as well as the conflict rate measure. The context-sensitive views control the default set of histograms shown to the user; irrelevant histograms are hidden.

After inspecting the conflict rate, we decide to initiate conflict resolution and continue with the workflow. The next step in the workflow is a generation of the conflict overview table. From the table, we can directly navigate ourselves to the subset of the collection with the conflict of interest; this is achieved using filter criteria. In situations when the subset is too big (thousands of digital objects) to investigate manually, we employ the sampling. The samples reduce the amount of manual effort required for the examination of the original digital objects. Given the objects, the user needs to have experience of looking for relevant information on possible conflict resolution strategies. For example, in the case of file format identification, there are tools like PRONOM registry, W3C validators, Apache Tika, standard viewers, and editors.

Once we devise a resolution strategy, we create a conflict resolution rule in C3PO. Later we execute the rules and the results of execution immediately affect the conflict rate.

We create 40 conflict resolution rules and achieve the residual conflict rate of 2,992% after executing the rules. The rules resolve 389.578 individual conflicts. A single rule resolves conflicts in characterization results of at most 45.295 digital objects. There are 28.330 characterization results with conflicts left in the collection after executing all the rules. The drop of the conflict rate after application of each of the rules is presented in Fig. 11.

If the organization wanted to achieve an even higher level of data quality, e.g., by reducing the conflict rate to 2%, then we would need to continue addressing conflicts from the conflict overview table. In the case of the Govdocs1 collection, the conflict rate of 2% is achievable when at least the first 129 conflicts from the conflict overview table are addressed.

Based on statistics within this evaluation, the average time to analyze a conflict (using third-party tools and validators, assessing the samples) is 37.3 min. The average time to create a rule is 3.85 min. The statistics may vary for other users.

6.4 Observations

We collected various data per each rule, such as calculated representative samples, the number of resolved conflicts, incorrectly resolved conflicts, time to create and apply rules [22].

The results of running CRE on Govdocs1 are presented in Fig. 12. It shows the distribution of resolved conflicts in affected characterization results with respect to their format. Most conflicts are resolved in the characterization results that correspond to digital objects of "HTML" and "MS Powerpoint" formats. Interestingly, the most frequent format in Govdocs1, "PDF", is not presented in the diagram. The characterization tools agree on property values of PDF files more often than on property values of other formats.

The diagram gives an overview of the resolved conflicts but does not tell us what leads to the conflicts. Next, we shed light on what leads to the conflicts. The generated rules contain the knowledge about each conflict case.

A closer look at the trigger part of the generated resolution rules reveals some patterns. The rules can be grouped into three types by the similarity of the addressed property values: different values, related values, and noisy values.

Rules of the type *"differing values"* are unrelated to each other. The conflicting values belong to different content types. Conflict 2 in Table 2 is an example of this type. The

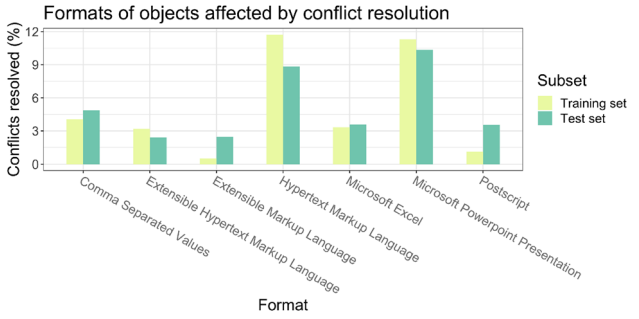

Fig. 12 HTML and Powerpoint have the highest conflict percentage in both training and test sets

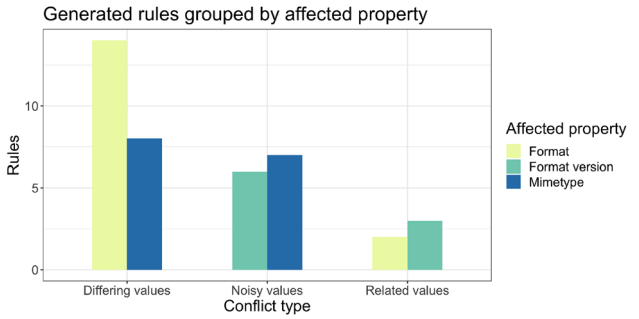

Fig. 13 Distribution of conflict types with respect to similarity of property values

characterization tools report values "Microsoft Powerpoint Presentation" and "Microsoft Excel Format".

The second group contains rules that address "*related values*" of related formats. These values describe formats that are structurally similar, but their differentiation is acknowledged. For example, Conflict 4 in Table 2 occurs because of format property values "Hypertext Markup Language" and "Extensible Markup Language".

The rules of type "*noisy values*" address conflicts that agree on the property value, but the strings do not completely match or the characterization tools mark these properties as conflicted by mistake. An example of this type is Conflict 8 in Table 2. Two values "HTML 4.01" and "4.01" describe the same label for the property "format version".

Figure 13 shows the distribution of the resolution rules by the addressed property and grouped by the mentioned conflict types. The rules of the first type, resolving different values, are the most frequent. The characterization tools produce unrelated values during file format identification more frequently than other conflicts.

Conflicts that are resolved by rules of the type "noisy values" can be avoided when characterization tools use a common vocabulary or a centralized knowledge base for possible property values. Solving this kind of conflict can be done by providing a mapping to such a knowledge base. This mapping should link a less common property value label to a more common property value label. A correct value can be looked up from the mapping before being written to the database. This would ensure the tools report the same values.

So far, the rules were designed based on the sample objects. To validate correctness and coverage of the rules, we split the collection into two subsets (train, test) by taking, correspondingly, the first 90% of objects to generate rules and the last 10% of objects to validate the rules. An evaluation of the correctness of the conflict resolution on the test set provides information about how well the approach resolves conflicts on unseen data.

The evaluation is done manually by examining digital objects with conflicts that are addressed by a given rule. First, we determine representative samples for each rule from the training set and from the test set.

Then, the samples from the test and the training sets are examined for the presence of false positives and false negatives. A false-positive outcome is an inaccurate conflict resolution, e.g., a resolution rule affects a digital object when it should not. A false-negative outcome is a resolution that did not occur although it should have.

The manual examination of the samples shows that the method correctly resolves conflicts on the test subset of the Govdocs1 collection. We extract 10 samples from both the training and the test sets for each generated rule. We correlate the number of resolved conflicts in both sets after each execution of the rules. Pearson's test shows a 93.3% correlation rate between results of applying resolution rules to the training and test sets with p-value < 0.0001. The correlation is not 100% because the distribution of file formats in the subsets is not identical, e.g., the ratio of HTML files in the test set is greater than the one in the training set.

During the manual assessment, we did not encounter any false-positive or false-negative outcomes. No situation of an incorrect resolution is detected. CRE works as expected on both subsets.

We see that the rules work well on the unseen data. However, the characterization results in the unseen data must be produced by the same toolset that was used in the generation of the training data.

There are situations when a resolution is not clear:

- Invalid HTML, HTML Transitional, XHTML, HTML Strict, and XML documents may contain ambiguous values, e.g., an invalid HTML Strict document with an iframe tag can be characterized by conflicting values "HTML Strict" and "HTML Transitional" at the same time.
- CSV files and plain text files are similar. It is not clear how to differentiate them, as CSV lacks any CSV-specific flags.

Both cases can be addressed by improving the characterization tools. In the first case, the precise type of HTML

is not defined when the document is invalid. In the second case, a repeatable pattern of delimiter and new-line symbols differentiates CSV files from plain text files.

During the evaluation, we found that some tools produce correct characterization results more often than the other tools. Twenty-six out of the 40 rules mention DROID 3.0 as a source of correct values. It is possible that the tool is better suited for the Govdocs1, whereas the others might produce more correct results on others types of formats.

Regarding the content profiling platform, C3PO is still a prototype tool and requires improvements to bring more value to other users and researchers in the community. We have identified the following shortcoming. The tool was initially designed to assist in data exploration through overviews, filtering, drill-downs, and sampling. Unlike the mentioned operations, the conflict resolution workflow requires the user to perform the steps in the defined order. C3PO lacks a centralized view, which would help to guide the users in the process. This can be improved in the next versions of the tool.

7 Conclusions and outlook

Technical metadata about objects in large-scale digital repositories come from the output of characterization tools aggregated and analyzed through content profiling. However, the accuracy and correctness of the tools vary, and they frequently produce contradicting outputs. The resulting metadata conflicts raise risks for repository management and preservation. Existing work has either ignored these conflicts or avoided them. The latest platforms provide a baseline architecture that makes these issues visible so that they cannot be avoided but do not resolve them. This article presented and evaluated a rule-based approach to improving data quality in this scenario through expert-guided conflict resolution. We described the content profiling platform C3PO, outlined gaps in information visualization to support this domain, and characterized the nature of the data quality problem caused by conflicting and ill-described characterization labels.

The proposed conflict resolution approach provides new capabilities to the content profiling platform, including the conflict resolution workflow and the stratified sampling algorithm. The approach was evaluated, and the conclusion was drawn based on the resulting data quality improvements in a study on the Govdocs1 collection.

The results demonstrate that the targeted task-focused information visualization supports effective conflict resolution and that outcomes are highly effective in reducing the number of conflicts in real-world data sets. In the study, we reduced the conflict rate from 17% to 3% in the data set. This increases the quality of the resulting content profiles. The revised platform architecture enables deployment on cloud platforms that offer scalable solutions to data storage and processing.

This method for improving data quality presents a significant improvement in content profiling technology for digital repositories since the enhanced data quality can improve risk assessment and preservation management in digital repository systems. The practitioners can try out the tool on their collections to gain a deeper understanding of the content. They can also assess the data quality and improve it to match their policies.

All code, documentation, and data are freely available [22].

Acknowledgements Part of this work was supported by the Vienna Science and Technology Fund (WWTF) through the project BenchmarkDP (ICT12- 046) as well as the National Science and Engineering Research Council NSERC through RGPIN-2016-06640, the Canada Foundation for Innovation and the Ontario Research Foundation. The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

Funding Open access funding provided by TU Wien (TUW).

References

1. Abrams, S., Morrissey, S., Cramer, T.: What? So what: the next-generation JHOVE2 architecture for format-aware characterization. Int. J. Digit. Curation **4**(3), 123–136 (2009). https://doi.org/10.2218/ijdc.v4i3.122
2. Arocena, P.C., Glavic, B., Mecca, G., Miller, R.J., Papotti, P., Santoro, D.: Benchmarking data curation systems. IEEE Data Eng. Bull. **39**,(2016)
3. Batini, C., Scannapieco, M.: Data quality issues in data integration systems. In: Data and Information Quality, Data-Centric Systems and Applications, pp. 279–307. Springer (2016). https://doi.org/10.1007/978-3-319-24106-7_10
4. Becker, C., Duretec, K.: Free benchmark corpora for preservation experiments: using model-driven engineering to generate data sets. In: Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, pp. 349–358. ACM, New York (2013). https://doi.org/10.1145/2467696.2467719
5. Becker, C., Duretec, K., Rauber, A.: The challenge of test data quality in data processing. J. Data Inf. Qual. **8**(2), 1–4 (2017). https://doi.org/10.1145/3012004
6. Becker, C., Faria, L., Duretec, K.: Scalable decision support for digital preservation. OCLC Syst. Serv. Int. Digit. Libr. Perspect. **30**(4), 249–284 . https://doi.org/10.1108/OCLC-06-2014-0025

7. Becker, C., Faria, L., Duretec, K.: Scalable decision support for digital preservation: an assessment. OCLC Syst. Serv. Int. Digit. Libr. Perspect. **31**(1), 11–34 (2015). https://doi.org/10.1108/OCLC-06-2014-0026
8. Birch, J.B., Tukey, J.W.: Exploratory Data. Analysis Behavioral Science: Quantitative Methods. Addison-Wesley, Reading (1978). https://doi.org/10.2307/2286300
9. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. **41**(1), 1–41 (2008). https://doi.org/10.1145/1456650.1456651
10. Brody, T., Carr, L., Hey, J.M.N., Brown, A., Hitchcock, S.: PRONOM-ROAR: adding format profiles to a repository registry to inform preservation services. Int. J. Digit. Curation **2**(2), 3–19 (2008). https://doi.org/10.2218/ijdc.v2i2.25
11. Cochrane, E.: Rendering matters. Report on the results of research into digital object rendering. Archives New Zealand (2012). http://archives.govt.nz/rendering-matters-report-results-research-digital-object-rendering
12. Dong, X.L., Naumann, F.: Data fusion. Proc. VLDB Endow. **2**(2), 1654–1655 (2009). (https://doi.org/10.14778/1687553.1687620)
13. Duretec, K., Kulmukhametov, A., Rauber, A., Becker, C.: Benchmarks for digital preservation tools. In: Proceedings of the International Conference on Digital Preservation (IPRES 2015), Chapel Hill, NC, USA (2015)
14. Esteva, M., Xu, W., Jain, S.D., Lee, J.L., Martin, W.K.: Assessing the preservation condition of large and heterogeneous electronic records collections with visualization. Int. J. Digit. Curation **6**(1), 45–57 (2011). https://doi.org/10.2218/ijdc.v6i1.171
15. Ferro, N., Silvello, G., Buelinckx, E., Doubrov, B., Fresa, A., Geber, M., Jadeglans, K., Justrell, B., Lemmens, B., Martinez, J., et al.: Evaluation of conformance checkers for long-term preservation of multimedia documents. In: JCDL, pp. 145–154 (2018)
16. Garfinkel, S.: Lessons learned writing digital forensics tools and managing a 30tb digital evidence corpus. Digit. Investig. **9**, 80–89 (2012). https://doi.org/10.1016/j.diin.2012.05.002
17. Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G.: Bringing science to digital forensics with standardized forensic corpora. Digit. Investig. **6**, 2–11 (2009)
18. Heslop, H., Davis, S., Wilson, A.: An approach to the preservation of digital records. Technical report (2002). http://trove.nla.gov.au/version/38579727
19. Holden, M.: Preserving the web archive for future generations. In: The Memory of the World in the Digital Age: Digitization and Preservation, pp. 783–795. United Nations Educational, Scientific and Cultural Organization, Vancouver (2012)
20. Hutchins, M.: Testing Software Tools of Potential Interest for Digital Preservation Activities at the National Library of Australia. National Library of Australia, Staff Papers (2012)
21. van der Knijff, J., Wilson, C.: Evaluation of characterisation tools. Part 1: Identification. {SCAPE}{Deliverable} 270137 (2011)
22. Kulmukhametov, A.: Experiment results on conflict resolution in digital preservation (2020). https://doi.org/10.6084/m9.figshare.5877153.v2
23. Kulmukhametov, A., Becker, C.: Content Profiling for Preservation: Improving Scale, Depth and Quality, pp. 1–11. Springer, Berlin (2014). https://doi.org/10.1007/978-3-319-12823-8-1
24. Kumaravel, H.V., Dearborn, C., Witt, M., Kuang, Y.: Measuring the Accuracy of file format identification tools. In: Proceedings of the 11th International Digital Curation Conference. DCC, Amsterdam (2016)
25. Lee, D.: Practical maintenance of evolving metadata for digital preservation: algorithmic solution and system support. Int. J. Digit. Libr. **6**, 313–326 (2007). https://doi.org/10.1007/s00799-007-0014-9
26. Müller, H., Freytag, J.C., Leser, U.: Improving data quality by source analysis. J. Data Inf. Qual. **2**(4), 1–38 (2012). https://doi.org/10.1145/2107536.2107538
27. Naumann, F., Häussler, M.: Declarative data merging with conflict resolution. In: Proceedings of the International Conference on Information Quality (IQ 2002), pp. 212–224 (2002). https://doi.org/10.18452/9206
28. NDSA: 2015 National Agenda for Digital Stewardship. National Digital Stewardship Alliance (2014). http://hdl.loc.gov/loc.gdc/lcpub.2013655119.1
29. Reimer, N., Moldrup, P.: A case study: content profiling using C3PO. Technical report, SCAPE project (2013)
30. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: The Craft of Information Visualization, pp. 364–371. Elsevier, Washington (2003). https://doi.org/10.1016/B978-155860915-0/50046-9
31. Tarrant, D., Carr, L.: LDS3: applying digital preservation principals to linked data systems. In: Proceedings of the International Conference on Digital Preservation (IPRES) (2012)
32. Thibodeau, K.: Overview of technological approaches to digital preservation and challenges in coming years, pp. 4–31. Council on Library and Information Resources, Washington, DC (2002)
33. Xu, W., Esteva, M., Jain, S.D.: Visualizing personal digital collections. In: Hunter, J., Lagoze, C., Giles, C.L., Li, Y.F. (eds.) Proceedings of the 10th Annual Joint Conference on Digital Libraries - JCDL '10, p. 169. ACM Press, New York (2010). https://doi.org/10.1145/1816123.1816147
34. Xu, W., Esteva, M., Jain, S.D., Jain, V.: Analysis of large digital collections with interactive visualization. In: Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 241–250. IEEE (2011). https://doi.org/10.1109/VAST.2011.6102462
35. Xu, W., Esteva, M., Jain, S.D., Jain, V.: Interactive visualization for curatorial analysis of large digital collection. Inf. Vis. **13**(2), 159–183 (2014). https://doi.org/10.1177/1473871612473590