Bagman – A Tool that Supports Researchers Archiving Their Data

Claus Zinn
Seminar für Sprachwissenschaft
Universität of Tübingen, Germany
claus.zinn@uni-tuebingen.de

Abstract

Getting researchers to archive their data properly is hard. Many factors are at play. In this paper, we present *Bagman*, a software that aims at alleviating research data management significantly. Bagman is a web-based software that supports researchers to package their data, assign a minimal set of metadata for their description, define a licence for the data's future distribution, and to submit the entire package in a safe manner to an archive of their choice.

1 Motivation

Research data management is an essential ingredient of good scientific practise. Theories explain the data, and for one researcher to validate another researcher's theoretic models, the inspection of data is central. Nevertheless, many researchers regard the management of research data as a necessary evil. Although one clearly acknowledges the benefits of proper research data management, it is also perceived as something that is not done with overwhelming desire or pleasure.

Fear of scientific scrutiny and competition aside, proper research data management feels like household chores; one needs to make an inventory of all research data, clean-up the data, iron-out a proper file and directory structure of all data, document the procedures and scripts for data annotation and analysis *etc.* When everything is in order, one needs to describe the data with metadata, and then bundle and safely transfer it to an archive of one's choice, so that eventually – once it is ingested into the archive and published – fellow researchers can find and make proper use of it.

The assignment of metadata is a particular nuisance. For this, researchers have to become familiar with metadata standards, registries, profiles, editors, validators, and best practises. Moreover, researchers are expected to take care of licensing issues, and last but not least, know about archives that are well suited to host their precious data.

Our new software, *Bagman*, aims at supporting researchers in all of the aforementioned areas to ease their pain as much as possible. At the same time, the *Bagman* developers strive to improve overall metadata quality, and also support archive managers to receive properly packaged research data.

2 Background

Getting your research data archived constitutes a workflow that varies across institutions. Details aside, it includes data packaging, metadata description, and transfer. Each step is accompanied by some quality control to minimize mishaps in these processes.

2.1 Packaging

In the worst case, researchers send their archive managers an email where all data is attached to the email. Sometimes data is put into some cloud space, or on portable storage devices for manual delivery. Such worst case scenarios often include data loss, files whose formats do not comply with archiving standards or whose names disobey naming conventions. Moreover, metadata descriptions might be anything from absent, incomplete or invalid XML. To avoid such mishaps, the art of packaging needs appreciation.

There are a number of tools that help researchers to bundle their research data into a single package. The open source software `docuteam packer` [URL-1] helps users bundling research data into a single package that can then be transferred to archives (Docuteam, 2018). The stand-alone Java application turns files into a *Submission Information Package* (SIP), a single data package that is delivered

Claus Zinn 2022. Bagman – A Tool that Supports Researchers Archiving Their Data. *Selected papers from the CLARIN Annual Conference 2021. Ed. by Monica Monachini and Maria Eskevich.* Linköping Electronic Conference Proceedings 189, pp. 181–189. DOI: https://doi.org/10.3384/9789179294441

to an archive or repository for (semi-)automatic ingestion, and which contains technical, structural and descriptive metadata in METS, PREMIS and EAD (`www.loc.gov/[mets|premis|ead]`).

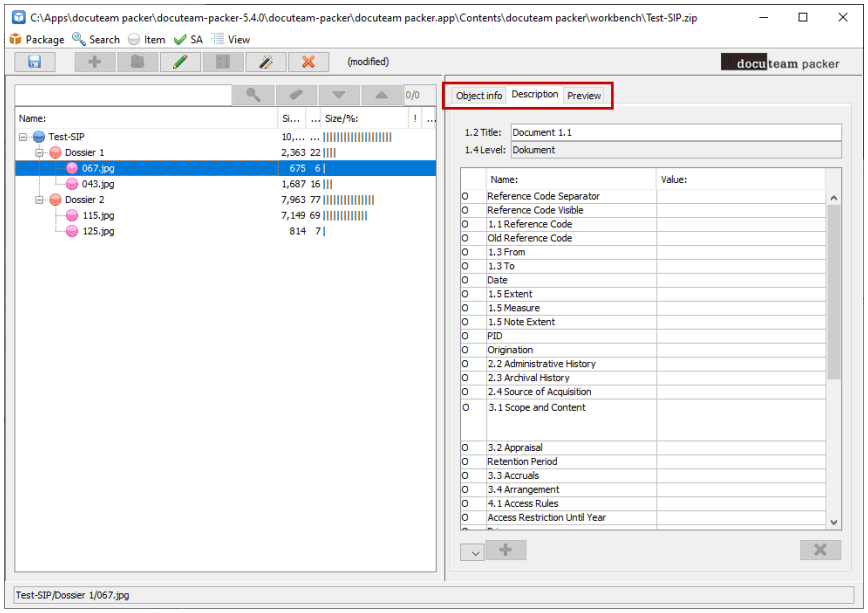

Figure 1: Software docuteam packer.

Fig. 1 depicts how users of Docuteam packer can add their research data in an incremental manner to the SIP. New files can be added to the file tree, and parts of the tree can be rearranged; also, for each object in the tree, metadata can be assigned. Usually, both researchers and archive managers will use the software. Researchers will use it to organize their research data into a tree, and to assign metadata to it to the best of their knowledge; then archive managers will use the software to complement metadata where it is missing.

```
myfirstbag/
|
|   manifest-md5.txt
|    (49afbd86a1ca9f34b677a3f09655eae9 data/27613-h/images/q172.png)
|    (408ad21d50cef31da4df6d9ed81b01a7 data/27613-h/images/q172.txt)
|
|   bagit.txt
|    (BagIt-version: 1.0                                              )
|    (Tag-File-Character-Encoding: UTF-8                              )
|
\--- data/
     |
     |   27613-h/images/q172.png
     |    (... image bytes ...                                        )
     |
     |   27613-h/images/q172.txt
     |    (... OCR text ...                                           )
     ....
```

Figure 2: A simple bag.

A software called `Bagger` was created for the U.S. Library of Congress as a tool [URL-2] to produce a package of data files according to the BagIt specification (Kunze et al., 2018). The specification is a set of hierarchical file layout conventions for storage and transfer of arbitrary digital content. Simply speaking, it can be seen as a shopping cart (bag) together with a shopping bill that lists each of the items with its location (path) and its price (an MD5 or SHA checksum). Those who receive the bag can use the inventory to check whether all goods were received in a complete and correct manner.

A simple example bag is given in Fig. 2. The figure shows the bag's manifest file (the shopping bill) together with the inventory listed under data as well as some technical metadata about the BagIt version used and the file encoding.

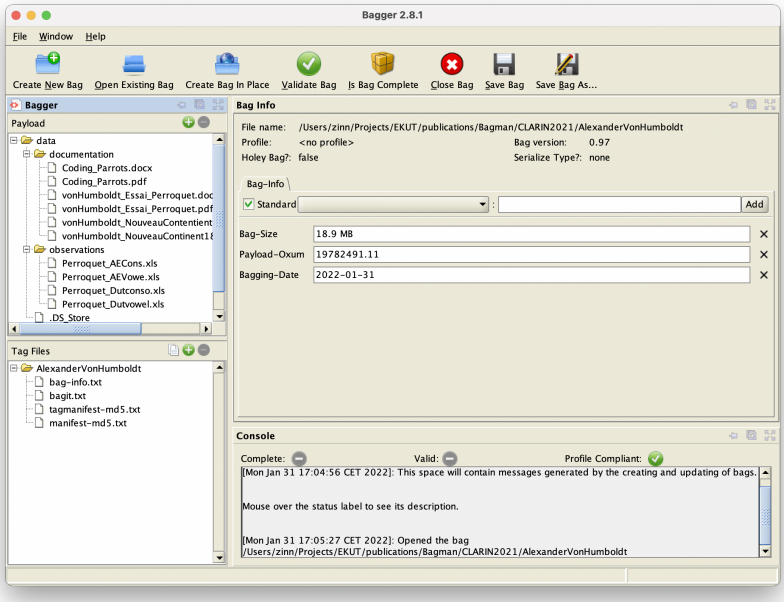

Figure 3: The Bagger Tool from the Library of Congress on our example data.

Fig. 3 shows the Bagger tool, a Java-based and desktop-bound application, in action. The functionality "Create Bag In Place" transforms a given data location on the user's hard drive into a bag that is conform to the BagIt specification. That is, Bagger moves all research data into a subdirectory called data, computes the checksum for each file, and generates the file "manifest-md5.txt" along with the other tag files. In principle, users could be asked to install the Bagger application onto their local machine, create a bag, compress the bag into a zip archive, and then send it to their archive manager; the archive manager on the receiving end then unzips the archive and then uses the Bagger application to validate the bag.

Our software, Bagman, makes use of the BagIt format to help CLARIN researchers packing-up their research data so that it can be transferred to an archive in a correct and complete manner. Similar to docuteam packer, users are given the opportunity to describe their research data with metadata. Rather than asking users to fill out rather technical tables (see right-hand side of Fig. 1), Bagman aims to provide a more user-friendly approach by avoiding metadata jargon.

2.2 Metadata

Metadata plays a key role in any research infrastructure. Good metadata ensures that research data or any digital object are Findable, Accessible, Interoperable and Reproducible. The area of metadata research and practise is vast with many hundreds of metadata standards in use, and hundreds of policies in place to ensure that the FAIR principles are being followed [URL-3]. To support reproducible computational research, metadata formats must be sufficiently expressive to describe input (raw data, intermediate data), tools to process such data (with their version, dependencies, licence *etc.*), statistical reports and notebooks (*e.g.*, session variables, parameters), pipelines (dependencies between tools, provenance), and the resulting scientific publication (research domain, keywords, attribution *etc.*), see (Leipzig et al., 2021) for an overview.

In the CLARIN community, for researchers to assign metadata to data, they need to make use of the CMDI metadata framework (Broeder et al., 2012). For many researchers, this exercise feels like taming

multi-headed monsters in a landscape that feels rough and bracketed from every angle. Researchers need to consult the CMDI component registry [URL-4] to find a metadata profile that best fits their research data, and once they have identified a profile, they have to instantiate it to the best of their knowledge. This is not a trivial matter given that there are hundreds of profiles to choose from, but not a single metadata editor that gives intelligent help with instantiating the numerous different metadata fields.

No wonder, most CMDI-based descriptions have a rather poor descriptive power, taming the beast is exhaustive, and at some point one rather leaves it alone. As a result, researchers must be supported by dedicated archive management staff that is knowledgeable about the CMDI zoo of beasts, and that is armed with XML magic, best practises, and metadata processing tools to keep them at bay.

In Bagman, users are kept away from editing CMDI content directly. Information is gathered via simple forms, and information stemming from bagged resources is automatically added to the CMDI description. As a result, Bagman users are empowered to provide administrative, descriptive, and technical metadata with ease and minimal effort.

2.3 Archiving

The CLARIN infrastructure offers its community members a good number of repositories to store, preserve, and make available to others their research data. The CLARIN Virtual Language Observatory lists nearly 50 different data providers that host over 800 collections of valuable language-related resources. Finding the right archive for your research data is by means trivial when your home institution fails to provide an archive that fits your needs such as content fit or certification requirements, see [URL-8].

The German CLARIN website offers a “find your archive” service that helps researchers identifying the archive that is best suited to host their data [URL-5]. Users are requested to answer questions about the modality of their research data (spoken language, written language, multi-modal language, sign language), its lingual type (German, multi-lingual, historical *etc*), the type of their resource (*e.g.*, lexicon, corpus, treebank), and whether they choose a public licence or not. As a result, the centres that fit the answers best are returned, together with the contact details of the respective archive managers.

Bagman will use the information submitted by the user to suggest archives that are suitable for hosting the user’s research data. Once the user selected the archive, the bag will be safely transferred to a neutral place; the archive manager can download the bag from there, inspect the package, and then contact the user to proceed with the archiving procedure. Bagman hence aims at acting as a broker between researcher and archive manager.

3 Bagman

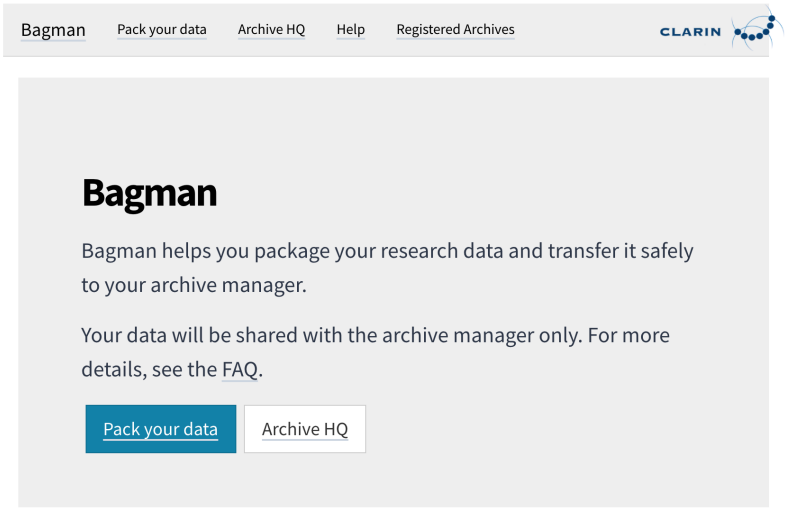

Figure 4: Bagman - Welcome Page.

Bagman aims at supporting researchers and archive managers alike. The software uses `Java` for the back-end and `react-js` for the front-end. Fig. 4 depicts the welcome page of Bagman; it gives access to its two core functionalities: “Pack your data” and “Archive HQ”. The first functionality is targeted at researchers who want to archive their research data; the second one is aimed at archive managers to get access to the research data packages submitted by users. In this paper, we will focus on the first aspect. Fig. 5 depicts Bagman’s user interface for collecting data from its users via simple forms.

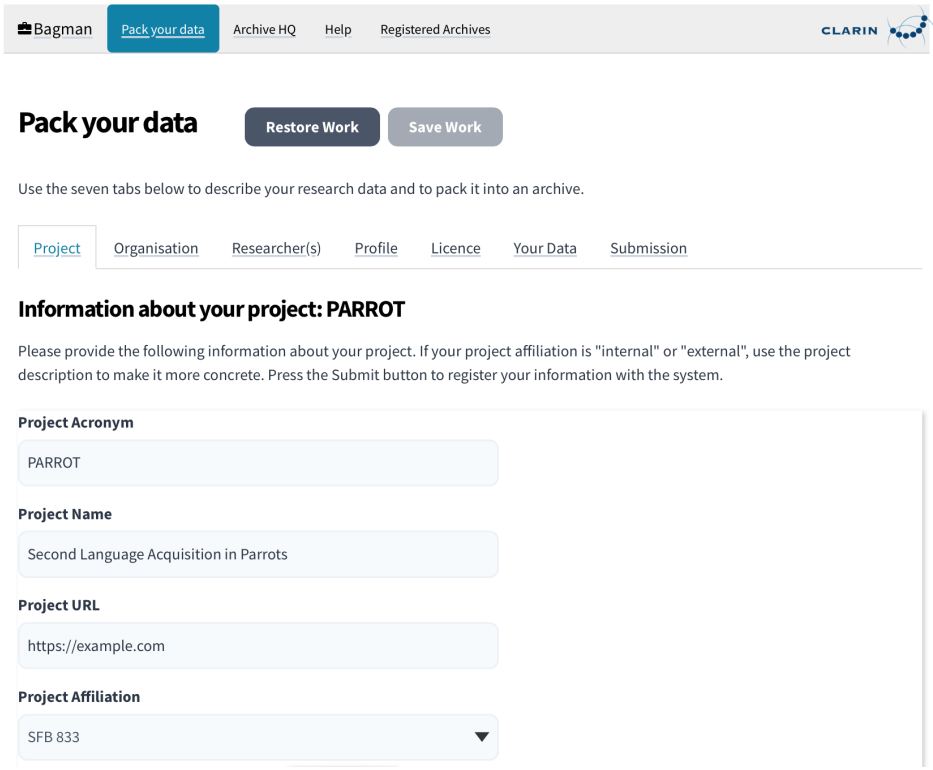

Figure 5: Bagman - Requesting Metadata.

Researchers are requested to describe their research data with respect to the project where it has been collected and the researchers and their organisations that were involved. Users then classify their data in terms of a resource type and by answering a number of targeted follow-up questions about the chosen type. In the fifth step, users can select a licence for their research data. In the sixth step, users can upload their data by selecting a directory from their file system, see top-left part of Fig. 6. Note that some icons in the resulting tree are highlighted in red to signal file formats not suitable for archiving. Here, users are encouraged to convert, say, proprietary file formats to non-proprietary ones, or to delete superfluous ones. Note that Bagman delegates the main task for organising directory structures to users’ existing tools such as Finder (Mac OS), Explorer (Windows), or Files (Ubuntu), and file conversion software, say, Numbers, Excel, or OpenOffice. Once users have post-processed the directory tree, they can prepare the submission process (last step). Preparation includes the *automatic* generation of a CMDI file from known inputs as well as the submission package, the bag where all files all listed together with their checksums (see top-right and bottom part of Fig. 6). The back-end of Bagman takes care of all storage of research data, and it also implements basic functionality for CMDI generation. In detail, the back-end implements an API for (i) the generation of XML-based CMDI from JSON input, which is passed on from the client; (ii) the transferal of bags in ZIP format from client to server as well as methods for getting and deleting bags for archive management. Bagman also implements functionality for matching a bag with an archive that is best suited for hosting it.

Fig. 7 show a fragment of the CMDI file for the component `ResourceProxyList`, which is a

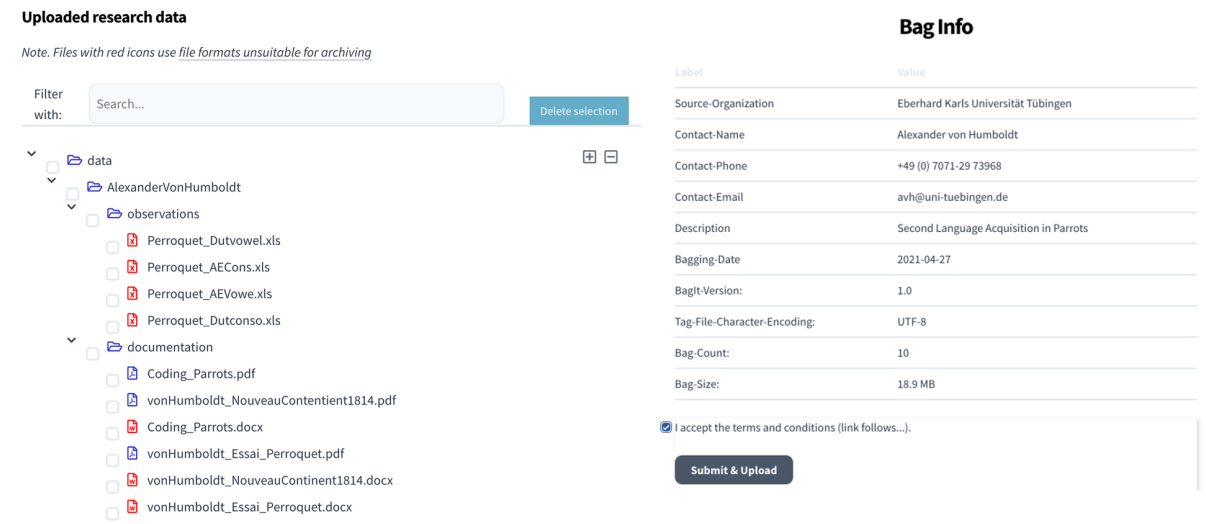

Figure 6: Bagman -Various Screenshots.

central part of the CMDI header. Each resource that our example user has uploaded is tagged as ResourceType “Resource” together with the mimetype that Bagman identified. The `id` attribute of the `ResourceProxy` component assigns a unique `id` to each resource.

Fig. 8 shows the corresponding `ResourceProxyListInfo` component, which reuses the aforementioned unique identifiers. Here, additional information about each resource is given, in particular, the resource’s file size and its checksum in the cryptographic encodings “md5”, “sha1”, and “sha256”.

4 Current State and Future Work

We have built a prototype of Bagman that implements its core functionality and which is now open for beta testing at the website `https://weblicht.sfs.uni-tuebingen.de/bagman/`. We invite readers to explore the tool and encourage their feedback. At the time of writing, only a single archive has been connected to Bagman to test and validate the transfer of data between researchers and archive managers. With research data temporarily stored on Bagman’s back-end, adding new archives to Bagman means giving their managers a login so that they can get access to the bags submitted to them. At the time of writing, Bagman supports the five major resource types hosted by TALAR, the Tübingen Archive of Language Resources [URL-6]; our software hence allows the automatic instantiation of CMDI profiles for the description of lexical resources (LexicalResourceProfile), text corpora (TextCorpusProfile), speech corpora (SpeechCorpusProfile), tools (ToolProfile), and experiments (ExperimentProfile), all identifiable via the Group Name “NaLiDa” in the CMDI component registry. The use of these profiles ensures that the corresponding resources can be easily found using faceted browsing in the Virtual Language Observatory, say, by searching the facets for language, collection, resource type, modality, or availability [URL-7].

The design of Bagman walks a fine line between researchers (often taking research data management

```
<cmd:Resources>
    <cmd:ResourceProxyList>
    <cmd:ResourceProxy id="id-2cf274a2-bed0-4737-86a1-25d353784b68">
            <cmd:ResourceType mimetype="application/vnd.ms-excel">Resource</cmd:ResourceType>
            <cmd:ResourceRef>observations/Perroquet_Dutvowel.xls</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-77030d3c-983f-4394-a5ed-6d4e3202e2fb">
            <cmd:ResourceType mimetype="application/vnd.ms-excel">Resource</cmd:ResourceType>
            <cmd:ResourceRef>observations/Perroquet_Dutconso.xls</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-117f8cfd-4b16-4e51-936f-6075f709d359">
            <cmd:ResourceType mimetype="application/pdf">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/Coding_Parrots.pdf</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-dfb55091-87c8-4b69-9b63-fe299e80b18e">
            <cmd:ResourceType mimetype="application/vnd.ms-excel">Resource</cmd:ResourceType>
            <cmd:ResourceRef>observations/Perroquet_AECons.xls</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-c32cb84e-a621-4391-aba0-1fbedce415f1">
            <cmd:ResourceType mimetype="application/vnd.ms-excel">Resource</cmd:ResourceType>
            <cmd:ResourceRef>observations/Perroquet_AEVowe.xls</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-a8e9053a-11f4-437e-a240-e3ca7348269d">
            <cmd:ResourceType mimetype="application/pdf">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/vonHumboldt_NouveauContentient1814.pdf</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-c06a7a20-31eb-42b9-8f2d-39ef133485bb">
            <cmd:ResourceType mimetype="application/vnd.openxmlformats-officedocument.wordprocessingml.document">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/Coding_Parrots.docx</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-3c188547-668c-4e90-836b-69c2f86fb382">
            <cmd:ResourceType mimetype="application/vnd.openxmlformats-officedocument.wordprocessingml.document">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/vonHumboldt_Essai_Perroquet.docx</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-04c27ac0-08ae-4607-b908-57ace1e203de">
            <cmd:ResourceType mimetype="application/vnd.openxmlformats-officedocument.wordprocessingml.document">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/vonHumboldt_NouveauContinent1814.docx</cmd:ResourceRef>
        </cmd:ResourceProxy><cmd:ResourceProxy id="id-a60ca8ef-8cc6-469a-9b95-b0a0188c9b4b">
            <cmd:ResourceType mimetype="application/pdf">Resource</cmd:ResourceType>
            <cmd:ResourceRef>documentation/vonHumboldt_Essai_Perroquet.pdf</cmd:ResourceRef>
        </cmd:ResourceProxy></cmd:ResourceProxyList>
    <cmd:JournalFileProxyList> [4 lines]
    <cmd:ResourceRelationList> [1 line]
</cmd:Resources>
```

Figure 7: CMDI Excerpt – ResourceProxyList component.

as a necessary evil) and archive managers (taking it for something absolutely necessary, with an emphasis on "the more metadata the better"). When Bagman users, for instance, identify their data as a lexical resource, they are given the opportunity to specify the type of the lexicon (*e.g.*, dictionary, glossary, thesaurus), the type of the headword, and the subject language, but they may skip the step if they want to. Also, they can put more information about their resource in an open-ended lexicon description field when they feel that more information needs to be put somewhere. Note, however, that Bagman delegates any metadata-related issues to a subsequent one-to-one communication between researcher and archive manager. Metadata fields left open during a Bagman session can often be filled at a later stage when archive managers feel they require more information than researchers provided.

Bagman is browser-based software, and hence, special care needs to be taken to ensure that users can provide their input in a flexible, piece-wise manner. At any time, users can save the current session, that is, write-out all metadata that has been entered to their file system. At a later time, when users like to resume their work, they can then easily restore their session.

At the time of writing, Bagman is only connected to TALAR, but it supports all the archive's profiles. For TALAR users, Bagman has entered production mode. The feedback we obtain from these real-world users informs the further development of Bagman, strengthening its usability and stability. Once Bagman has matured, we will ask other archives whether they want to be connected to Bagman, and we will investigate how their archiving requirements can be meet with the software. Currently, it is too soon to speculate about the detailed implementation roadmap for the archiving aspect of Bagman. It is clear that other archives will like to see their metadata profiles and archiving policies supported. Here, Bagman would need to adapt its front-end to collect information specific to the new profiles, and the back-end to generate ready-to-use and valid CMDI instances that other archives are happy to work with.[1]

Bagman does not prescribe any guidelines on the granularity of the research data that needs to be archived. Each set of resources is different, and Bagman *per se* does not attempt to promote a *one-size-fits-all model*.[2] Naturally, there exist research data that are not easily or adequately described with

[1] At the time of writing, Bagman pre-fills some form fields such the organisation's name or address. The default values of such fields are specific for TALAR users, but can be overwritten. With each new archive being connected, Bagman would also need to adapt its front-end to provide default values specific to the archive.

[2] As a rule of thumb, all research data created to support a scientific finding should be bundled into a single archival unit.

```xml
<ResourceProxyListInfo cmd:ComponentId="clarin.eu:cr1:c_1470820607607">
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673"
        cmd:ref="id-2cf274a2-bed0-4737-86a1-25d353784b68">
        <ResProxItemName>Perroquet_Dutvowel.xls</ResProxItemName>
        <ResProxFileName>observations/Perroquet_Dutvowel.xls</ResProxFileName>
        <FileSize>4866560</FileSize>
        <Checksums>
            <md5>e0288dce9a55ffdef4af1e73e650c747</md5>
            <sha1>636a3e8803a09c958994201c3ff1fff5879366dc</sha1>
            <sha256>23f35f38af09f1868e327b978d942dac6d5deaa345690881cfc60be59bf81267</sha256>
        </Checksums>
    </ResourceProxyInfo>
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673"
        cmd:ref="id-77030d3c-983f-4394-a5ed-6d4e3202e2fb">
        <ResProxItemName>Perroquet_Dutconso.xls</ResProxItemName>
        <ResProxFileName>observations/Perroquet_Dutconso.xls</ResProxFileName>
        <FileSize>4871680</FileSize>
        <Checksums>
            <md5>c614c88a495920aebecd29594074be40</md5>
            <sha1>aecbe906280a8b077cdc1a5ebc06c7393122e195</sha1>
            <sha256>7207f1cdf331e8435ebcb72271addacd5c83ae25ca1b40a5f8048a9ff4c08413</sha256>
        </Checksums>
    </ResourceProxyInfo>
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
    <ResourceProxyInfo cmd:ComponentId="clarin.eu:cr1:c_1361876010673" [10 lines]
</ResourceProxyListInfo>
```


Figure 8: CMDI Excerpt – ResourceProxyListInfo component.

Bagman and the CMDI profiles it currently supports. While the TALAR-based CMDI profiles have a good descriptional power, they might have shortcomings when it comes to the description of heterogeneous research data that researchers see as a single archival unit. For now, most users are unaware of Bagman. They contact the TALAR archive manager because they would like to have their resources deposited. Once the contact has been established and any open questions between the two parties addressed (*e.g.*, granularity or licence issues), the users are then explicitly directed to Bagman to build, describe, and submit their package to the archive via Bagman.

One important aspect to Bagman's usability is the packaging. When the archive manager is informed of a new bag being submitted via Bagman, he can download the bag from Bagman's "Archive HQ" GUI, unzip the bag and run BagIt software to verify that the package has been transferred in a complete and correct manner.[3] Our TALAR archive managers find this functionality very useful and reassuring indeed, and a necessary first step before looking into the CMDI, and contacting the researchers for any follow-ups, such as resolving metadata issues, or the drafting and signing of data depositing agreements.

Note that the use of the BagIt specification duplicates information that is also present in the CMDI file generated by Bagman, in particular, the information shown in Fig. 7 and Fig. 8. The duplication of such technical metadata, however, is well justified. The bag delivered to the archive is used to ensure that all research data is being transferred in a complete and correct manner. and archive managers can use the aforementioned toolchain to validate the bag. The CMDI file, of course, is used by metadata harvesters such as the VLO to being able to link to the resources the metadata describes.

Getting users to archive their research data is hard. Bagman offers users a single pit-stop approach to get their data archived without too much hassle. Bagman helps users with metadata management as it generates a CMDI automatically from the information and research data supplied by the user. Given such

[3]The command `python3 -m bagit --validate bag` verifies the bag.

data, Bagman then helps users to decide on an archive to host their resource, and then helps ensuring that all data is transferred to the archive in a complete and correct manner. In sum, Bagman fills-in a gap in the CLARIN infrastructure; its ease of use encourages users to get their data archived; and its automatic generation of CMDI from known inputs ensures the generation of expressive and high-quality metadata.

Acknowledgements

Our work was funded by the German Federal Ministry of Education and Research, the German Science Foundation (SFB-833), and CLARIN-D.

References

Broeder, D., Windhouwer, M., Van Uytvanck, D., Goosen, T., and Trippel, T. 2012. CMDI: a Component Meta-Data Infrastructure. In *Describing LRs with metadata: towards flexibility and interoperability in the documentation of LR. Workshop at LREC-2012*, Istanbul, Turkey.

Docuteam. 2018. Software – our tools for digital archives. Available at `https://www.docuteam.ch/en/products/it-for-archives/software/`.

Kunze, J., Littman, J., Madden, E., Scancella, J., and Adams, C. 2018. The bagit file packaging format (v1.0). Technical report, RFC 8493, DOI 10.17487/RFC8493, October. See `https://www.rfc-editor.org/info/rfc8493`.

Leipzig, J., Nüst, D., Hoyt, C. T., Ram, K., and Greenberg, J. 2021. The role of metadata in reproducible computational research. patterns. *Patterns*, 2(9). https://doi.org/10.1016/j.patter.2021.100322.

Links

[URL-1] Docuteam packer, see `https://docs.docuteam.ch/packer/6.1/en/index`.
[URL-2] Bagger, see `https://docs.docuteam.ch/packer/6.1/en/index`.
[URL-3] The FAIR principles, see `https://fairsharing.org`.
[URL-4] Component registry, see `https://catalog.clarin.eu/ds/ComponentRegistry`.
[URL-5] Centre finder, see `https://www.clarin-d.net/en/preparation/find-a-clarin-centre`.
[URL-6] TALAR, see `https://talar.sfb833.uni-tuebingen.de`.
[URL-7] Virtual Language Observatory, see `https://vlo.clarin.eu`.
[URL-8] Core Trust Seal, see `https://www.coretrustseal.org`.