Engineering a Software Environment for Research Data Management of Microscopy Image Data in a Core Facility

Dissertation

Presented to the Department of Biology/Chemistry,
Osnabrück University
in partial fulfilment of the requirements for the degree of
'Doctor Rerum Naturalium'

Susanne Kunis

Osnabrück, October 2021

First Reviewer:	Prof. Dr. Jacob Piehler
Second Reviewer:	Prof. Dr. Elisa May
Date of Disputation:	17.02.2022

Contents

List of Figures

List of Listings

List of Tables

Zusammenfassung

Diese Dissertation befasst sich mit Konzepten und Lösungen bezüglich des Datenmanagements im wissenschaftlichen Alltag für mikroskopische Bilddaten aus der Biologie. Der Schwerpunkt der formulierten Anforderungen lag bisher auf publizierten Daten, die nur eine kleine Teilmenge der im wissenschaftlichen Prozess erzeugten Daten darstellen. Mehr und mehr rücken die Daten des Forschungsalltags in den Fokus der schon früh formulierten Prinzipien für das Management von Forschungsdaten (FAIR-Prinzipien). Dabei ist die adäquate Verwaltung dieser zumeist multimodalen Daten hinsichtlich der Heterogenität und Umfanges eine große Herausforderung. Es fehlt an standardisierten und etablierten Arbeitsabläufen und auch die bisher verfügbaren Softwarelösungen bilden die besonderen Anforderungen dieses Bereiches nur ungenügend ab. Der Erfolg jedes Datenmanagementprozesses hängt jedoch stark vom Grad der Integration in den Arbeitsalltag ab. Das Datenmanagement muss sich, soweit möglich, nahtlos in diesen Prozess einfügen.

Die Mikroskopiedaten im wissenschaftlichen Prozess sind eingebettet in ein Preprozessing, das aus vorbereitenden Laborarbeiten besteht und der analytischem Auswertung der Mikroskopiedaten. Die Bilddaten bilden in ihrem Volumen oft den größten Anteil an Daten, die innerhalb dieses gesamten Forschungsprozesses erzeugt werden. In dieser Arbeit konzentrieren wir uns auf Konzepte und Techniken bezüglich der Handhabung und der Beschreibung dieser Bilddaten und befassen uns mit den dazu nötigen Grundlagen. Ziel ist die verbesserte Einbettung der vorhandenen Datenmanagementlösung für Bilddaten (OMERO) in den wissenschaftlichen Arbeitsalltag. Dazu wurden im Rahmen dieser Dissertation zwei unabhängige Softwareerweiterungen für OMERO implementiert: OpenLink and MDEmic. OpenLink vereinfacht den Zugriff auf die im integrierten Repositorium gespeicherten Daten, um diese für weitere Auswertungen etablierten Workflows zuzuführen und ermöglicht neben dem internen auch den externe Austausch von Daten, ohne die Vorteile des Datenrepositoriums abzuschwächen. Der Schwerpunkt der zweiten implementierten Softwarelösung, MDEmic, liegt auf der Erfassung von relevanten Metadaten für die Mikroskopie. Durch die erweiterte Metadatenerfassung wird eine entsprechende Verlinkung der multimodalen Daten mittels eindeutige Beschreibung und dem entsprechenden semantischen Hintergrund angestrebt. Die Konfigurierbarkeit von MDEmic ist darauf ausgerichtet, die zurzeit sehr dynamische Entwicklung zugrundeliegender Konzepte und Formate Rechnung tragen zu können. Hauptziel von MDEmic ist den Arbeitsaufwand zu minimieren und Prozesse zu automatisieren. Damit steht dem Wissenschaftler ein Werkzeug zur Verfügung, um diese komplexe und umfangreiche Aufgabe der Metadatenerfassung für mikroskopische Daten in einfacher Form zu bewältigen. Mit Hilfe der Software kann eine semantische und syntaktische Standardisierung erfol-

gen, ohne dass sich der Wissenschaftler mit den technischen Konzepten auseinandersetzen muss. Die generierten Metadatenbeschreibungen werden automatisch in das Bildrepositorium integriert und können gleichzeitig von den Wissenschaftlern in Formate übertragen werden, die bei der Publikation der Daten benötigt werden.

Abstract

This thesis deals with concepts and solutions in the field of data management in everyday scientific life for image data from microscopy. The focus of the formulated requirements has so far been on published data, which represent only a small subset of the data generated in the scientific process. More and more, everyday research data are moving into the focus of the principles for the management of research data that were formulated early on (FAIR-principles). The adequate management of this mostly multimodal data is a real challenge in terms of its heterogeneity and scope. There is a lack of standardised and established workflows and also the software solutions available so far do not adequately reflect the special requirements of this area. However, the success of any data management process depends heavily on the degree of integration into the daily work routine. Data management must, as far as possible, fit seamlessly into this process.

Microscopy data in the scientific process is embedded in pre-processing, which consists of preparatory laboratory work and the analytical evaluation of the microscopy data. In terms of volume, the image data often form the largest part of data generated within this entire research process. In this paper, we focus on concepts and techniques related to the handling and description of this image data and address the necessary basics. The aim is to improve the embedding of the existing data management solution for image data (OMERO) into the everyday scientific work. For this purpose, two independent software extensions for OMERO were implemented within the framework of this thesis: OpenLink and MDEmic. OpenLink simplifies the access to the data stored in the integrated repository in order to feed them into established workflows for further evaluations and enables not only the internal but also the external exchange of data without weakening the advantages of the data repository. The focus of the second implemented software solution, MDEmic, is on the capturing of relevant metadata for microscopy. Through the extended metadata collection, a corresponding linking of the multimodal data by means of a unique description and the corresponding semantic background is aimed at. The configurability of MDEmic is designed to address the currently very dynamic development of underlying concepts and formats. The main goal of MDEmic is to minimise the workload and to automate processes. This provides the scientist with a tool to handle this complex and extensive task of metadata acquisition for microscopic data in a simple way. With the help of the software, semantic and syntactic standardisation can take place without the scientist having to deal with the technical concepts. The generated metadata descriptions are automatically integrated into the image repository and, at the same time, can be transferred by the scientists into formats that are needed when publishing the data.

1 Introduction

Biological experiments generate a large amount of heterogeneous and interdependent data. The representation and comprehensibility of this correlation significantly depends on the structuring of the data and their documentation. Since a variety of protocols, tools, data formats, and context-specific parameters influence the data in the course of the experimental process, this results in a complex challenge of data management, which must cover and combine many aspects in order to ensure the reproducibility of the study. To simplify this process, many models, guidelines, and standards have been developed in recent times. However, the process of managing research data is constantly evolving and also integrates newer cutting-edge technologies and concepts to address the complexity of biological experiments. Suitable software solutions support the biological researcher in the application of these concepts, so that research data management can be integrated more and more naturally into everyday experimental work. Nevertheless, the requirements of the different research groups are so heterogeneous that further adaptation of the software and workflows to everyday scientific life is necessary.

1.1. Motivation and Scientific Environment

This thesis deals with components of research data management with regard to microscopy image data that are generated within the framework of scientific studies in an interdisciplinary centre in the associated imaging facility. These data require the coherent documentation of both the technical aspects of image acquisition and the corresponding documentation of further experimentally relevant information in order to make them usable for other participating scientists. One focus of this thesis is on the collection of this metadata directly after the acquisition process at the microscope and the integration of the metadata into the existing management structure for microscopy data. Moreover, a simplified access to the image data integrated in this central management structure is an important aspect. Research in an interdisciplinary centre requires working with data across departments and sites. The amount and size of the respective data and the increased transfer volume due to the heterogeneous decentralised software landscape in the research groups for data post-processing and analysis requires adapted solutions for effective data exchange.

1.2. Specific Objectives

The components of the research data management workflow generated in the course of this thesis with regard to the experimental data are intended to keep the overhead of work for the researcher as low as possible.

1. **Data access for post-processing and data sharing:** Decentralised software solutions are often used for post-processing and further analysis of microscopy data. On the one hand, this is due to the diversity of the microscopy data itself, as the proprietary file formats can often only be used for analysis with special software. On the other hand, modern research approaches also require the use of specialised post-processing solutions that are developed locally. Due to the size of the data, centralised solutions are strived for, but do not yet sufficiently cover all requirements or are not feasible, e.g., for licensing reasons. The goal is to enable fast data access without undermining the benefits of the research data management solution in terms of data integrity. To simplify the data exchange itself and increase collaboration, a solution that also enables global data access is preferred. The need and priority of this aspect was also highlighted by the 2020 pandemic-related constraints.

2. **Gathering relevant metadata for microscopy experiments:** Documenting the experimental process generates additional work, but is essential for the quality of the data. Data that is not documented leads to misinterpretations that can affect the entire ongoing research process. However, the additional documentation effort is an obstacle, so this step is often postponed to the end of the experimental process. A suitable tool will minimise this effort and integrate it as early as possible in the experimental process. At the same time, it should promote standardised metadata to the extent that structures and applications based on it facilitate networking and the search for data correlations. This increases the efficiency and effectivity of the research work.

1.3. Contributions and Impact

In the context of this thesis, the established data management software `OMERO` was extended by functional components for faster data download and exchange and made available to the users. This functionality has also been made available to the community as open-source software in the publicly accessible repository `GitHub` [1]. Furthermore, a software tool to support the collection of metadata on microscopy experiments has been developed and integrated into the established microscopy data management software. This local integration has been officially adopted into the existing software by the maintainers of the data management software and is available to the community as standard release [2].

The tool for metadata annotation was published with a corresponding use-case [3] and influenced the development of further software solutions in this field [4].

Moreover, I have contributed to the community by co-initiating the formation of the Research Data Management for microscopy (RDM4mic) working group in 2019. I have a key role in this group, which meets regularly to discuss and accelerate further standardisation and the establishment of workflows at national level [5]. Members of the RDM4mic working group initiated a joint research proposal with four applicant institution, including University Osnabrück, within the DFG programme "Information infrastructure for research data" with the goal to establish information infrastructures for bioimaging data (I3D:bio). Funding for this project was granted in July 2021. I also took over the leadership of the German BioImaging (GerBi) society's working group Image Data Analyses and Management in 2020. At the international level, I co-chair the metadata working group of QUAREP-LIMI [6]. Currently, the NFDI4BIOIMAGE [7] consortium is in the application phase, where I am co-applicant as head of the Image (meta)data formats and standardization task area.

1.4. Content of this Thesis

The further document structure is as follows:

- "**Chapter 2 - Basics**" provides the background information in the field of research data management and in concepts and software solutions of a multi-collaborative initiative that are essential for understanding this thesis.
- "**Chapter 3 - Engineering Software Support for Data Management for Microscopy**" specifically addresses the research data management needs of an interdisciplinary centre's imaging facility and includes descriptions of two solutions implemented to improve the local research data management environment and workflow.
- "**Chapter 4 - Conclusion**" contains the summary and provides an outlook for the further development of the various components.

2 Basics

The first part of this chapter provides an overview of the current requirements for research data management in a research project in general, as well as established standards and concepts for meeting these requirements. This is followed by the requirements specifically for bioimaging and microscopy data and their metadata. In particular, the requirements and concepts related to metadata will be addressed. In the second part, we will present the results of a collaborative initiative to establish standards for file formats and metadata for microscopy data. This part also describes the underlying already established software components on which the tools we are developing build to help users manage their research data.

2.1. Research Data Management

Research data management includes all activities in terms of preparation, storage, and publication of data during a research project. The scientist must be aware that the whole process of planning, generating, analyzing, storing and sharing data is part of research data management. This includes all data in the different stages of the research process, such as the description of material and products, e.g. cell samples or specially designed primers, but also procedures, such as protocols for sample preparation. However, research data management also includes finding, reusing of data and citing publicly available data [8].

Good data management is characterised by the fact that required information can be found and retrieved at any time. How the data and additional information (metadata) are organised and structured significantly affects the searchability and the possibility of filtering the data effectively. The findability of data and the quality of the data description in turn have an impact on the reproducibility of the data and results and its verification. For optimal organisation and structuring, data management in the research process must be planned in advance. During the planning of the experiment, one should consciously deal with the existing and newly generated data and workflows beforehand, both of the experimental and the analytical workflow [9]. The visualisation of the data flow, based on the data lifecycle model [10], could be very helpful in this case. It illustrates when data is created and how it is to be further processed (Figure 2.1). The devices used, the previous or future storage of the data, as well as the type and location of the data processing and the people involved in each case are important information to consider. The categorisation of

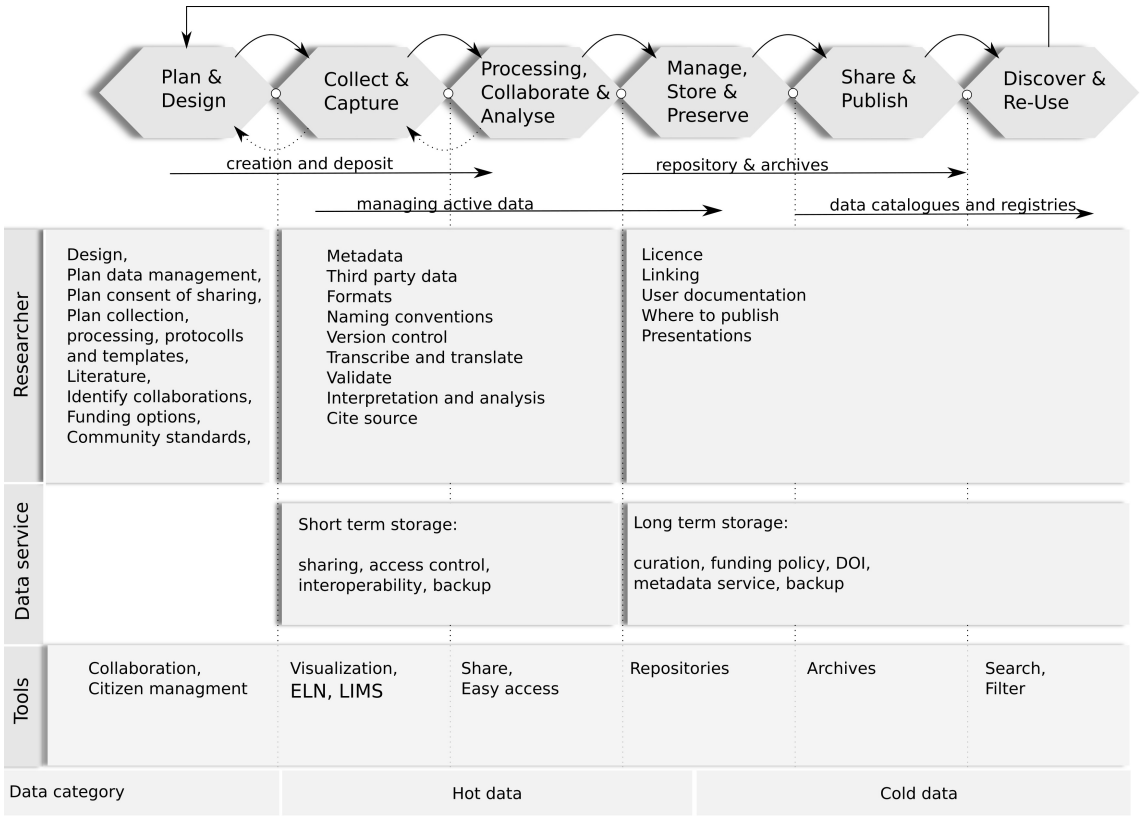

Figure 2.1. Data lifecycle. An overview of responsibilities for researchers and requirements for services and tools. Generated data can be categorised in hot data (red), and cold data (blue). Hot data represents data that is used during the scientific process, for example for analysis and post-processing. Cold data represent data that should remain available in the long term but are not actively used, such as published datasets.

the resulting data and the specification of the required documentation is another important step in research data management in order to make the data creation process and the data itself reproducible. When specifying and defining the data, aspects of machine readability should be taken into account in order to enable subsequent use, e.g. by meta-studies or artificial intelligence algorithms. In order to be able to read and process data automatically by a computer, it is important that the data is available in a structured form and suitable format. A simple example of this is saving a table in a spreadsheet format, like csv instead of a pdf format. Since research data management aims at long-term storage, further considerations are necessary regarding formats and data conversions to ensure the readability of the data in the future. In addition, the software support required for data readability must also be considered for long-term storage. For both data formats and software required for readability, independent open source solutions widely used in the community should be chosen to avoid licensing or availability issues later on.

The main outcome of good data management is to increase of research impact and visibility. Strategic research data management facilitates the extraction of conclu-

sions and testable hypotheses from data. Furthermore a high degree of data re-use fosters innovation and new discipline-specific and cross-disciplinary collaborations. Research as a whole becomes more transparent and traceable, which will also improve the research methods themselves. On the other hand, double work is avoided which leads to a cost reduction during research processes. Finally, the re-usable data can also serve as a basis for education and training.

2.1.1. FAIR Principles

An important keystone of research data management is the formulation of principles for the findability, accessibility, interoperability and re-usability of data - the so-called FAIR principles. These can also be considered and used as guidelines for good data management (Figure 2.2). By implementing these principles, a sustainable

Figure 2.2. Overview of FAIR principles [11].

reuse of research data can be achieved. In the following we will go into more detail on the individual principles [12].

Findable: Here, mechanisms are described or specified that increase the findability of the data. In addition to the machine readability of the data and its existing and descriptive metadata, this also includes the storage of data and metadata in a database or repository in order to guarantee the constant and consistent linking of data and metadata. It also includes the assignment of persistent identifiers such as DOIs.

F1: (Meta)data are assigned a globally unique and eternally persistent identifier.
F2: Data are described with rich metadata.
F3: (Meta)data are registered or indexed in a searchable resource.
F4: Metadata specify the data identifier.

Accessible: Here, conditions are formulated under which access to the data is made possible by humans and machines. Access here means downloading and using the data, i.e. the exchange of the data. This is made possible, for example, by standard communications protocols such as HTTPS.

A1: (Meta)data are retrievable by their identifier using a standardised communications protocol.
A1.1: The protocol is open, free, and universally implementable.
A1.2: The protocol allows for an authentication and authorisation procedure, where necessary.
A2: Metadata are accessible, even when the data are no longer available.

Interoperable: A prerequisite for interoperability is the linkability of the data sets to each other. This is only possible if the data are described by means of ontologies, formulated and stored in machine-readable formats, such as Resource Description Frameworks (RDF, see Section 2.1.4) and and a suitable serialisation format, so that a connection can be established.

I1: (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2: (Meta)data use vocabularies that follow FAIR principles.
I3: (Meta)data include qualified references to other (meta)data.

Re-usable: Re-usability is significantly influenced by the scope and content of the metadata. Subject-specific community standards should be adhered to where they exist, such as for ontologies and formats. The description of the data origin, such as devices or procedures used, is also important for re-usability. Clear licences that grant rights of use in a simple way and prevent licence incompatibilities in combinations of differently licensed data, e.g. from Creative Commons such as CC0 or CC-BY, and persistent identifiers enable easy citation of the data.

R1: Meta(data) have a plurality of accurate and relevant attributes.
R1.1: (Meta)data are released with a clear and accessible data usage license.

R1.2: (Meta)data are associated with their provenance.
R1.3: (Meta)data meet domain-relevant community standards.

In summary, the FAIR principles imply that the quality factors for good data management are standards for formats, technical language used, process protocols, metadata and the application of persistent identifiers. The implementation of the FAIR principles is only made possible by a suitable technical FAIR ecosystem (Figure 2.3), which supports the FAIRification process in a subject-specific way. The

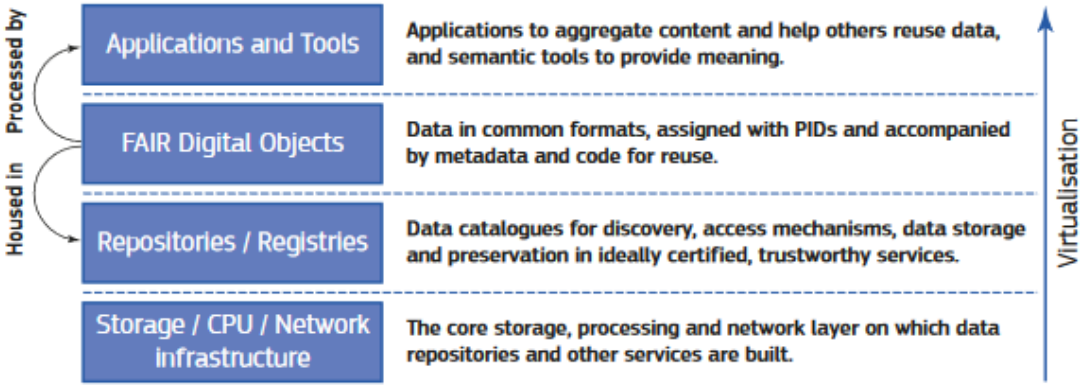

Figure 2.3. FAIR ecosystem. The technical infrastructure layers and increasing degrees of virtualisation (adapted from [13]).

FAIRification process is significantly influenced by the metadata [14]. That means, many of the FAIR principles can be implemented at the metadata level. The scope and content of metadata thus determines the quality of research data management. However, a lack of standardised procedures and software tools in daily work makes the collection of metadata a very labour-intensive task in many areas. The more complex and extensive the metadata and their interrelationships are, the more this delays the FAIRification process.

2.1.2. Standardisation

In order to link the different data of a research project and to put them into a common context, appropriate standards are necessary. Standardisation takes place at and between different levels (Figure 2.4).

At the working group level, coordination among scientists is required concerning data and metadata. In order to maintain the locally agreed upon standard regarding data structuring, formats, metadata, and naming conventions, principles and guidelines similar to those used in experiments must be formulated. Guided workflows and predefined input forms support the scientist in implementing this workgroup standard.

At the institutional level, standards can be specified by data guidelines. Providing the requested software infrastructure at the institutional level, like repositories and electronic lab books, can prevent isolated solutions at the workgroup level. Isolated solutions created by individual working groups themselves are less well networked, have limited accessibility, and are usually not sustainable because expertise on the infrastructure used is lost when the responsible scientists leave the working group. Institute-wide training promotes awareness, acceptance, and ultimately the implementation of standards in research data management.

At the domain-specific or community level, standards need to be developed regarding the vocabulary used in the form of ontologies (see Section 2.1.5) that will be adopted in the levels below. Similarly, standards for formats as well as the data models or schemes used should be defined here. An application and implementation in domain-specific repositories and journals is necessary to create broad acceptance. However, these standards also require continuous development and adaptation by national and international working groups.

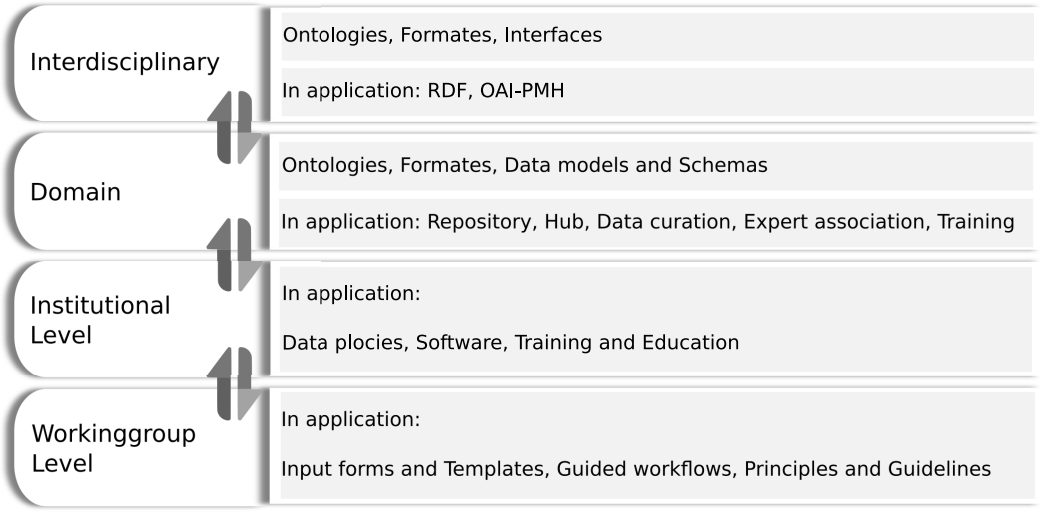

Figure 2.4. Standardisation on different working levels.

The greatest challenge is standardisation in the interdisciplinary field. Here, standards enable the linking of scientific knowledge across disciplines and thus the formation of correlations, the acquisition of new knowledge, which in turn forms the basis for new scientific projects. At the level of interdisciplinary data, we are dealing with so-called Big Data, as they are characterised by a high degree of variance and inconsistencies. To enable exchange, concepts from the Semantic Web are suitable, where finding, sharing, combining and reusing information plays an important role. The relationships between data and objects are described in common data formats to enable machine-supported analysis and linking. All semantic information needed for interpretation is linked accordingly. Such Linked Data concepts (see Section 2.1.4) and formats thus enable the linking of a wide variety of information and the formation of directed knowledge graphs in machine-readable format (see also

Section 2.1.4). The exchange of information is another important area of standardisation. Here, suitable interfaces or structures must be defined and implemented. One such interface is, for example, the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), a simple mechanism for exchanging information between public repositories.

In the following, the standardisation of metadata and formats in relation to bioimaging and microscopy will be addressed in particular.

2.1.3. Research Data Management for Bioimaging and Microscopy

This work focuses specifically on research data management with respect to image data generated by microscopy techniques in biology.

Bioimaging represents techniques for the non-invasive imaging of biological processes and the visualisation of 2D or 3D structures ranging from fixed subcellular material to living multi cellular organisms. The leading method of bioimaging is microscopy. Over the years, microscopy techniques have constantly evolved and, at the same time, the amount of data generated by imaging has steadily increased. The purely descriptive technique has turned into a quantitative -omics like approach. This can generate highly complex multidimensional data of up to several terabytes for a single experiment.

In order to ensure high-performance storage for the various techniques, the number of microscopy file formats, the so-called image data files or image containers, which were designed by the manufacturers, grew along with the number of techniques. These usually contain not only the image data (binary data) itself but also descriptive data regarding the imaging technique used and its components as well as the configuration of these components. To date, there is no uniform standard for container formats or for the original metadata embedded in them. The terminology and specifications as well as the scope of the metadata differ considerably between the respective microscope manufacturers (example in Figure 2.5). These vendor-specific and proprietary file formats make software-based metadata harvesting challenging [15]. Complete insight is only guaranteed by the software provided by the respective microscope manufacturers for the respective format. This in turn inhibits the system-independent comparison and sharing of data and thus also the re-usability and linking of data with each other. At this point, processes are needed that meet the needs of microscope manufacturers for high-performance specialised formats and the needs of users for easy access to data and metadata.

Research data management for biological imaging involves a range of very different data. The data generated by microscopic imaging itself accounts for a significant part. However, data from sample preparation or post-processing as well as software and methods are also part of research data management in this field. The following problems are characteristic of data management for microscopy imaging:

- Enormous heterogeneity in data and technology,
- Increasing complexity of data,

- Growing volume of data,
- A wide range of non-standard, proprietary file formats.

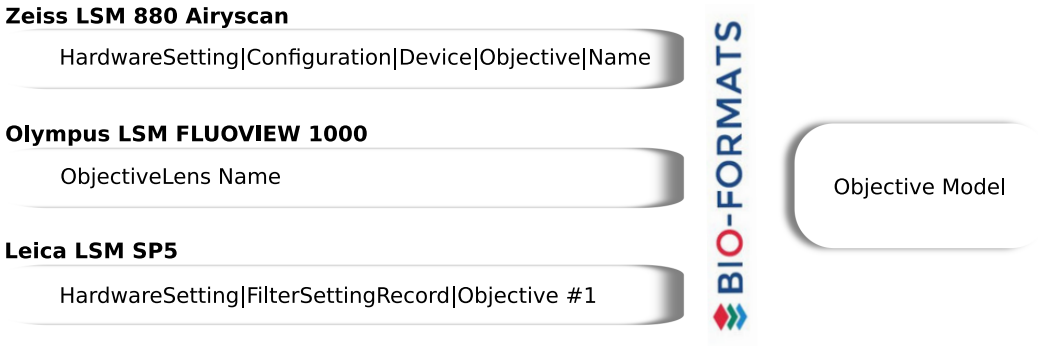

Figure 2.5. Objective model name representing in different vendor formats. As original metadata (left) and after harmonisation by Bio-Formats (right).

2.1.4. Metadata

In the following, we focus on metadata related to bioimaging. Metadata for bioimaging is the information that contributes to the understanding of both the image in the technical sense and its content and context. This also includes data to identify resources as well as links to other components to document structural relationships. We distinguish between metadata related to the experiment, the sample, and the techniques [16]. We formulate the following three main categories for metadata related to experiments in microscopy:

Experimental metadata: Metadata on the experimental procedure, are mostly captured in laboratory notebooks and includes general information such as the cell line used. Experimental metadata can also include the contextual information of the individual sample, such as the organism under consideration or the stage of the cell cycle. In sum, experimental metadata specify the objects or processes studied by microscopy.

Sample preparation metadata: Another group is formed by the metadata relating to the sample preparation. This can be, for example, data regarding the chemical fixation of the sample or treatment, such as how the sample was tagged, for better visualisation. These data are usually recorded in corresponding experiment protocols.

Technical metadata: The third group is formed by the metadata for image acquisition. These data describe both, the technique used and its configuration, as well as, for example, the image dimension or the scaling of the physical dimension with

respect to the pixel size (mapped into the sample space). Therefore the group of image acquisition metadata or technical metadata can again be divided into three subgroups (Figure 2.6).

Category	*Explanation of metadata*	*Examples*	*Collection method*	*Standards*
Microscope Hardware Specifications	Describes the microscope used for image acquisition and lists its hardware components	Objective manufacturer, catalog number, and magnification	In principle, easy to automatically capture, log and maintain during acquisition; in practice, varies significantly based on microscope manufacturer and image file format	Example: OME; 4DN-BINA-OME (NBO) Microscopy Metadata specifications #
Image Acquisition Settings	Describes the specific settings employed during image acquisition	Illumination power settings, exposure time, detector gain		
Image Structure	Structure of the image data file	Pixel size, number of focal planes, channels, and time points, dimension order		

Figure 2.6. Subcategories of technical metadata. Metadata category recommendations in terms of image acquisition adapted from [16].

In addition, metadata for analysis and processing, the so-called analytical metadata, and metadata for publication and data dissemination are added in the further process. The generated metadata can be stored in various forms and formats. For example, they could be captured as free text in text files or spreadsheets, or saved in the data file format itself.

2.1.4.1. Vocabular

In order to comply with the FAIR principles and to make data linkable, this metadata should be subject to certain standards [17]. A first level of standardisation is a predefined vocabulary for the used metadata. Processes and experiments can only be related to each other if the described objects or processes are named by means of unique vocabulary. The different spelling alone makes it difficult to associate them with each other. The challenge now is to choose the metadata in such a way that the description we want to achieve is unambiguous for humans and machines. This means that there should be as little room as possible for interpretation of what is being described based on the chosen vocabulary. Mostly, the understanding of chosen vocabulary requires a certain context to avoid misinterpretations. But e.g., the scientist from biology assumes a different context than e.g., a computer scientist. Here we can distinguish two main types of misinterpretations. On the one hand, one describes the same thing with other words (synonyms). Thus, *Dyschromatopsia* is a synonym for *Colour blindness* and *Abnormality of the eye*. On the other hand, one uses the same word with different meaning. We demonstrate this with the word *Sample*. Depending on the context and field of study, *Sample* can have different meanings [18]:

- Sample (statistics), a subset of a population (complete data set),

- Sample (signal), a digital discrete sample of a continuous analog signal,
- Sample (material), a specimen or small quantity of something,
- Sample (graphics), an intersection of a colour channel and a pixel.

Searching a large data pool for a particular term from our vocabulary yields incomplete results because synonyms are not related or the data found are disjoint. So, in addition to a vocabulary, we also need a context for the description and an indication of what the description might be called in another context. This semantic linkage can be generated by means of ontologies. A good example of the meaningful linking of data from multiple imaging modalities is the open source platform IDR. Combining the data with corresponding controlled ontologies enables the analysis of gene networks and reveals functional interactions [19].

2.1.4.2. Schema

Another level of standardisation for metadata is the metadata schema. According to ISO 23081, the following schema definition applies: "A schema is a logical plan that shows the relationships between metadata elements, usually by specifying rules for the use and management of metadata, particularly with respect to the semantics, syntax and optionality (commitment level) of values." [20]. A metadata schema thus contains information not only on the semantics, but also on the overall structure of the metadata. This metadata schema usually relates to a specific scientific domain. In order to establish a relationship to other metadata standards, so-called crosswalks are required, i.e. a specification of how elements of one metadata schema can be mapped to elements of another schema [21].

2.1.4.3. Linked Data

To ensure that the semantic context of the data is clearly defined and that the metadata meet the requirements of machine readability, it makes sense to integrate so-called Linked Data into the metadata schema. "Linked Data is a way of structuring and sharing information using links." [22]. Because of the links, the data can be interpreted by machines via semantic queries. Linked Data is often based on web technologies such as Hypertext Transfer Protocol (HTTP), Uniform Resource Identifiers (URI) or Resource Description Frameworks (RDF) [22], as described in Section 2.1.2. The example in Figure 2.7 shows the transformation steps of information from a message to a structured message, into Linked Data and finally the transformation into a knowledge graph based data model. In our example, at the end all semantic information is formulated in RDF and represented as tuples of object, subject and predicate [23]. In this graph data model, a bundle of statements about a single subject (like Tim in our example) is called a resource [22]. Serialisation formats are used to transfer the data model into a universal storable format from which the data model can be reconstructed afterwards. Such formats are used for the exchange or transfer of information between different applications. There are various serialisation formats for RDF such as Turtle, RDF/XML, JSON-LD, RDFa

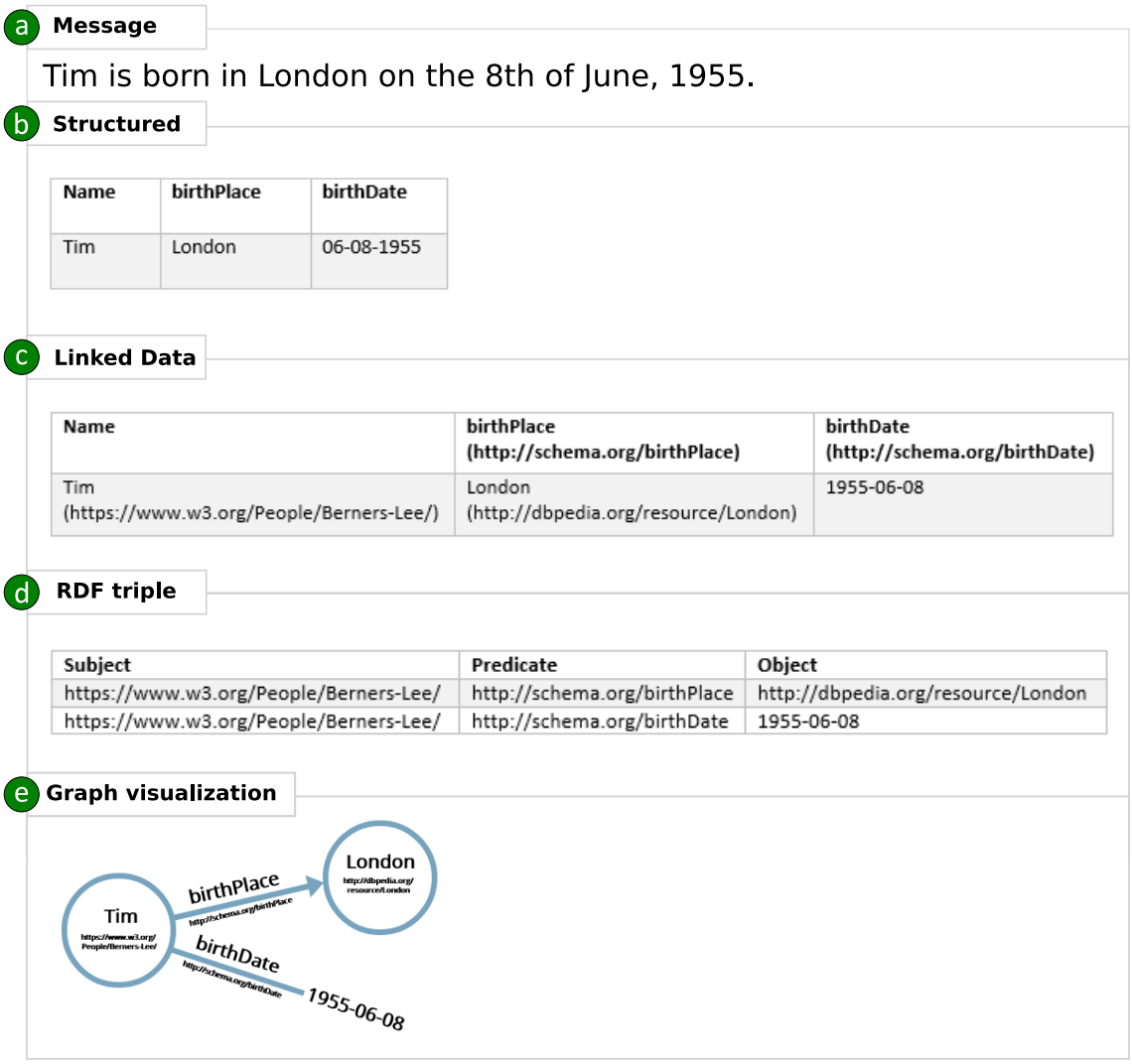

Figure 2.7. From the message to the knowledge graph. Reformating the message **(a)** to a structured message **(b)** lacks semantic background and leads to questions like "Who is Tim?", "Which London?" and "Which date format is used?"; **(c)** Adding links to integrate the semantic background; **(d)** Modelling in RDF: Putting each piece of information into its own statement (triple); **(e)** Visualising RDF as a graph, see [22].

(RDF inside HTML) (Listing 1,Listing 2) with different advantages and disadvantages. Which format is most suitable depends on the application. In summary, such Linked Data concepts or formats as RDF/Xml or JSON-LD enable the linking of a wide variety of information and the formation of directed knowledge graphs in machine-readable format [24].

2.1.5. Ontologies

An ontology denotes a set of controlled relational terms [25] of semantic information. Ontologies specify not only the vocabulary used, but also the domain specific concept

```
@prefix tim: <https://www.w3.org/People/Berners-Lee/>.
@prefix schema: <http://schema.org/>.
@prefix dbpedia: <http://dbpedia.org/resource/>.

<tim> schema:birthDate "1955-06-08"^^<http://www.w3.org/2001/XMLSchema#date>.
<tim> schema:birthPlace <dbpedia:London>.
```

Listing 1 RDF Serialisation in Turtle. Serialisation of example in Figure 2.7 in Turtle [24]

```
{
 "@context":{
  "dbpedia": "http://dbpedia.org/resource/",
  "schema": "http://schema.org/"
 },
 "@id":"https://www.w3.org/People/Berners-Lee/",
 "schema:birthDate": "1955-06-08",
 "schema:birthPlace":{
  "@id": "dbpedia:London"
 }
}
```

Listing 2 RDF Serialisation in JSON-LD. Serialisation of example in Figure 2.7 in JSON-LD [24].

and the relation to other concepts. For illustration, let us consider the previous example for synonyms and elaborate the concept. *Dichromism* is defined in the ontology of human phenotypes as a subset of *Abnormal eye physiology*. The `class`, also called `concept` or `type`, *Abnormal eye physiology* contains all possible subclasses, and *Dichromacy* is an instance of this class (Figure 2.8). The association between two entities is called a `property` and is categorised as follows:

- `Definition`: relationship between a term and a text
- `Is-a`: relationship of two terms.

The relationship between two terms from different ontologies is called `mapping` or database `cross reference`. Thereby, a mapping can be further described in the type of relationship it exactly represents. However, often the simple relation `Same-as` is used.

Over time, many discipline-specific ontologies have been developed. Examples of ontologies commonly used in microscopy are National Centre for Biotechnology Information Ontology (NCBI), Biological Imaging Methods Ontology (FBbi), and Bioimaging Ontology (EDAM-BIOIMAGING). Nevertheless, it is difficult to relate the appropriate ontology to the research project in everyday scientific life. In practice, the application of ontologies is slowed down by various factors. It is often challenging for the researcher to filter out from the multitude of available ontologies the appropriate one by means of which the research project can be best described.

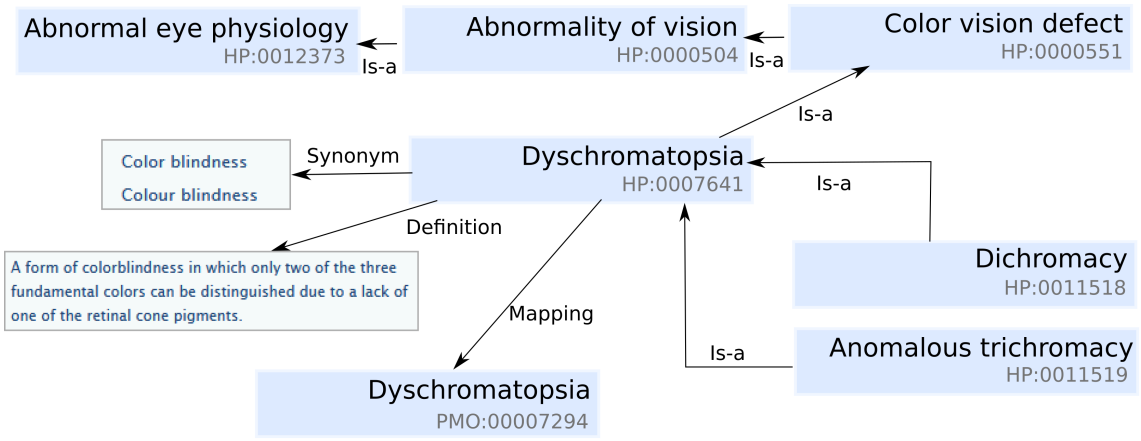

Figure 2.8. Relationships between ontology entities. Visualization of instance Dyschromatopsia in Human Phenotype (HP) Ontology [26].

Nevertheless, for the FAIRification process it is important that the corresponding terms are referenced by an ontology. However, the scope of an ontology does not always do justice to the research process, and further development and adaptation is essential, driven from within the community. Community-driven approaches [13] help to push ontology updates and recommend appropriate ontologies. This community process can also drive the implementation of appropriate software tools to accelerate the use of ontologies at the institutional level prevent the increase of documentation for researchers. Access and reuse of ontologies can be fostered by universally accessible repositories for ontologies. Examples of such repositories are BioPortal [27] and Ontology Lookup Service (OLS) [28]. Both repositories provide, on the one hand, a web-based application for the scientist to search ontologies and, on the other hand, a programming interface to access the content of the ontologies by machine.

2.2. Open Microscopy Environment (OME)

The Open Microscopy Environment (OME) consortium is a multi-collaborative initiative consisting of academic laboratories and commercial vendors dedicated to developing free software tools and specifications for managing biological microscopy images. This includes storage, visualisation, annotation, and analysis [29] of these data. The main focus is on the development of standards for file formats and metadata to enable sharing and interoperability of proprietary microscopic image containers. The OME Data Model and the OME-TIFF file format, the software library Bio-Formats and the image data platform OME Remote Objects (`OMERO`) have become established standards in biological imaging. The OME took a key role in enabling sharing and publishing microscopy data [30]. However, the ongoing development in microscopy and storage technology demands continuous development in this area. As mentioned above, standards regarding metadata and their structure play an important role in data management. Without standards and appropriate crosswalks, metadata can be misleading for other scientists and thus impede the extraction of knowledge and negatively influence the subsequent use of the data.

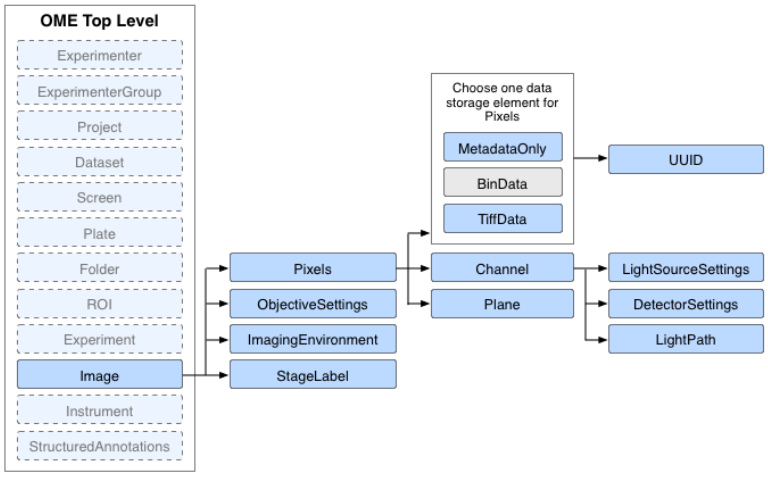

Figure 2.9. OME Data Model in detail (image branch). Branch of image object and related subbranches of OME Data Model version 6.0.1.

2.2.1. OME Data Model

The OME Consortium developed the OME Data Model [31], a specification for the storage and exchange of metadata in biological imaging. This data model has been established for 15 years and is currently the de facto standard. The data model provides a formal description of the structure, meaning, and interrelationship [30] of mainly technical metadata describing microscopic imaging. The model includes metadata regarding the physical aspect of the image, such as dimensions, but also image acquisition and other annotations [32] (details are provided in Figure 2.9 and

Figure 2.10). To reflect the rapid progress in the development of new microscopy technologies and applications, the OME Data Model is being reviewed and extended by the international community. The formulation of extensions such as the 4DN-BINA OME model [33] or RIKEN MetaDatabase [34, 35], requires an iterative process with repeated evaluation in practice. An ever-growing international network of researchers from the fields of microscopy and imaging and image analysis continuously drives this process [16, 35, 6].

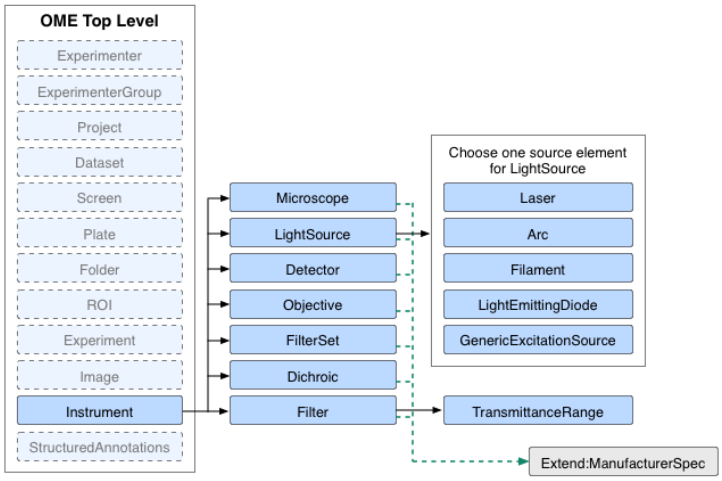

Figure 2.10. OME Data Model in detail (instrument branch). Branch of instrument object and related subbranches of OME Data Model version 6.0.1.

2.2.2. OME-XML

The OME Data Model is described in Extensible Markup Language (XML). XML represents hierarchically structured data in a text file that is readable and interpretable by both humans and machines. The established file format OME-XML is used to store metadata according to the OME model. Thereby, all the metadata stored in the image container regarding image acquisition and technical experiment are stored according to the OME Data Model in this easy and standardised readable form. OME-XML can be used as a general file format for data migration of microscopic metadata from one site or user to another [32].

2.2.3. OME-TIFF

OME-TIFF is a multi-layer TIFF file that embeds metadata in the header in the form of OME XML. This allows the pixels to be read with any TIFF-compatible program and the metadata to be extracted with any OME-enabled application [32]. OME-TIFF is also supported by many microscopy vendors as well as public image repositories [36].

```
<!--OME-XML Tag Image: -->
<Image ID="Image:0" Name="Series 1">
  <InstrumentRef ID="Instrument:0"/>
  <ObjectiveSettings ID="Objective:0:0"/>
  <Pixels BigEndian="false" DimensionOrder="XYCZT" ID="Pixels:0"
    Interleaved="false"
    PhysicalSizeX="0.069" PhysicalSizeXUnit="µm"
    PhysicalSizeY="0.069" PhysicalSizeYUnit="µm"
    PhysicalSizeZ="0.25" PhysicalSizeZUnit="µm"
    SignificantBits="12"
    SizeC="3" SizeT="1" SizeX="1024" SizeY="1024" SizeZ="34"
    TimeIncrement="1.0" TimeIncrementUnit="s" Type="uint16">
    ...
    </Pixels>
</Image>
```

Listing 3 OME-XML in detail. Example of conversion to OME-XML metadata read from *.oib file format and originating from Olympus LSM FLUOVIEW 1000 setup.

2.2.4. Bio-Formats

Bio-Formats is a widely disseminated software library written in `Java` for reading and converting proprietary microscopy file formats into open and accessible file formats, for example OME-TIFF. The metadata contained in the microscopy data are transferred into a common structure, the OME Data Model [15, 37]. By transferring the data and metadata into a unified format, working with different microscope setups and the resulting data is made much easier or even possible in the first place. Bio-Formats is important both for visualisation independent of the microscope manufacturer's software and for subsequent analysis of microscopy data. Tools such as `Matlab` and `ImageJ/Fiji` use Bio-Formats to enable visualisation and analysis of microscopy data independent of the vendor-specific format.

For metadata, Bio-Formats distinguishes three different types: [37]:

Core metadata: represent only metadata for understanding the basic structure of the pixels stored in the image container (dimension size and order, colour arrangement, resolution and so on).

Original metadata: are key/value pairs specific to the original file format. There is no naming consistency and no mandatory compatibility between the different microscope formats with respect to these metadata.

OME metadata: are metadata from core and original metadata converted to the OME Data Model. The amount of metadata converted varies by format and remains a development process. A complicating factor is often the missing or not open documentation of the original metadata used by the microscope manufacturers. Since many microscopy vendors have their own metadata representation with specific vocabulary, comparing and linking multimodal data in terms of metadata is

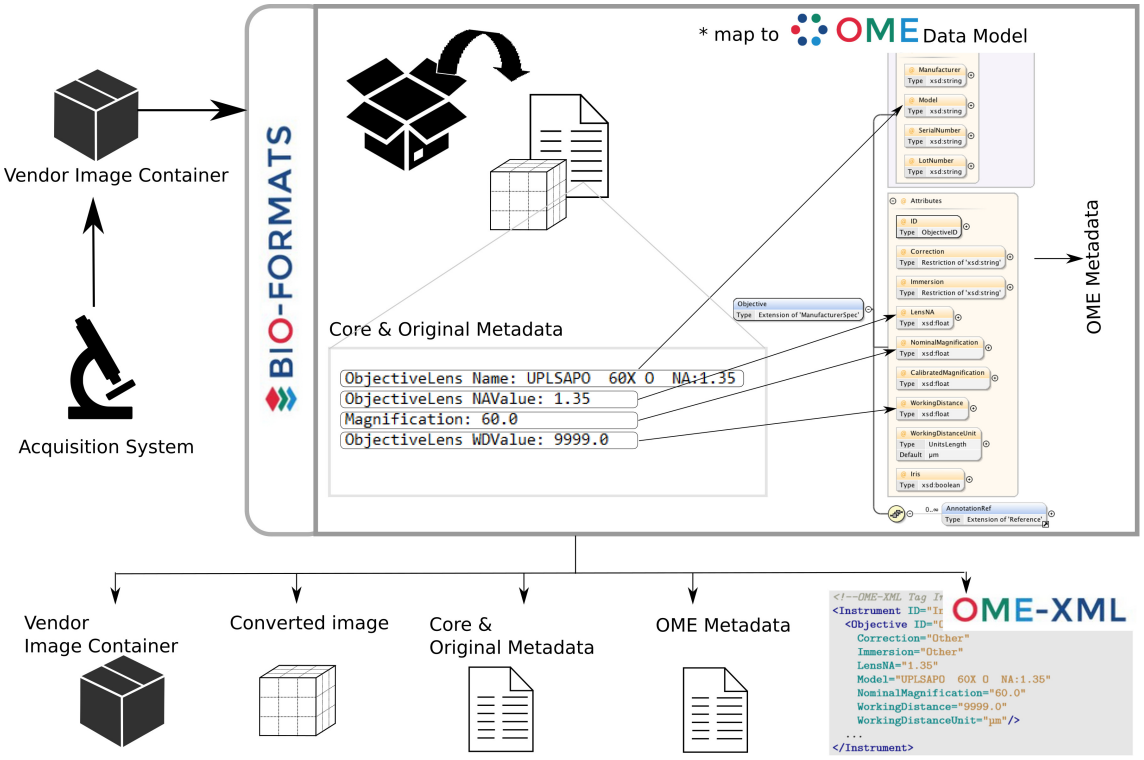

Figure 2.11. Metadata harmonisation via Bio-Formats. Original metadata read from the image file are harmonised via Bio-Formats v6.0.1. according to the OME Data Model. Bio-Formats offers various output formats and conversions for data and metadata.

challenging. With the help of Bio-Formats, the vendor-specific metadata of different imaging data formats can be harmonised (examples in Figure 2.5). By mapping the different metadata specifications to the OME Data Model, the OME metadata results in a unified metadata representation with respect to the entities captured in the OME Data Model (Figure 2.11). This facilitates categorisation or filtering of multi modal data based on this standardised metadata.

2.2.5. Software Platform `OMERO`

OME Remote Objects (`OMERO`) is a `Java`-based open-source software platform specialised for the management, annotation, cooperation, and visualisation of biological microscopy data. Designed as an image repository, the images are stored in raw format on a central database-driven instance, the `OMERO` server. The images are placed in the file system in an `OMERO` proprietary system structure. `OMERO` provides standardized interfaces for processing and analysis of this microscopy images and enables access to image and metadata independent of file format by Bio-Formats. The metadata stored in the image is read out after the image data has been transferred to the server using Bio-Format and stored in the database in the form of OME metadata. Data added by the user, such as further annotations, but also ren-

dering settings, are also stored in the database. In addition, it is possible to store non-image data on the server and link them to folders containing images or directly to images.

2.2.5.1. Clients

The data is accessed by means of special software, the clients. Software-restricted access to the data also ensures that the original data once transferred to the server cannot be manipulated. The original therefore remains unchanged. The functionality of the clients covers in the basic implementation of the organisation, annotation and visualisation of the data, as well as the sharing of the data and the search and filtering of the data. The clients consists of several `Java` applications as well as `Python` bindings and a `Django`-based web application [38]:

`OMERO.cli`: is a command-line based `Python` tool with high functionality. It can be used for administration, deployment as well as advanced user tool. Unlike the other clients, Windows is not supported as an operating system here.

`OMERO.insight`: is a desktop client. To use `OMERO.insight`, the appropriate `Java` version must be installed on the local computer. `OMERO.insight` is in a maintenance mode, i.e., it is updated only in case of major bugs.

`OMERO.importer`: is integrated into `OMERO.insight`, but can also be run as a standalone application. This software provides the transfer functionality for image data to a running `OMERO` server platform. By using Bio-Formats, the proprietary image formats are prepared for upload to the repository.

`OMERO.web`: is a web-based client and will be supported by different browsers. The base module does not require any additional installations. The functionality is extensible by so-called plugins and microservices. By default, it covers almost the same functionality as `OMERO.insight`, except for the transfer of image data to the repository.

2.2.5.2. Permission Control

Access to `OMERO` is regulated by an authorisation control system for users via groups. The only exception is a dedicated area, the PUBLIC group. All data in this area can be viewed by the public without any login. The data in a public group is no longer assigned to a specific user via the `OMERO` system.

All other data in `OMERO` are bound to a specific user as well as to a specific group. Each user belongs to at least one group. All user actions inside `OMERO` are always assigned to the currently selected group. A user in the `OMERO` system can have 4 different roles [39]:

- **Administrator:** full system and group control,

- **Restricted administrator:** subset of administrator privileges for a defined set of task, independently of group permissions,
- **Group owner:** additional rights within a certain group,
- **Group member:** standard user.

The access to the data is regulated by permission levels, which depend on both the role of the user and the permission level of the group (Figure 2.12).

When a user creates an object or a link in OMERO, the owner of this object or link is normally marked as the user himself. An exception is the import of a restricted administrator for another user, in which case the data belongs to the user on whose behalf the import was performed. A standard user has unrestricted access to all his or her data and can perform all actions (except for a change in ownership) [39]. The group permission level defines the user's access to the data of the other users. There are four different levels, i) Private, ii) Read-only, iii) Read-annotate, and iv) Read-write [39]. The appropriate permission level is derived from the planned handling of the data and the expected level of collaboration.

The role of a user is assigned by the administrator. The permission level of a group can be changed by the group owner or the administrator. They can also add or exclude individual users from groups.

Action	Private	Read-only	Read-annotate	Read-write
View	**Y** / Y / N	**Y** / Y / Y	**Y** / Y / Y	**Y** / Y / Y
Annotate	**N** / N / N	**Y** / Y / N	**Y** / Y / Y	**Y** / Y / Y
Delete	**Y** / Y / N	**Y** / Y / N	**Y** / Y / N	**Y** / Y / Y
Edit	**Y** / Y / N	**Y** / Y / N	**Y** / Y / N	**Y** / Y / Y
Move between groups	**Y** / N / N	**Y** / N / N	**Y** / N / N	**Y** / N / N
Remove annotations	**Y** / Y / N	**Y** / Y / N	**Y** / Y / N	**Y** / Y / Y
Mix data	**N** / N / N	**Y** / Y / N	**Y** / Y / N	**Y** / Y / Y
Change ownership	**Y** / Y / N	**Y** / Y / N	**Y** / Y / N	**Y** / Y / N

Figure 2.12. Groups and permission system in OMERO. Permission table covers administrators/owner/member privileges in OMERO version 5.6.3 [39].

2.2.5.3. Transfer of Data

Several tools are available for transferring or downloading data in the OMERO repository. Which tool is most suitable is decided by the system configuration and the user client base.

For users with no affinity for command lines, the `OMERO.importer` is a software tool for transferring data to the repository. The user is guided through this process by a graphical user interface (GUI). `OMERO.importer` offers an interface for data selection and for specifying the storage location as well as for categorisation by means of keywords. A graphical feedback informs the user about the progress of the data transfer. Both `OMERO.insight` and `OMERO.web` offer GUIs for downloading and exporting

image data and annotations. The image data can be downloaded in the original format or exported to another format such as JPEG, PNG, TIFF or OME-TIFF (in version 5.6.3). The original metadata can be downloaded in text format. Additional metadata will be included in the OME-TIFF export. With the help of `OMERO.scripts`, a scripting service for processing and analysis (see Section 2.2.5.7), annotations such as key-values can also be saved or exported in a comma-separates values (csv) format.

For users with experience in command line usage and the appropriate operating system, `OMERO.cli` provides the corresponding functions. However, the transfer process must be carried out and specified by the user. For example, before transferring, newly specified storage locations must be created in the form of projects or data record directories. It is possible to specify additional configurations, e.g. to have the import of files processed in parallel. Extended import functions are also available, which can be used to link other files or add metadata during import.

2.2.5.4. Sharing Data

In `OMERO`, data sharing is mainly implemented by assigning the data to a group with corresponding permissions. Furthermore, it is possible to move data to other groups. To make data available in different groups at the same time, a duplicate of the data can be created at the command line level as link, which is transferable to another group without moving the original data itself. These duplicates are dependent on the original data, i.e., if the original data are removed, the duplicate can no longer be used. In addition, `OMERO.web` provides a sharing functionality for individual objects with any member in the `OMERO` system. However, OME has announced to disable this functionality.

2.2.5.5. Metadata

In `OMERO`, metadata is captured at different levels. The metadata stored in the image container is read and harmonised with the OME Data Model via Bio-Formats and stored in `OMERO`'s database. This metadata cannot be modified by the user, but it is possible to export this data to a text file format for download. In addition, users can add metadata as free text in the form of tags or key-value pairs to their own data, or for data they have appropriate permissions. Tags are useful for categorisation e.g. by naming the microscope system or for labelling data for publication. Key-value pairs are suitable for describing tags with different values, e.g., the type of organism studied. At commandline level, namespaces can be added for better structuring of the key-values (Figure 2.13). Metadata descriptive files are linkable with an object such as a directory or file by adding them as an annotation.

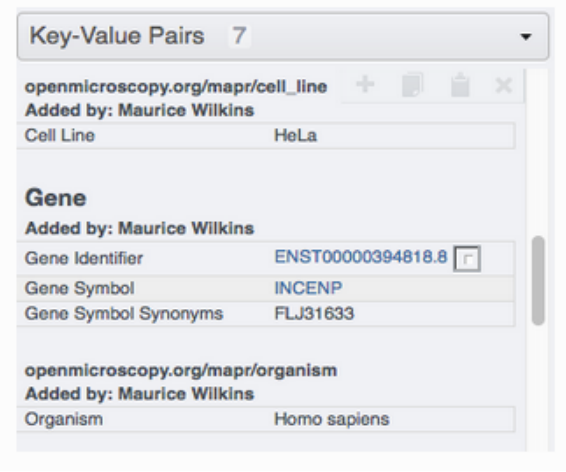

Figure 2.13. Structured key-value pairs in `OMERO`. Example of key-value pairs with structured namespaces (cell_line and organism) in `OMERO` version 5.6.3, see [40].

2.2.5.6. Application Programming Interfaces

An Application Programming Interface (API) describes generic code based software interfaces. They support the connectivity and exchange of information between applications, independently of there implementation and provide a set of tools that programmers can use. Via the API, specific functions are accessible to a proprietary software application. `OMERO` offers application programming interfaces in `Python`, `Java`, `C++`, `R`, and `Matlab`. `OMERO` also supports a scripting service, `OMERO.scripts`, which allows `Python` scripts to be run on the server and to be called from any of the other client [41].

2.2.5.7. System Architecture

`OMERO` "is a multi-component data management platform comprised of servers and clients written in `Python`, `Java`, and `C++`" [42]. The central component of the infrastructure is the `OMERO.server` (Figure 2.14), a `Java` application that provides access to the underlying storage facilities and relational database connected to `OMERO`. The `OMERO.server` processes the data for delivery to the client applications. Via a standard internet connection, the client applications access the data using the `OMERO.server` API. According to the parameters stored in the database or according to the specifications of the requesting client application, the binary image data read in by the rendering service is scaled. Such parameters or client specifications can be, for example, rendering settings or projections, but also corresponding authorisation regarding data access.

Integrated into the `OMERO.server` is a scripting service `OMERO.scripts`, with which the internal processes can also be accessed using `Python`. These can be executed by the clients within the `OMERO.server` by means of a grid of processors distributed in the background, managed by the OMERO.grid service. All computed results are returned to the calling script [42]. Thus, `OMERO.scripts` provides a way to extend the

functions provided by OMERO without changing the underlying system.

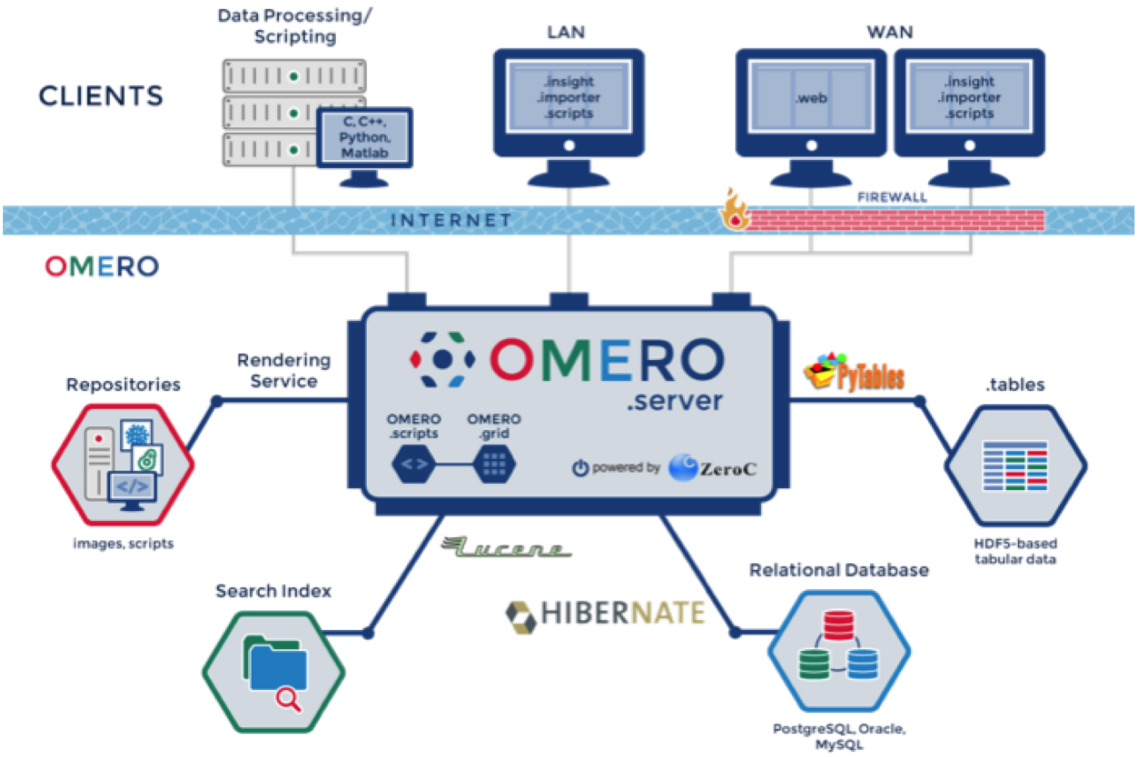

Figure 2.14. OMERO.server architecture. Architecture OMERO.server version 5.6.3 and components, see [43].

The OMERO.web as a framework functions in the first layer as a Python client that generates the corresponding HTML and JSON data response via modular Django web applications. Using HTML, a user-friendly presentation of the web content as well as the requested data is achieved, as the structured data formulated in JSON by the server can be easily integrated into HTML. In the background, OMERO.web also works as an independent web server. OMERO.web runs in production with nginx [44] to enable Web Server Gateway Interface (WSGI) [45], the Python standard for web servers and applications and primary deployment platform for Django. Communication with the OMERO.server is done remotely using the OMERO Python API omero.model via ZeroC's Internet Communication Engine (ICE) (Figure 2.15). To extend OMERO.web it is possible to integrate self-developed Django apps.

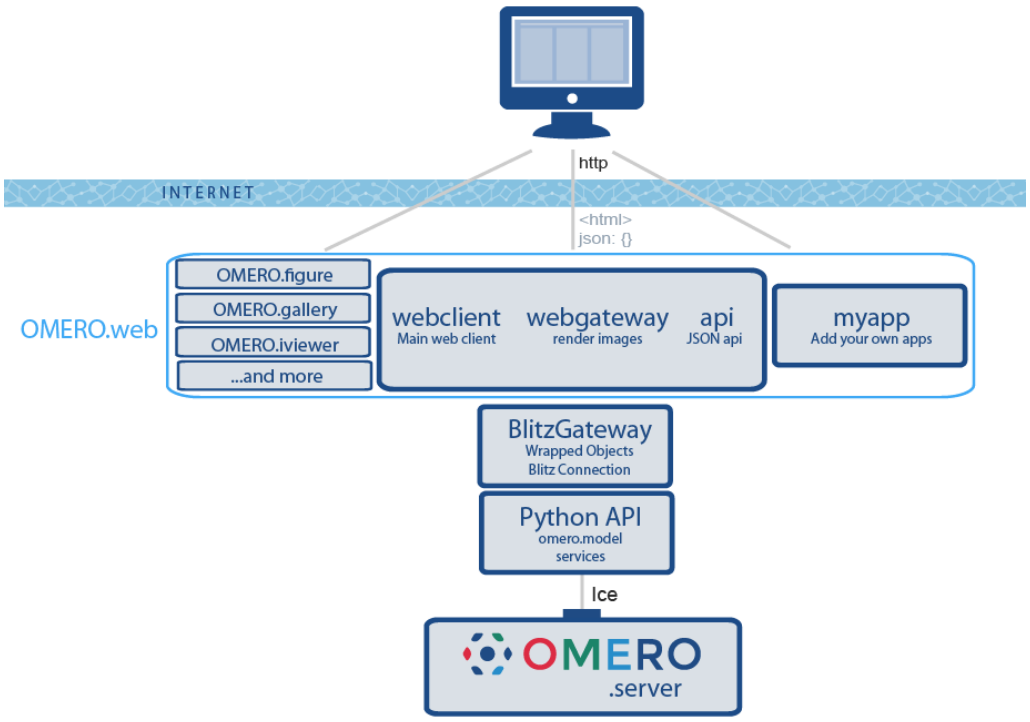

Figure 2.15. `OMERO.web` architecture. Architecture `OMERO.web` framework version 5.6.3, see [46].

3 Engineering Software Support for Data Management for Microscopy

In this chapter we consider the environment of the local imaging facility of the Center of Cellular Nanoanalytics Osnabrück (CellNanOs) located at the University Osnabrück. Based on an analysis of the local conditions, we can formulate specific requirements for data management and name specific points whose optimisation lead to the realisation of the FAIR requirements and to a higher acceptance. We concentrate on two points. Firstly, an extension of the already used software for microscopy data management for faster and more convenient downloading and simple sharing of data. This simplifies the existing workflow and further increases the acceptance of the introduced data management. Part of this work is to improve data downloading through a web-based approach and provide a significant speed advantage over the features offered by the data management solution. In addition, simplified data sharing with external colleagues should be established. This fosters data sharing and eliminates the need to use file sharing systems such as Dropbox or the current university's limited file sharing service, myshare. These requirements are implemented as a software extension, the `OpenLink` tool.

In the second part, we describe an extension of the data management software already in use to simplify the labeling of metadata and, depending on the requirements, to standardize it institution-wide or at least within a working group. Sufficient documentation is necessary for a systematic reuse of the data. Subject-specific standards, at least for the technical metadata, are available in the form of the OME Data Model. However, they do not cover special cases. Thus, there is a need for a guided workflow with standardisation elements related to the metadata associated with the microscope data. These requirements are implemented with `MDEmic` as an extension of the `OMERO.importer`, which forms the second aspect of this thesis.

3.1. Research Data Management at Imaging Facilities

3.1.1. Research Data Management Environment at iBiOs

In order to design research data management at the institutional level, it is necessary to analyse the type and amount of data, as well as the existing hardware and software infrastructure. The users' skills in handling computers and software have a

great influence on the design of the local system landscape and the selection of suitable supporting software tools. Furthermore, the previously established workflows have to be taken into account, as well as the requirements for data and research data management, e.g. from third-party funding bodies. This then results in the specific requirements that must be covered by the configuration and enhancement of local data management using suitable tools.

In Integrated Bioimaging Facility (iBiOs) at CellNanOs, the largest and most storage-intensive volume of data is generated by microscopes. In the exchange with e.g. external cooperation partners, further data formats may occur. As the technical development in the microscopic field also leads to a very significant increase in the amount of data, it is essential to focus on the management of this data and the optimisation of the workflows. Therefore, the other types of data generated during the preparation of experiments for microscopy and their post-processing will only be discussed in this thesis in marginal terms.

The existing hardware infrastructure of the department of Biology/Chemistry consists of a large number of user workstations and some work group servers. Due to the local infrastructure architecture, the servers are not accessible to users institute-wide or across work groups. There are also microscopy workstations and some analysis workstations. The use of Windows systems is common, with a few macOS exceptions. The users prefer a graphical user interface and have little or no experience with working on the command line. The iBiOs user group consists of only a few long-term employees. The majority of users are on site for a limited period of time (3-4 years) due to project work, as well as only for a short period of time (a few months) due to university involvement in the form of students. The data generated by this user group should be available and reusable even after they leave the institute. A large number of users belong to the university and therefore have access to the university's internal system infrastructure. However, there are also external users, for example in the form of collaborations with working groups from other research institutions, who do not have access to the university infrastructure but are dependent on the exchange of data.

The majority of the data is subject to the storage requirements of third-party funders. The external requirements regarding data management come firstly from the german research foundation "Deutsche Forschungsgemeinschaft" (DFG), since the local imaging facility and a large proportion of the participating working groups are integrated into a local Sonderforschungsbereich (SFB). The DFG requests that the data to be stored and made available in the long term as a contribution to the traceability and quality of scientific work. This includes the immediate availability of the data in such a way that it can be meaningfully reused and further used by third parties. Furthermore, archiving for a period of 10 years is recommended [47]. At the time of this thesis, the university is working on a university-wide data research policy, which refers to the subject-specific requirements.

3.1.2. Requirements and Specification

From the preceding considerations, requirements for the research data management environment can now be derived in detail. We analyse the established research data management structure with regard to these criteria and focus on requirements that still need to be optimised.

Because of the prevalence of different systems software, the data management environment should be independent of the type of operating system. Due to the high staff turnover, the training and introduction of new users for the research data environment is frequently requested. The barriers for initial use in the form of installation and configuration should thus be as low as possible. This also minimises the effort for the necessary support in the form of manpower by IT specialists. All tools and services used should offer a graphical user interface. Due to their easy availability, either central services or web-based tools that can also be made available to external cooperation partners are suitable here. This also reduced the requirements on the locally used system for operating the service in terms of third-party software dependencies. In consequence of the demand long-term storage, the structures and formats should be selected in such a way that readability can still be guaranteed after 10 years or that there is a suitable migration mechanism if system or format adaptations are necessary. The subsequent use by third parties requires structuring and documentation of the data according to subject-specific standards. The complexity and variance of microscopic image data increases the scope and the requirements in terms of methods or models used for documentation, and requires a good overview and linkage of parameters already captured.

Thus, the main focus when dealing with microscopy images is not only the readability and visualisation of the acquired data, but also the access to the metadata stored in the image container and the appropriate representation of associated data. Suitable software solutions are needed to organise the data, metadata, and linked data in a multimodal and format-independent way. The greatest challenge is the variety of proprietary file formats used by microscope manufacturers. At iBiOs the research data management was implemented with the help of OMERO (see Section 2.2.5). The OMERO platform enables cross-operating system, gui-supported access to data and supports over 150 different microscopic file formats. The functionality of the repository is entirely web-based with the exception of image data transfer. By configuring the system accordingly, access to the data taking place both within the university and externally via the designated permission system, without the need to install additional software. To support collaborative work, users are assigned to groups. Appropriate rights to the data can be assigned for these groups. This gives the individual working groups the option of working separately or joining forces on joint projects.

Using the OMERO platform as a centralised repository for acquired and processed microscopic data results in the following workflow (Figure 3.1): The data acquired at the microscope are uploaded directly from the microscope workstation to OMERO, unless intermediate processing is required due to quality assessment. The direct upload protects the raw data from modification in subsequent post-processing steps

and the data is immediately accessible for the cooperation partners (depending on the group assigned to the data). The data can now be annotated and used for analysis and linked to other data. In practice, there is a need for improvement in

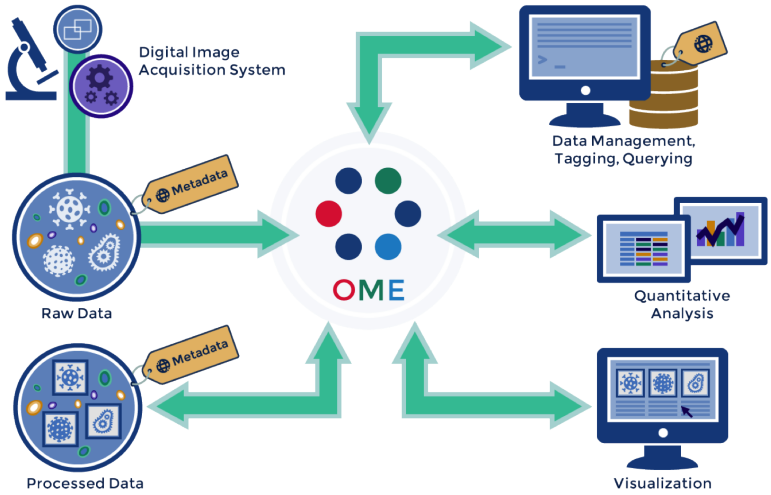

Figure 3.1. Data management workflow with OMERO. Workflow with centric OMERO repository as main data storage, see [48].

data transfer. Above all, post-processing using 3rd party modules, which cannot be made available on the server due to licence restrictions or technical barriers, requires an optimisation of the download processes. Direct access to the data stored in the repository, such as via a network share, cannot be realised due to the present research data management concept. Reason for this are, for example, the data structure created by the repository as well as user permissions.

OMERO offers the user the possibility to add metadata in the form of free-form key-value pairs and as separate files in any format. Metadata read from the image container using Bio-Formats is displayed in harmonised form as OME metadata (see Section 2.2.4). However, this metadata cannot be edited. But not all locally generated file formats are read correctly in terms of metadata. For example, the OME Data Model does not cover special cases, such as the local spinning disk setup with two detectors for one channel, or technical custom developments available in iBiOs, such as the Lattice Light Sheet setup. So there is a clear need for a guided workflow with standardisation elements for the metadata associated with the microscope data. This workflow should integrate the exact specification of these microscopes. Furthermore, a function is needed for the standardised input of metadata regarding other entities (see Section 2.1.4), which can also be used across work groups.

3.2. OpenLink - Data transfer and sharing

This chapter describes a software extension of the OMERO.web client consisting of a Python script and a Django web plugin. It is a functional extension that uses the given API and extends the existing functionality. Through an additional modification in the configuration of the architecture used, extended data access is made possible. In the following, the software realisation is described in terms of design and implementation, as well as the evaluation and comparison with already existing components of similar functionality. The design comprises the conceptual design, while the implementation describes the concrete realisation.

3.2.1. Related Work

OMERO itself offers various options for downloading data. It is possible in OMERO.insight and OMERO.web to select individual files for download, but not complete dataset or project folders. Linked files in form of attachments must be downloaded separately. It is not possible to download data from different groups together. In OMERO.insight, neither the progress nor the expected duration is displayed after the start of the download, which makes it difficult for the user to plan the time needed for the further work process. The OMERO.web download has an additional special limitation. The data selected by the user for download are combined in a zip file before transfer by the OMERO system. The system allows a maximum target size of 1 GB for this zip file. If this size exceeded, the data can only be downloaded individually or in smaller packages. The download itself is managed via the web browser, which also provides the progress visualisation. The command line download options are not discussed here because the relevant target users lack the prerequisite to use this option and Windows is not supported as an operating system.

Sharing in OMERO is mainly realised through the authorisation systems of the groups (Section 2.2.5.4). If the user is a member of a non-private group, he has access to the data of other users of this group according to the group rights (see Section 2.2.5.2). Any sharing function in OMERO requires that the user has access to the OMERO system in the form of an account and has the appropriate rights. The function for partial sharing of data for users without appropriate rights is limited to members of the local OMERO instance.

3.2.2. Design

To increase the usability of OMERO, we have implemented an improvement of data access and download in the OMERO.web client. Focusing on the web access of omero has several advantages. On the one hand, OMERO.web allows modular extensions without interfering with the software itself. On the other hand, a web browser, and thus access to the data and the integrated download functionality, is available without complexity. This means that no additional software is required on the local computer to access the data. For the file transfer we want to use a high-performance software tool for web downloads that enable the transfer of larger data via a URL. Since system and software environment independence is also a priority here, a solution

is needed that is natively available on most operating systems in order to avoid subsequent installations. The programme library cURL fulfils these conditions with the command line programme curl. The command line programme is integrated in Windows 10 by default and can be used easily on computers with other operating systems such as Linux or macOS.

In addition, the envisioned design should include a data sharing feature to ensure access to data stored in the OMERO repository by external colleagues or reviewers. In OMERO, it is not possible to share individual data with people who are not members of the local OMERO ecosystem without moving the data to the public group and removing any access restrictions. Sharing data with external parties requires downloading and sharing via a file sharing platform. This workflow is very time-consuming and in some cases also limited in terms of file size, which increases both the overall workload and the time required. Sharing data via a URL is a frequently used approach. In this case, the URL to the data is provided with a so-called hash, so that this URL cannot be guessed and only people who are in possession of this link can have read access to the data. These URLs are also so dynamic that simply disabling the URL is enough to block access to the data without having to move or delete the data itself.

To implement and provide the extended download and sharing functionality in OMERO.web, several steps are required: i) The system must be configured according to the components involved so that access to selected data is via secure URLs realised. ii) The download must be parameterized according to the specifications of curl and bundled in the form of a so-called batch file. The user must be provided with a graphical user interface to select the data and the possible settings for the download specification. iii) There must also be a user interface for managing and accessing the pooled data.

3.2.3. Configuration

We take advantage of the architecture of the OMERO platform and use the nginx server of OMERO.web (see Section 2.2.5.7) as a reverse proxy. nginx forwards requests from clients (e.g. web browsers) to one or more internal web servers, hiding the actual web server address. It is therefore possible to direct requests to a dedicated storage area via a specific URL. The nginx server maps this area to a specific URL. In order to activate this mapping, it is necessary to configure the nginx accordingly for the specific storage area (here presented by the variable OPENLINK_DIR) and the provision of the download data within this dedicated area. The configuration of the nginx is a general system configuration that defines, how nginx should handle the client requests [44]. In our example, we use an additional server domain (here the variable SERVER_NAME, which means that we create a new URL namespace as a mapping to the openlink storage area (see Listing 4). To prevent access to the entire content, you can place a file index.html in OPENLINK_DIR to block this location. This file will be displayed instead of the whole content of the directory and may contain an appropriate message if someone wants to access OPENLINK_DIR directly and not just a contained subfolder (example see Listing 5).

```
server {
    listen 80;
    server_name SERVER_NAME;  # url alias to this nginx site

    location /openlink {

        proxy_read_timeout 36000;  # 10 hours
        limit_rate 10000M;  # 10 GByte
        gzip on;
        gzip_min_length 10240;
        disable_symlinks off;  # enable symlinks
        autoindex on;
        autoindex_format html; # html, xml, json, or jsonp
        autoindex_exact_size off; # on off
        autoindex_localtime on; # on off  (UTC)
        alias OPENLINK_DIR/;  # directory containing OpenLink areas
    }
}
```

Listing 4 Configuration file for `nginx` to handle `OpenLink` requests. Access to the storage area `OPENLINK_DIR` is now enabled via http://`SERVER_NAME`/openlink by this `nginx` configuration.

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Omero OpenLink</title>
</head>
<body>

<a href=OMERO_URL>Please go first to the Omero-System to create OpenLinks!</a>

</body>
</html>
}
```

Listing 5 Block location by index.html. Access via the URL directly to `OPENLINK_DIR` is blocked and displays the content of index.html instead.

3.2.4. Implementation

3.2.4.1. Pool Data as `OpenLink` Area

The pooling of the data selected by the user for downloading is implemented by the `Python` script `CreateOpenLink.py` which can be easily integrated into `OMERO.web` and `OMERO.insight` (integration of `Python` scripts see Section 2.2.5.7). We describe in the following on the use of the script integrated in `OMERO.web`.

By selecting, projects, datasets or individual images are markable for download in the `OMERO.web` frontend. After starting a script, the data is assigned to an `OpenLink` area. If required, all linked files in the form of attachments can be added to this `OpenLink` area additionally. The user has the choice of adding data to an existing area or creating a new area.

A user is only authorised to download their own data. Except for group owners of non-private groups, who are allowed to download other group members' data. When the user starts the script with his data input, the further execution is taken over by

a routine of `OMERO.server` (see Figure 3.2). First, the access permissions are checked.

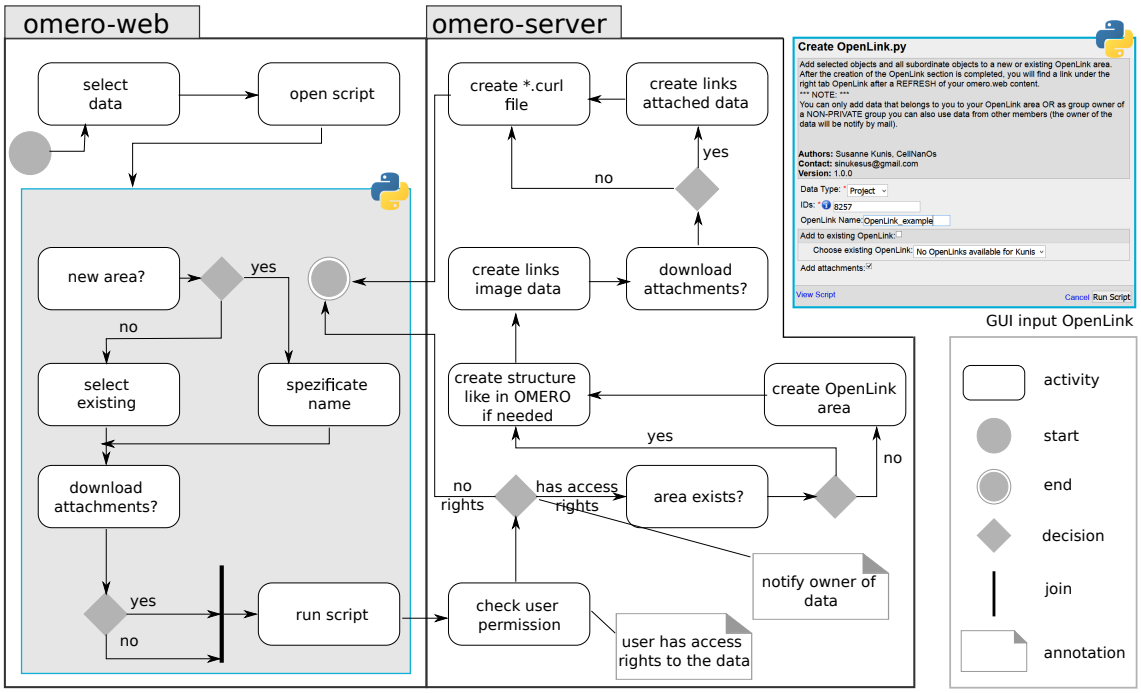

Figure 3.2. Activity diagram for the `Python` programm to generate `OpenLink` areas. Processes started in the `OMERO.web` environment run as omero-web tasks. Processes running in the `OMERO.server` environment run as omero-server tasks.

When downloading data as a group owner and when the data belongs to another user, this user will receive an e-mail notification. If necessary, a new `OpenLink` area is created in the directory `OPENLINK_DIR`. Since the name of the `OpenLink` area is also the entry point of the URL, the `OpenLink` area name specified by the user is combined with the user identification number used in `OMERO` and a randomised hash string. This concept protects the URL from being hacked and thus prevents unauthorised persons from accessing the data. Within the `OpenLink` area, the directory structures are created as they are visible in `OMERO`. Then the corresponding system links to the image containers in the storage system of the repository and, if desired, to files of other formats that have been linked to the data in `OMERO` as attachments are created (Figure 3.3). Finally, a file is created in the `OpenLink` area that can be used for downloading as a batch of the entire area using `curl`. The batch file also contains information about creating the corresponding file structures on the target system. To download the entire area, the user only needs to call up this file with a corresponding `curl` command to start the download. The `curl` command to be used as well as the URL of the generated `OpenLink` area is returned to the user as output when the generation is completed.

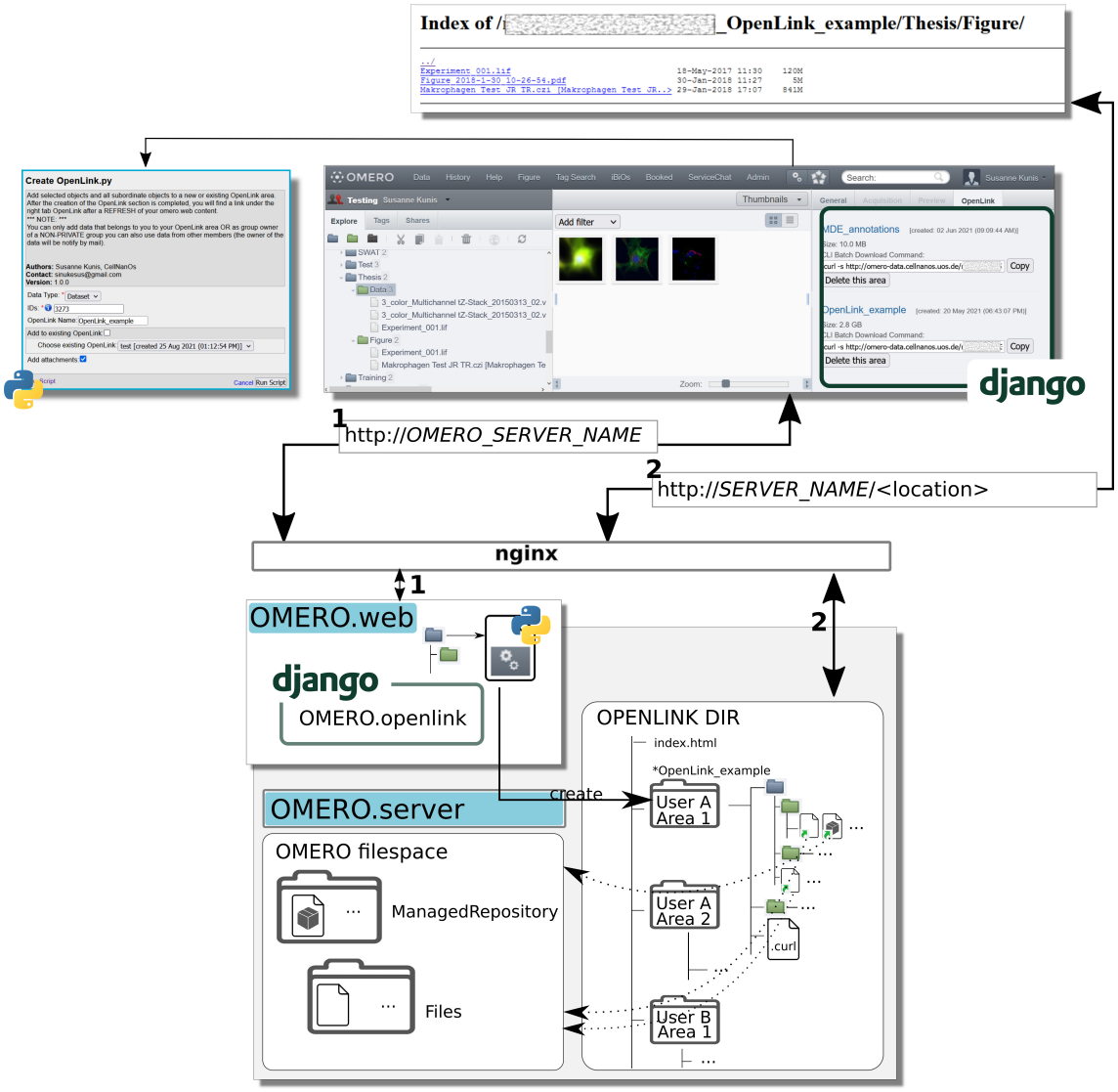

Figure 3.3. Components of `OpenLink`. Data sets are bundled via the `Python` script `CreateOpenLink.py` as an `OpenLink` area accessible via a URL and mapped by the `nginx` server.

3.2.4.2. Manage `OpenLink` Areas

The user can create an unlimited number of `OpenLink` areas. Therefore, it makes sense to provide the user with an overview and a simplified management function. Due to the limited flexibility of the user interface for scripts in `OMERO`, we implement the plugin `OMERO.openlink` in `OMERO.web` (see Section 2.2.5.7). The plugin provides the user with a cross-group overview of the created `OpenLink` areas as links with timestamp and size information. In addition, the corresponding batch download command for executing the `curl` download for copying and pasting into a command line tool is also available there. With `OMERO.openlink` the user can also delete his own `OpenLink` areas. If the user wants to share data from an `OpenLink` area with a user

who does not have access to the OMERO repository, he can share this data with this user by passing on the link.

3.2.5. Evaluation

OpenLink was introduced to provide an improvement in the field of data download in terms of functionality, batch download and download speed. In addition to merging data from different datasets, projects or groups, it is now also possible to download all linked files together. The respective data packages are listed as URLs on an HTML page. This can also be used for file sharing with external partners who do not have access to the OMERO system. The transmission protocoll used by curl optimises the download with respect to the network transfer, e.g. by automatic short-time compression for the data transfer. The user receives a detailed overview of data throughput, duration and status. We can only make a tendential statement or estimate about the acceleration of the download rate, as no idealised system could be used for the evaluation. We measured the data transfer rates during the download on the production system. During the test, side effects can influence the data transfer, such as data transfers from other users on the network, different system loads on the OMERO.server and network loads from other programmes on the local computer. Tests were carried out on two different systems in succession. Test system I is a Windows 8 computer with SSD 850 EVO 250GB and 1Gbit network connection, test system II is a Windows 10 computer with Samsung SSD 970 EVO 1TB and 10Gbit network connection. The download via OMERO.insight and OMERO.web was compared with the download via OpenLink. Since OMERO.web limits the data download to 1GB for the total amount of data and thus only the download of individual files is possible, all three download functionalities can only be compared for large data file sets. We use a data set with a size of 1.3 GB.

	Mbps		Time (sec)	
	I	II	I	II
insight	310	369	36	30
web	601	670	18	16
curl	671	3818	16	2

Table 3.1. Data transfer rate. Rate in Mbps and data transfer time of a 1,3 GB file (averaged over three runs in each case).

The evaluation (see Table 3.1) shows that both the download via OMERO.insight and via OMERO.web cannot use the possible data transfer rate accordingly. The download via OpenLink using curl, on the other hand, clearly shows the dependence on the available network capacity. OpenLink, with its download via curl, can make much better use of network capacity, while OMERO.insight and OMERO.web seem to be limited by implementation factors and extended network capacity has no impact on download speed.

3.3. MDEmic - Metadata Editor for Microscopy Data

The following chapter describes a completely new designed software for metadata annotation MDEmic, which was integrated into the existing OMERO.importer as OMERO.mde. First, the conceptual and functional realisation of the requirements is described, followed by the implementation part with a description of the requirements in detail as well as the interaction with OMERO.importer. Subsequently a configuration example and a use case scenario will be presented. Some parts of this chapter are based on the openly available documentation of OMERO.mde [49].

3.3.1. Related Work

Data management with OMERO opens up various possibilities for gathering metadata. One possibility is to link files containing metadata to the image data. Such files are either generated by the microscopy system during the acquisition process or can be created using the export function of microscopy systems. However, the metadata they contain are very heterogeneous in terms of their scope and the vocabulary used. The manual collection of metadata in a text-based format is also popular. The advantage of this is the intuitive use. However, the user cannot directly access the metadata visually within the management system, because the corresponding file must be opened in order to view this metadata. Another way to capture metadata is to enter free text in the form of key-value pairs. By specifying the documentation format within the working group, it is also possible to strive for standardisation. However, the amount of metadata generated in bioimaging cannot be covered by these methods, since neither the data already documented in the image containers are taken into account, nor a relationship between the data can be mapped without losing clarity. There is also a lack of mechanisms to restrict the annotation with regard to the vocabularies and formats used, so that incorrect or misleading information can be specified. The OMERO.forms [50] software tries to address these problems, but does not map the relationship of metadata to each other, nor does it allow the integration of ontologies. The application of OMERO.forms is exclusively designed for experimental metadata. Python tools that enable the acquisition of metadata in OMERO using command-based methods, such as omero-metadata [51] or PyOmeroUpload [52], cannot be used due to the user structure. However, these tools also capture metadata without standardisation mechanisms.

3.3.2. Design and Conceptional Framework

Storing the image data in the OMERO repository could facilitate the re-usability of the data if appropriate documentation of the images by metadata and linked information are available. On the one hand, the user gets a better structure for organising the data. On the other hand, this standardised structure increases reproducibility and enables subsequent use with differentiated questions, for example by artificial intelligence algorithms. In microscopy, not only technical information regarding image acquisition is important in terms of reproducibility, but also data on sample preparation or descriptions of the data content. For reasons of comprehensibility and to avoid redundancies, all information relating to an experiment should be

linked together. This in turn increases the retrievability and interpretability and thus ultimately the re-usability and comparability of the data. A common standard is needed for improved linkage and interpretion of metadata. The standard can specify which metadata are collected and to what scope, but also how and in what context they are formulated (see Section 2.1.5). Due to domain and sub-domain specific applications, different names for one and the same fact or different spellings cannot be avoided. For these cases, it is possible to establish a clear reference between the terms (like crosswalks in ontologies) if suitable structures are used. The considerable amount of metadata for an experiment in microscopy rapidly leads to a high documentation effort. In order to minimise the increased workload and the resulting psychological barriers for the scientists, as well as to ensure a minimum standard in documentation, it is necessary to offer appropriate software support. Aspects such as the use of existing technical know-how on site, simple configuration or adaptation to new techniques and standards, and re-usability minimise the documentation effort in total.

The earlier the metadata of an experiment are captured in the scientific workflow, the more complete and less error-prone it is. Along the way, project-specific data management requirements are achieved with less effort. For this reason, we integrate a metadata editor into the process of data transfer to the repository via `OMERO.importer`. After selecting the data to be transferred, it is possible to edit metadata in an intermediate step. The metadata can be edited for files individually or for all files in a folder in the batch. The requirements from the previous explanations lead to the following key points in the conception and functionality of the planned software solution for metadata annotation, which we divide into the points user interface, interoperability and configuration.

3.3.2.1. User Interface

Overview of metadata stored in image container This is a domain-specific requirement as microscopy data formats contain metadata information about the specific technical aspects. Due to the many different proprietary data formats in microscopy and the ongoing technical development, it is not always possible to read out and harmonise these original metadata correctly from the image containers of the microscopy data formats using Bio-Formats. In order to give the user an impression of the support of a particular image format, it is necessary to visualise this harmonised metadata.

Displayed metadata are editable All metadata displayed in `MDEmic` are editable and each image can be annotated separately. In addition, the metadata entered for a folder item is inherited by each image file contained in the folder.

3.3.2.2. Configuration

Facilitate standardisation The metadata input form and referenced vocabulary in `MDEmic` are based on an underlying standardised metadata schema. The schema and vocabulary can be customised and stored in a modified version. The modified

metadata forms are reusable institution-wide, but the individual user still has the option to further customise the structure of the schema at runtime.

Sets of valid terms are specifiable for the metadata values from which the user can select the appropriate value. This counteracts the use of different identifiers and different spellings. It is possible to create such a list automatically by referring to an ontology class. In this way, community standards regarding vocabulary is automatically integrated as a first step in direction of linked data annotation.

Modification the underlying data model is possible The OME Data Model is a metadata schema for technical aspects of microscopy. We have integrated this schema into the implemented software `MDEmic` as the basic metadata schema, which is visualised in the GUI as a tree structure. However, it represents a specific technical point in time of development. New technologies and their architectures might require an adaptation or extension of this schema. Therefore, the tree structure can be extended by further objects in the form of new nodes.

Integration of prefilled metadata The technical configuration of the local microscopes in the image processing facilities is well known by the technical management. This knowledge can be integrated into `MDEmic` in the form of prefilled objects and is accessible to the user for application when capturing the metadata.

Simple adaption and configuration Both the definition of new objects and their properties, the relation to other objects, and the prefilled of values are realised via a configuration file. This file also contains the thematic grouping of the elements in the form of setups. The configuration is not defined in the source code and can also be carried out by people without programming experience.

Filter View The visibility of the metadata and values can be configured. In detail, which objects and which subordinate metadata are visible, but also which prefilled values. For example, it makes sense to hide objects of the OME Data Model that are not relevant for a microscope in order to limit the user's input to the important metadata. `MDEmic` is able to focus on metadata that needs to be collected urgently. This partial view of the metadata finally increases the clarity for the user.

3.3.2.3. Interoperability

Different metadata output formats With the softwaretool, metadata can be stored directly in different file formats as well as in different styles such as formulated as linked Data for the given ontology classes. The representation both in the format of the keys and the values vary. All captured metadata is stored in the `OMERO` database with the import step and automatically linked to the corresponding image data.

Reuse of inputs Since in microscopy experiments are often repeated with slight modifications, it is possible to use metadata inputs that have been collected once

repeatedly.

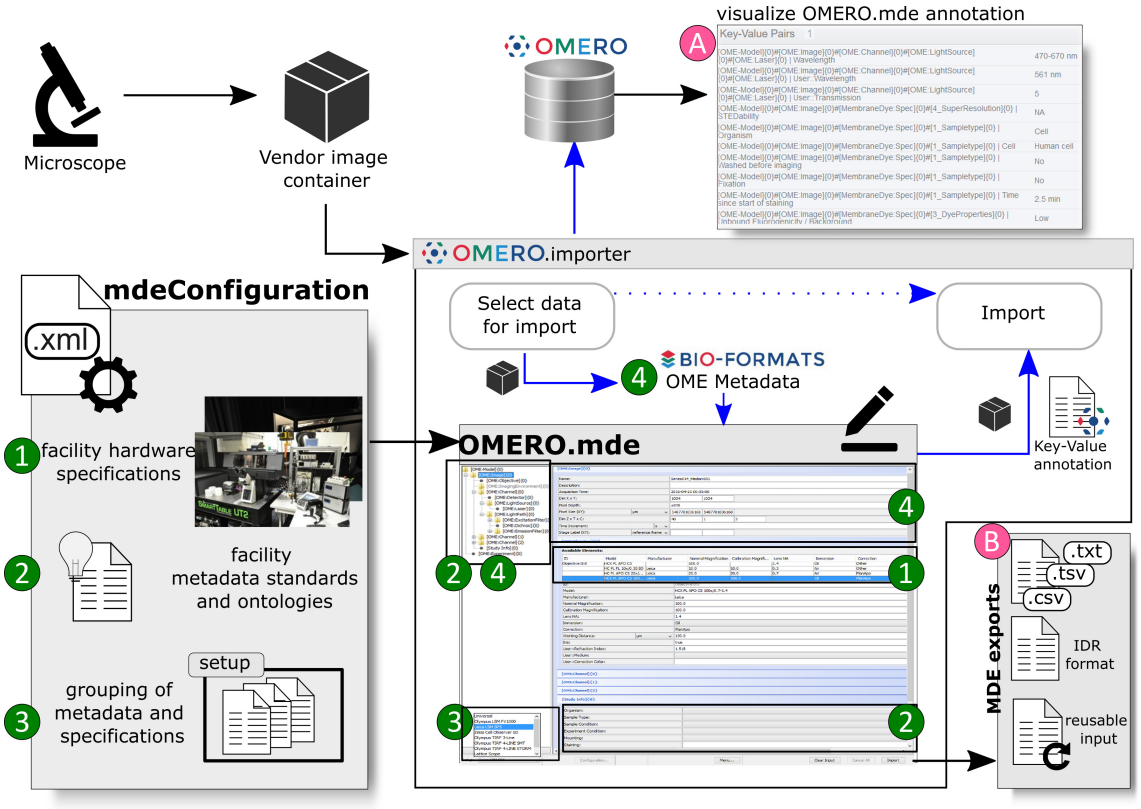

Figure 3.4. `OMERO.importer` **with integrated** `MDEmic` **as** `OMERO.mde`**.** In the `OMERO.importer` the `MDEmic` is integrated as an intermediate step for the selection of data for the import and the import itself. Metadata can now be added, which is then transferred to the repository together with the image data **(A)**, or the annotations can be exported in different formats in this step **(B)**. The `MDEmic` can be customised via a configuration file and loads the specifications from this file dynamically when the `OMERO.importer` is started (**1,2,3**). All technical metadata of the images marked in the previous step of data selection are read out by Bio-Format(**4**) and provided as values in the `MDEmic` respectively [3].

As already mentioned, the software component `MDEmic` is integrated into the data transfer process to the repository with the functions mentioned above. `OMERO.importer` guides the user through the process of transferring the data to the repository. The process is divided into two steps: firstly, the selection of images for import with indication of the storage location and secondly, the transfer of the images and post-processing. `MDEmic` has been integrated as a subcomponent and intermediate step in `OMERO.importer` (Figure 3.4) as `OMERO.mde`. To keep possible effects on the original software as small as possible, the interfaces were reduced to a minimum and the metadata annotation step could be skipped if necessary.

3.3.3. Implementation

The first step of data selection in the `OMERO.importer` forwards the list of data selected by the user for transfer to the repository to `OMERO.mde`. If the user adds or corrects metadata using `OMERO.mde`, these metadata are forwarded to the import process of the `OMERO.importer` as a key-value map with a link to the corresponding raw data. The component *OmeroImageServiceImpl*, which is responsible for data transfer to the server,has been extended to integrate this additional metadata into the transfer to the server, where it is linked to the respective image data (see Figure 3.5).

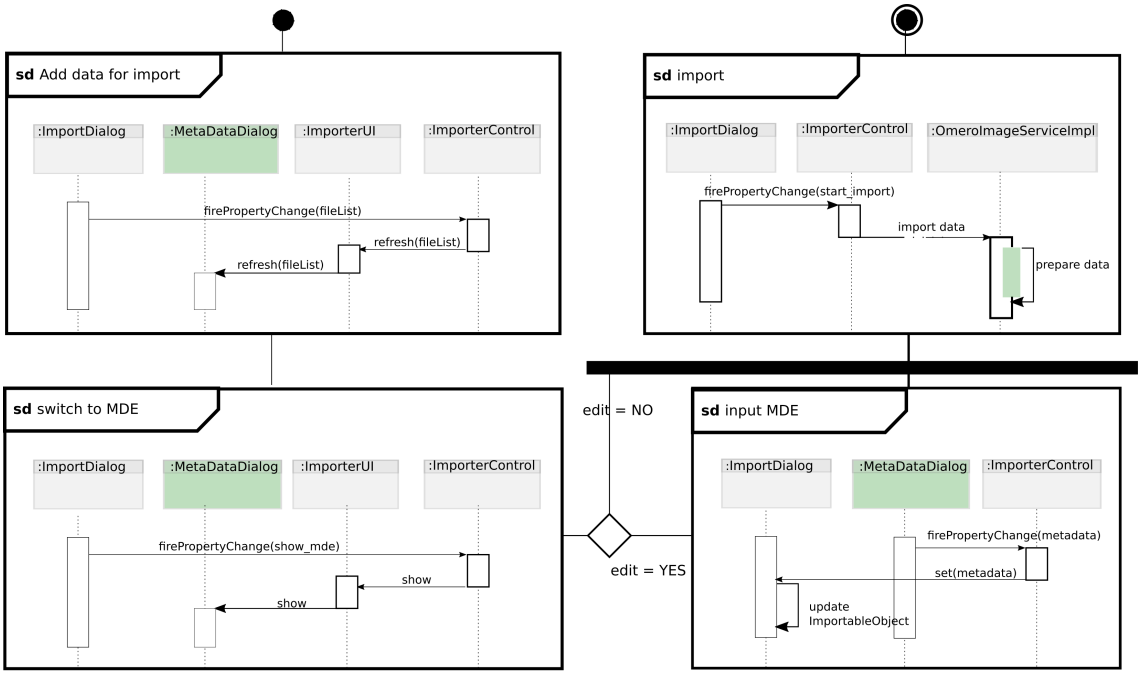

Figure 3.5. Interaction overview diagram `OMERO.mde`. The interaction of `OMERO.mde` (green) with components of the `OMERO.importer` (grey) is limited to a few classes and only takes place at certain events, such as adding or removing data from the import queue, switching to the input form panel and entering data into the forms. In addition, a subroutine is implemented in the component *OmeroImageServiceImpl* which integrates the additional metadata created by `OMERO.mde` into the already existing annotation object for the transfer step.

The following section describes the specific implementation of the requirements listed in the previous section and outlines the organisation of the code, components and modules.

3.3.3.1. User Interface

Overview metadata stored in image container

`OMERO.mde` displays the import queue originating from the `OMERO.importer` as a file tree. The user can now select a specific file in this import queue browser (Figure 3.6a).

The *MDEContent* panel of `OMERO.mde` (Figure 3.6b) displays all information related to the selected object in the file tree browser. For files, the metadata are read from

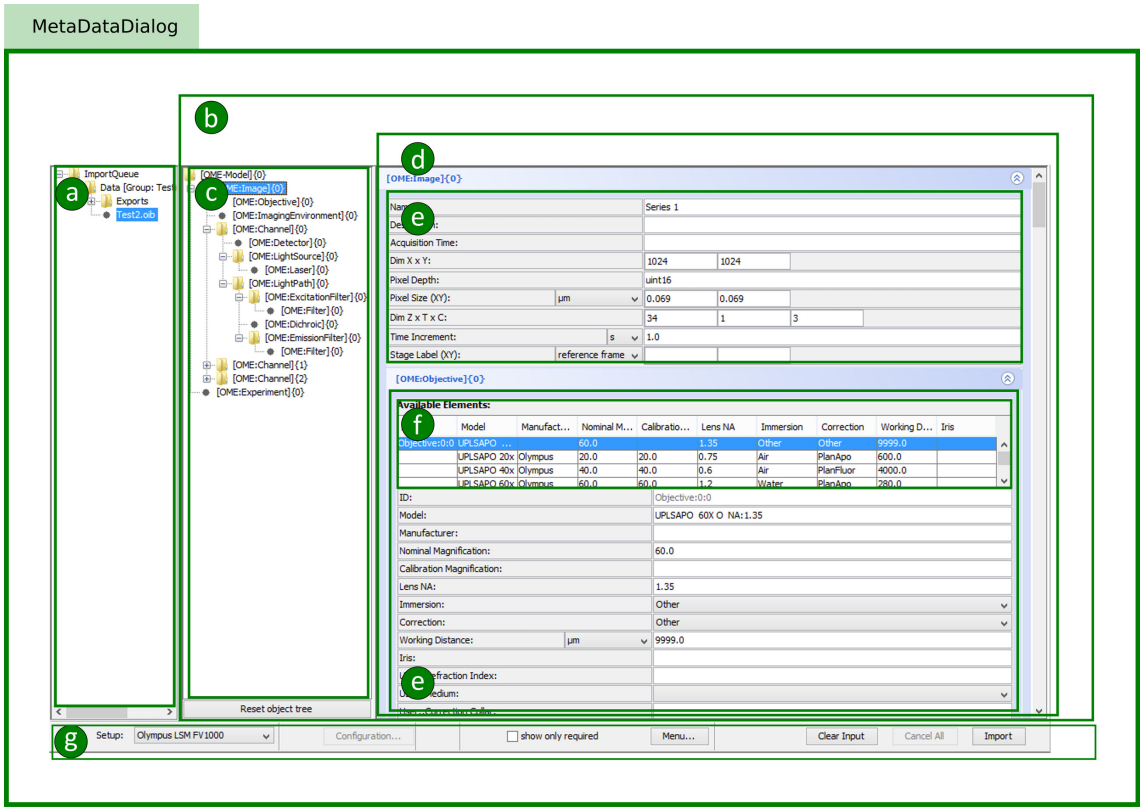

Figure 3.6. Graphical user interface components and classes of `OMERO.mde`. The graphical user interface of `OMERO.mde` consists of the main component *MetaDataDialog* with the following sub components: **(a)** File tree component (*FileTree*) visualise the import queue structure. **(b)** Related data for selected object in file tree (*MDEContent*). **(c)** Object tree component (*ModuleTree*) visualise metadata objects and their relation as tree structure. **(d)** Collection of input forms of selected object and child objects (*ModuleContentGUI*). **(e)** Input form for specific object (*ContentViewer*). **(f)** Table of `Available Elements` contains defined objects that are read from the image file and the configuration file. **(g)** Button bar with export, setup and filter options.

this selected image container by Bio-Formats and converted into OME metadata by mapping to the OME Data Model and displayed in a structured way (Figure 3.7). For this purpose, a graphical visualisation as a tree structure of the OME Data Model was adopted in the graphical user interface (Figure 3.6c), which is referred to as the object tree in the following. Each node of the tree is linked to the corresponding input mask, which displays the properties of this node in the form of metadata keys-values (Figure 3.6e). The metadata key corresponds to the label of an input field. The value corresponds to the content of the input field and is configured by default according to the type defined in the OME Data Model. If a unit is assigned to the value by definition, it is integrated in the label field as a dropdown form. The predefined displayed units correspond to the definitions in the OME

Data Model. All value fields are editable. The values read out and harmonised by Bio-Format are automatically entered into the fields. The available objects of an object class are specified in the table of `Available Elements` (Figure 3.6f) with the specific identifier assigned by Bio-Format. In summary, this provides a clear overview of which metadata of the image format is supported and can be extracted by Bio-Format to be available later in the target repository.

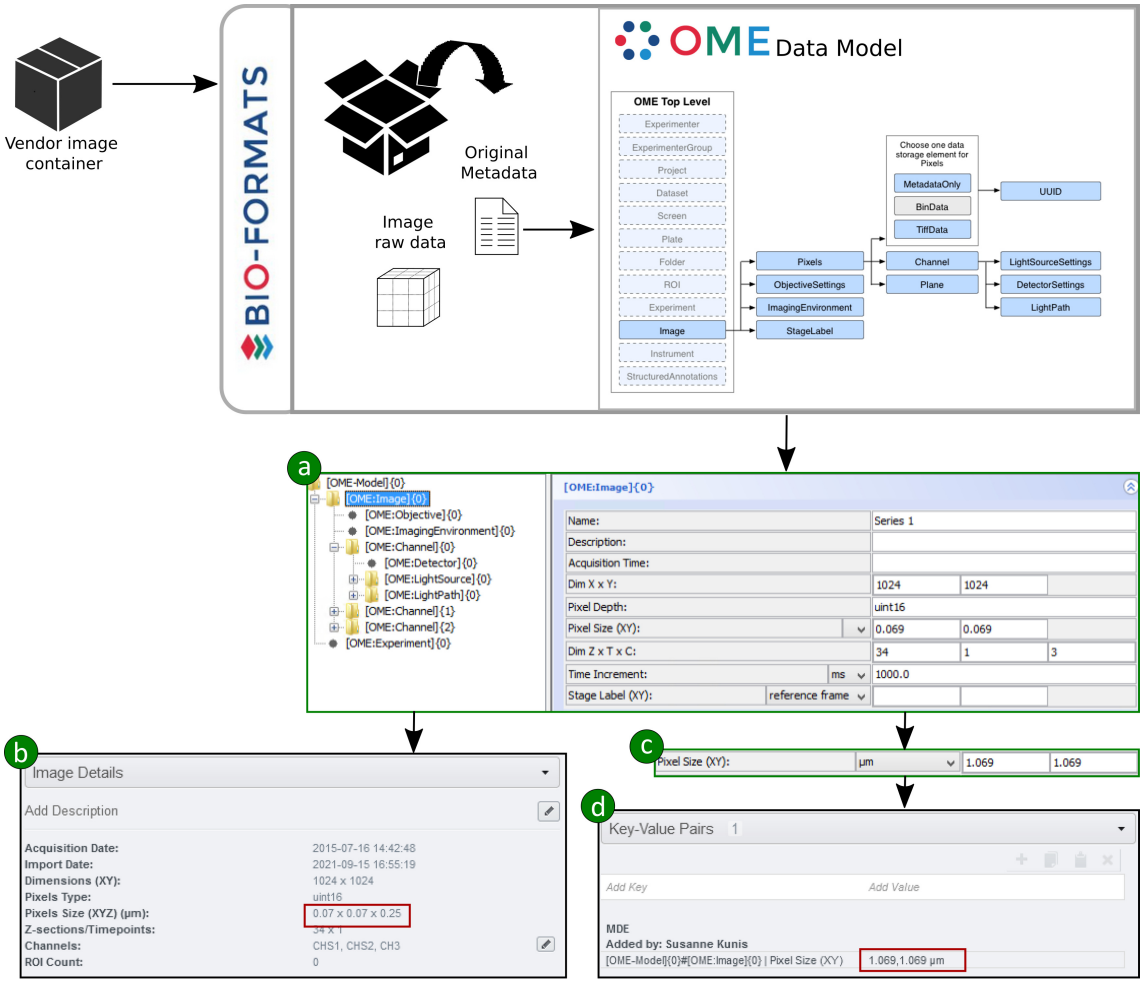

Figure 3.7. Metadata and their corresponding representation in `OMERO.web`. **(a)** Default object tree based on OME Data Model with example of input mask for object node `OME:Image`. **(b)** Corresponding representation of `PixelSize` after data transfer via `OMERO.importer` as OME metadata. **(c)** Corrected value for `PixelSize` by input in `OMERO.mde` and **(d)** its representation as key-value pair after transfer.

Displayed metadata are editable

All displayed metadata values and units can be edited. The user has the possibility to change the unit and to add or change values in the input fields. If the user changes the specified unit, the corresponding value is calculated accordingly. There

are different input form types for the value field to support editing. For example, values with fixed term vocabulary are displayed as a dropdown form field or there is an automatic format control for date and time fields. The units of measurements are displayed as a dropdown field. The use of specified fixed terms for metadata annotation as well as input control ultimately increases the quality and validity of the entered data through appropriate harmonisations.

To simplify the gathering of metadata for multiple files, a parent structure, such as a directory, can be annotated and all child files inherit this edited values.

3.3.3.2. Configuration

Facilitate standardisation

The initial configuration of `OMERO.mde` is based on the metadata schema of the OME Data Model, which is widely used in the context of microscopy image data. The data model is visualised as a tree structure, the so-called object tree. The individual nodes of the tree each represent grouped metadata in the form of objects. Data model and used metadata vocabular can be customized on different levels (Section 3.3.3.2) with the help of a configuration file of `OMERO.mde`. The advantage of this method is that these changes are usable by different users of the `OMERO.importer` in saved form. In iBiOs, for example, the `OMERO.importer` starts from a central instance. The loaded associated configuration allows facility-wide access to differently modified schemes. In this way, the standardisation of metadata annotation can be implemented across groups or institutions.

Modification the underlying data model

To meet the technical changes in particular, the object tree is modifiable in two different ways. On the one hand, various modified object trees defined in the configuration file can be loaded. On the other hand, the object tree is adjustable by the user during runtime via a pop-up menu. Manual modification at runtime includes the addition or deletion of already defined objects or nodes of the object tree. This concerns all objects of the OME Data Model, but also objects that were previously defined in the configuration file (see Figure 3.8a). Modifications to the object tree made via the pop-up menu cannot be saved in their structure for reuse and serve more as a proof-of-concept approach to test new schemes or models.

By specifying the object tree structure in the configuration file, a certain object tree architecture is available via a setup. In this way, different data models or structures are accessible for different user groups across institutions. In addition, objects can also be specified for this purpose, in order to be inserted into a structure manually by the users later on. Certain relationships can be defined in advance by specifying the possible parent nodes for an object. This ensures a minimum standard for the manual configuration of the metadata structure by the user. In order to be able to reconstruct the adapted structure from the database after saving it as a key-value annotation, the path in the object tree to an object is also saved in the key. At the metadata level, individual fields of the metadata can be hidden. The definition of corresponding values of metadata is done as a set of fixed terms or by

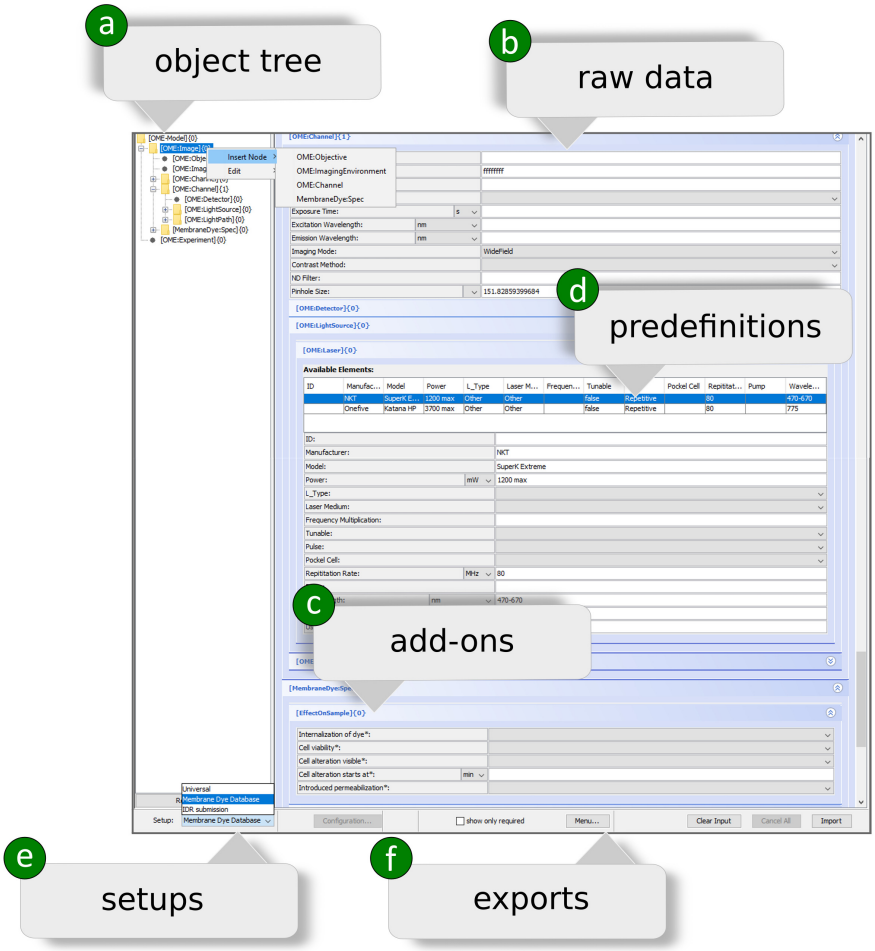

Figure 3.8. Graphical user interface `MDEmic`. a) The object tree is based on the structure of the OME Data Model and can be extended by objects of this data model or self-defined objects. Objects are a collection of related metadata. The extension can be done manually at runtime by using the context menu or by selecting the appropriate setup from the configuration file. **b)** The technical metadata stored in the image file is read using Bio-Formats and provided in the input forms as editable values. **c)** The user can define new objects with defined metadata keys and a selection of predefined values by specifying them in the configuration file. These values can also be loaded automatically by reference to an ontology class. **d)** Predefined metadata can be specified for all objects in the configuration, which the user can choose from. **e)** A setup is a bundle of data model modifications, input form configurations and/or various associated predefinitions (such as hardware definitions of a microscope setup or an experiment protocol). **f)** All metadata can be exported directly to a text file or as a reusable template for later annotation using `MDEmic` [3]

means of a link to an ontology class (see Figure 3.9). Saving the different modified models as selectable setups makes it easy to switch between them.

Integration of prefilled metadata

Under `<MDEPredefinitions>` (see Listing 6), the prefilled metadata values for objects can be defined in the configuration file. The grouping of several objects in a setup then results in a microscope-specific overall picture. The user can load these prefilled values in the `Available Elements` list by selecting the corresponding setup (see Figure 3.8d). For example, a setup may present a specific microscope with all its associated components. The technical hardware components are then prefilled, such as a specific internal detector. If the user now selects this setup, in addition to the objects extracted from the image container itself, the prefilled local objects of this microscope are also displayed in the list of `Available Elements`. In our example, the user can compare the metadata of the extracted detector from the image file with the metadata of the prefilled detector coming from the configuration. If necessary, the user the user is now able to complete the detector metadata that could not be extracted by clicking on the predetermination.

Simple customization and configuration

The configuration of the `OMERO.mde` is done via an XML-based file. By configuring using an additional file, the adjustment does not have to be implemented in the source code and therefore also be managable by people without programming experience. Being designed in XML, a textually structured schema, it is easy to understand even for non-experts. When `OMERO.importer` is started, this configuration file is loaded and `OMERO.mde` is dynamically adapted according to these specifications. The configuration options include (see Listing 6):

- `<Definitions>`: Definitions of objects and their (metadata)p roperties and the relationship to other objects,
- `<MDEPredefinitions>`: Different presetting of metadata values for an object, assigned to a specific setup,
- `<SetupPre>`, `<SetupConf>`: Setup related groupings,
- `<Configurations>`: Configuration of object tree, objects and properties belonging to an setup.

An object is defined by its type, the associated metadata as properties (`<TagData>`), and possible parent nodes `<Parents>` for this object in the metadata structure. The metadata definitions consist of the name, which is displayed as the name of the input field, the input field type, the default unit, and the default values. Different types are available for the input field and can be specified as listed in Table 3.2.

By assigning objects and their configuration to individual setups, it is possible to switch between different metadata models and prefilled values of objects, e.g. for different microscopes, during runtime (see Figure 3.8c,e).

Filter view

The input forms for metadata are configurable in various ways and can each be saved as a setup. A setup is therefore the implementation and restriction to a specific

TextField	`<TagData DefaultValues="" Name="Tag of Type TextField" Type="TextField" Unit="" Value="" Visible="true" />`
TextField with unit	`<TagData DefaultValues="" Name="Tag of Type TextField with unit" Type="TextField" Unit="nm" Value="" Visible="true" />`
TextArea	`<TagData DefaultValues="" Name="Tag of Type TextArea" Type="TextArea" Unit="" Value="" Visible="true" />`
ArrayField with 2 separate input fields	`<TagData DefaultValues="2" Name="Tag of Type ArrayField" Type="ArrayField" Unit="" Value="" Visible="true" />`
ArrayField with unit and 3 separate input fields	`<TagData DefaultValues="3" Name="Tag of Type ArrayField with unit" Type="ArrayField" Unit="s" Value="" Visible="true" />`
ComboBox	`<TagData DefaultValues="Value1,Value2,Value3" Name="tag of Type ComboBox" Type="ComboBox" Unit="" Value="Value1" Visible="true" />`
CheckComboBox for multiple selection	`<TagData DefaultValues="Value1,Value2,Value3" Name="Tag of Type CheckComboBox" Type="CheckComboBox" Unit="" Value="Value1" Visible="true" />`
TimeStamp	`<TagData DefaultValues="" Name="Tag of Type TimeStamp" Type="TimeStamp" Unit="" Value="" Visible="true" />`
Combobox with linked ontologie values	`<TagData DefaultValues="" Name="Tag of Type ComboBox val from ontology href" Type="ComboBox" Unit="" Value="" Visible="true"> <Ontology URL_restapi="http://data.bioontology.org" Acronym="EFO" ID_href="http://purl.obolibrary.org/obo/OBI_0000272" /> </TagData>`

Table 3.2. Input field types `MDEmic`.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
   <MDEConfiguration>
       <!-- Predefinitions for properties of objects group by setups-->
       <MDEPredefinitions>
           <SetupPre Name="SetupName">
               <ObjectPre>
                   <TagData.../>
                   ...
               </ObjectPre>
               ...
           </SetupPre>
           ...
       </MDEPredefinitions>
       <MDEObjects>
          <!-- Specification of objects, properties and hierarchy-->
           <Definitions>
               <ObjectDef Type="ObjectName">
                   <TagData.../>
                   ...
                   <Parents.../>
               </ObjectDef>
               ...
           </Definitions>
        <!--Definition of setups and configuration of objects and
       properties belonging to the setup -->
           <Configurations>
               <SetupConf Name="SetupName">
                   <ObjectConf Type="ObjectName">
                       <TagDataProp.../>
                       ...
                   </ObjectConf>
                   ...
               </SetupConf>
               ...
           </Configurations>
       </MDEObjects>
   </MDEConfiguration>
```

Listing 6 `OMERO.mde` configuration file structure.

view or specific data. On the one hand, metadata or entire objects is individually configurable in terms of its visibility. On the other hand, metadata can be marked as particularly important or required. Such metadata are marked with a `*` in the metadata input form labels. By activating the option `Show only required` in the button bar, the metadata overview in *MDEContent* pane is limited to objects that contain metadata marked as required. The displayed list of predefined values in the table `Available Elements` is filtered by objects according to the setup assignment. The content of the input forms ultimately refers to the selected image file and the configuration according to the selected setting.

3.3.3.3. Interoperability

Different metadata output formats

Metadata read out from the image (raw metadata) that was updated or added in `OMERO.mde` is stored as a new key-value pair. In the image container itself, however, the metadata is not overwritten so that the original data is not changed and data integrity is ensured. The `OMERO` platform stores all metadata read by Bio-Formats in a database after transfer to the repository. This data is displayed as OME metadata in the user interface of `OMERO.web` and `OMERO.insight`. In addition to the harmonized representation of the metadata, the user has access to a list of the original metadata. We do not overwrite the corresponding value of the OME metadata in the repository database of `OMERO`, as these changes would not be traceable later and would lead to irritation for the user if the displayed OME metadata and the original metadata differ. Instead, all metadata collected with `OMERO.mde` are stored in `OMERO` as key-value annotation (Figure 3.7d). By exporting the image data container into the widely used standardised file format OME-TIFF from the OMERO system, metadata gathered in addition to the OME metadata and the original data can be integrated into an image container.

`OMERO.mde` offers different file formats for the export, like different text-based formats but also a format specialised for IDR publications, which saves the ontology references as well. By default, only the edited metadata is exported, but the user can choose to export all existing metadata, including those read from the image file. This functionality enables interoperability with other research data management tools to promote integration with other data types[3].

Reuse of inputs

Another output format is the template export. This exports metadata to a template file that can be selectively reused for subsequent metadata annotations by selecting the required objects to reload. These templates are also usable for annotations of datasets already in the repository via a corresponding `Python` script provided in `OMERO`.

3.3.4. Example Configuration

In Figure 3.9 we present an example for such a customization of `OMERO.mde`. The corresponding configuration in xml can be found in Appendix B. We have defined an `Available InputFields` object with the possible input field types describe above as metadata fields. We linked the class `protocol` from the Experimental Factor Ontology (EFO) to load a list of fixed term values from its subclasses for the example field `Tag of Type (Combobox val from ontology href)`. The values of the associated sub-nodes of this class are loaded as a list of possible terms that can be selected for this metadata mapping. In addition, we have added a possible value assignment for the object `Available InputFields` and associated it with the setup `Example Setup: Fields`. This default value is displayed in the related table `Available Elements`. Furthermore, we have associated a modified object tree to this setup. If you now switch to this setup in `OMERO.mde`, the corresponding configurations are loaded in the *MDEContent* area.

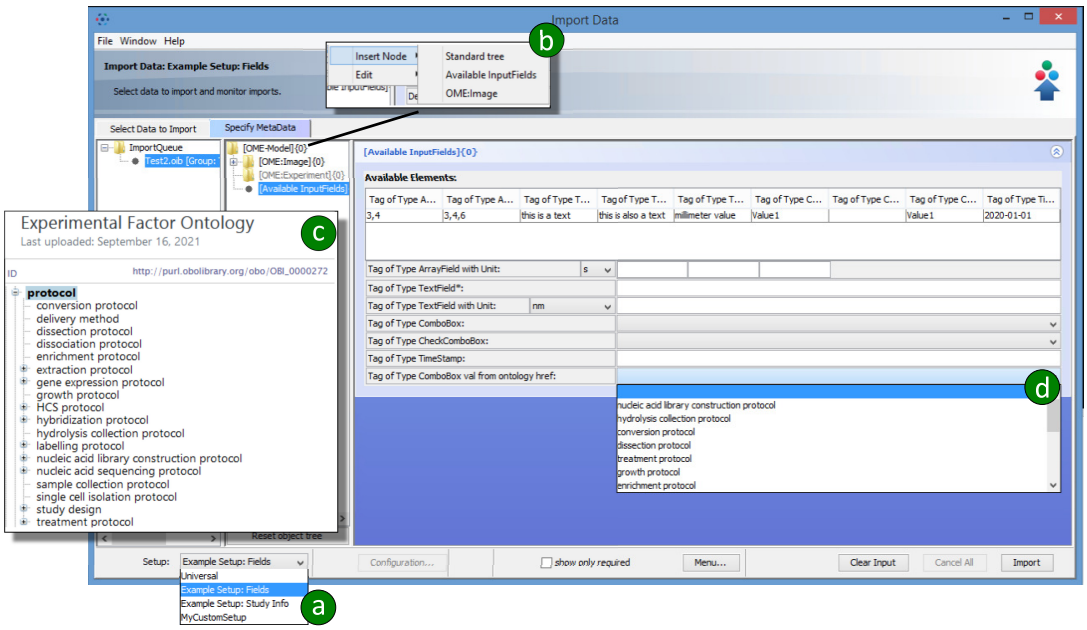

Figure 3.9. `OMERO.mde` customization and configuration. **(a)** The sample configuration includes user-defined setups `Example Setup:Fields`, `Example Setup:Study Info`, and `MyCustomSetup`. The selected setup `Example Setup:Fields` contains a specification of the new object `Available InputFields`, which has been integrated into the object tree assigned to setup and is now also available in the object tree configuration popup menu **(b)**. Various metadata fields have been defined for `Available InputFields`. By associating a particular ontology class with a metadata value field specification, the corresponding values were generated from the subclasses of that ontology class **(c)** and are available to the user for selection **(d)**.

3.3.5. Example Worflow

As an example of a workflow, we describe a collaboration project together with the collaborative research centre (CRC) at the University of Düsseldorf to build a membrane dye database as a use case scenario. Within the framework of the institutional image repository `OMERO`, image data is made available to the members of the research field after application of different membrane dyes. Microscopy techniques are used to study very different biological samples with regard to the identity and dynamics of membrane systems. The samples include mammalian cell cultures, plant tissue and fungi as well as bacteria. In addition to the visualisation of the different data, further descriptions about the performance of the membrane dye in the respective sample and under the respective imaging conditions are available in the image database for membranes, in parallel to the technical metadata. To capture the description of these metadata uniformly within the research team, we use `OMERO.mde` with a suitable configuration. To do this, we define a metadata object `Membrane Dye` in the configuration file and assign various other newly defined objects, such as `Effects On Sample`, to this object. Together with the technical description of the lasers used in the research institute, we make these input forms available under the setup `Membrane Dye Database` (Figure 3.10).

This configuration of `OMERO.mde` can be done on site by a data steward or a researcher from the imaging core facility. If a user wants to import image data into the membrane database, he or she is guided to enter membrane-specific information after the data selection step. After the transfer to the repository, the complete metadata set is immediately available in the membrane dye database and, via the search functionality, the researchers have access to all mebrane-specific information, which is now available as standardised key-value pairs.

This use case describes a clearly defined purpose and is therefore suitable for introducing such a tool as `OMERO.mde` and the intention to capture standardised metadata. By collecting metadata using controlled terms and values and in a defined context, we achieve harmonisation of metadata at different levels.

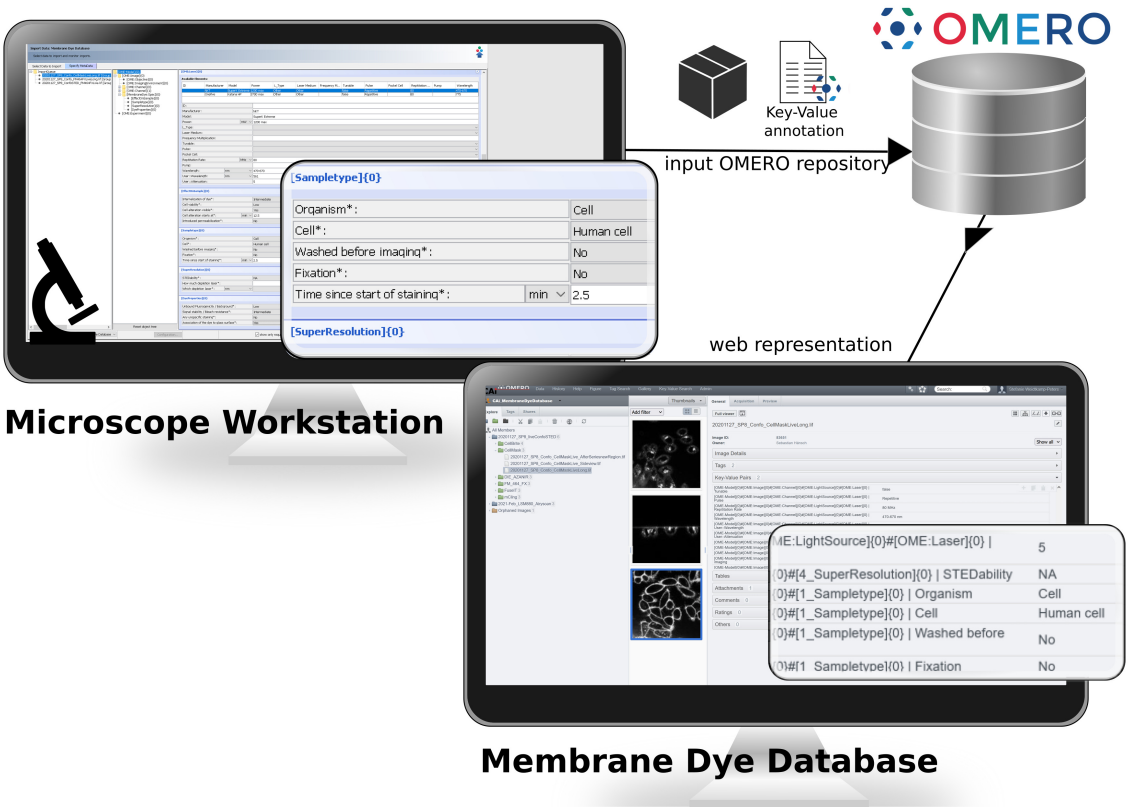

Figure 3.10. The workflow from microscopes to (tailored) `OMERO` repositories. After data acquisition on the microscope, start the software tool (`OMERO.importer`) for data transfer to the Membrane Dye Database. For the selected data, the metadata contained in the data container are read out using Bio-Formats. Now the required metadata for a membrane dye dataset can be added using the given input mask. Finally, the data and the added metadata are transferred to the Membrane Dye Database [3].

4 Conclusion

4.1. Summary

In this thesis, we are working on improving the existing data management environment for microscopy data in an imaging facility. First, a broad background was analysed to gather requirements and select relevant techniques and models that we can use to most effectively improve the existing environment.

Regarding the data download, we managed to overcome the system restrictions in terms of data flow and substantially improve it by means of an independent software extension. The implementation allows for further customisation in all respects (see Section 4.2.1). The benefit of this software solution is immediately apparent to the user through faster download speeds and its functionality as a file share system.

We have acquired metadata concepts as well as the technical implementation of these concepts and implemented them within the framework and possibilities of the existing data management system. Since the field of image data management is currently undergoing a very strong development process, especially with regard to annotation with metadata, the software solution was also implemented with a view to integrating new models and technologies. In a next step, the direct use of metadata annotations is demonstrated to the users by means of corresponding use cases. For research consortia at the institutional level, such as a Collaborative Research Centre, high-quality standards for metadata are extremely valuable, as data can be shared more easily within a given research area and seamless knowledge exchange is ensured. This standardisation is made possible by the software solution we have integrated. It also lays the foundation for offering promising technologies such as RDF and their corresponding serilisation in a user-friendly way (see Section 4.2.2).

4.2. Outlook

4.2.1. `OpenLink`

Possible improvements to the software solution described could be to increase the user-friendliness of `OpenLink` by merging the two modules into one plugin for creating, extending and managing the bundled data. However, this was deliberately refrained from in this work, as such a module implemented as a web plugin would no longer allow the creation of `OpenLink` areas under `OMERO.insight`. `OMERO.insight` is, however, preferred by some users. Nevertheless, a next step is to make it possible to

call `CreateOpenLinkArea` from the webplugin as well. An enrichment of the `OpenLink` functions will be the download of metadata and ROIs as well as starting the `curl` functionality on the local computer without using the command line. A restructuring of the data during download or an export of the data format is another promising functionality to achieve compatibility with structures of other communities and to merge multimodal data.

4.2.2. `MDEmic`

Possible improvements to the functionality of `MDEmic` consist in the further development of the key-values used to capture metadata in RDF triples. The corresponding visualisation of the resulting graph can represent the metadata relationships much better than a tree structure. However, since RDF cannot be captured in a suitable form by the `OMERO` platform, a first step is to export it into a serialisation format of RDF, such as JSON-LD. Another useful functionality would be the integration of text-based files and their metadata into the data model used. This can be realised site-specifically via the configuration file.

Bibliography

[1] Susanne Kunis. *OMERO.openlink*. [Online; accessed 21. Oct. 2021]. Oct. 2021. URL: https://github.com/sukunis/OMERO.openlink/releases.

[2] OME. *Releases omero-insight*. [Online; accessed 19. Oct. 2021]. Oct. 2021. URL: https://github.com/ome/omero-insight/releases.

[3] Susanne Kunis et al. "MDEmic: a metadata annotation tool to facilitate management of FAIR image data in the bioimaging community". In: *Nat. Methods* 18 (Dec. 2021), pp. 1416–1417. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01288-z.

[4] Alex Rigano et al. "Micro-Meta App: an interactive tool for collecting microscopy metadata based on community specifications". In: *Nat. Methods* 18 (Dec. 2021), pp. 1489–1495. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01315-z.

[5] Susanne Kunis et al. *RDM4mic working group– Research Data Management for microscopy data as a community task*. [Online; accessed 20. Sep. 2021]. Sept. 2021. DOI: 10.22443/rms.elmi2021.187.

[6] Glyn Nelson et al. "QUAREP-LiMi: A community-driven initiative to establish guidelines for quality assessment and reproducibility for instruments and images in light microscopy". In: *J. Microsc.* 284.1 (Oct. 2021), pp. 56–73. ISSN: 0022-2720. DOI: 10.1111/jmi.13041.

[7] NFDI4BIOIMAGE. *Letter of Intent*. [Online; accessed 20. Sep. 2021]. Sept. 2021. URL: https://www.dfg.de/download/pdf/foerderung/programme/nfdi/absichtserklaerungen_2021/2021_nfdi_4bioimage.pdf.

[8] Embl-Ebi. *What is data management? Bringing data to life*. [Online; accessed 17. Mar. 2021]. Mar. 2021. URL: https://www.ebi.ac.uk/training/online/courses/bringing-data-life-data-management-biomolecular-sciences/what-is-data-management.

[9] Andreas von der Dunk and Torsten Gille. "Ohne Fundament geht nichts : Forschungsdatenmanagement in der Praxis". In: *Forschung Lehre* 27(2020) (2020), pp. 922–923. ISSN: 0945-5604.

[10] Philippa C. Griffin et al. "Best practice data life cycle approaches for the life sciences". In: *F1000Research* 6 (2017). DOI: 10.12688/f1000research.12344.2.

[11] ANDS. *FAIR data training*. [Online; accessed 4. May 2021]. May 2021. URL: https://www.ands.org.au/working-with-data/fairdata/training.

[12] Mark D. Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific Data* 3.1 (2016), p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[13] Simon Hodson et al. “Turning FAIR data into reality: interim report from the European Commission Expert Group on FAIR data”. In: (June 2018). DOI: 10.5281/zenodo.1285272.

[14] GO FAIR. *FAIRification Process - GO FAIR*. [Online; accessed 17. Mar. 2021]. Apr. 2019. URL: https://www.go-fair.org/fair-principles/fairification-process.

[15] Melissa Linkert et al. “Metadata matters: access to image data in the real world”. In: *Journal of Cell Biology* 189.5 (May 2010), pp. 777–782. ISSN: 0021-9525. DOI: 10.1083/jcb.201004104.

[16] Maximiliaan Huisman et al. “A perspective on Microscopy Metadata: data provenance and quality control”. In: (2021). arXiv: 1910.11370 [q-bio.QM].

[17] Ugis Sarkans et al. “REMBI: Recommended Metadata for Biological Images—enabling reuse of microscopy data in biology - Nature Methods”. In: *Nat. Methods* (May 2021), pp. 1–5. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01166-8.

[18] Wikipedia. *Sample - Wikipedia*. [Online; accessed 27. Apr. 2021]. Dec. 2020. URL: https://en.wikipedia.org/w/index.php?title=Sample&oldid=992147980.

[19] Eleanor Williams et al. “Image Data Resource: a bioimage data integration and publication platform”. In: *Nat. Methods* 14 (Aug. 2017), pp. 775–781. ISSN: 1548-7105. DOI: 10.1038/nmeth.4326.

[20] ISO. *ISO 23081-1:2017(en), Information and documentation — Records management processes — Metadata for records — Part 1: Principles*. [Online; accessed 4. May 2021]. May 2021. URL: https://www.iso.org/obp/ui/fr/#iso:std:iso:23081:-1:ed-2:v1:en.

[21] ISO Committee. *Building a metadata schema – where to start*. [Online; accessed 4. May 2021]. Aug. 2015. URL: https://committee.iso.org/files/live/sites/tc46sc11/files/documents/N800R1%20Where%20to%20start-advice%20on%20creating%20a%20metadata%20schema.pdf.

[22] Ontola. *A brief introduction to linked data*. [Online; accessed 10. Aug. 2021]. July 2018. URL: https://ontola.io/what-is-linked-data.

[23] S. Decker, P. Mitra, and S. Melnik. “Framework for the semantic Web: an RDF tutorial”. In: *IEEE Internet Computing* 4.6 (2000), pp. 68–73. DOI: 10.1109/4236.895018.

[24] Ontola. *What’s the best RDF serialization format?* [Online; accessed 11. Aug. 2021]. June 2019. URL: https://ontola.io/blog/rdf-serialization-formats.

[25] Nicola Guarino and Christopher Welty†. “A Formal Ontology of Properties”. In: *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*. Berlin, Germany: Springer, July 2002, pp. 97–112. ISBN: 978-3-540-41119-2. DOI: 10.1007/3-540-39967-4_8.

[26] Melanie Courtot. *Ontologies for life sciences: examples from the gene ontology*. [Online; accessed 2. May 2021]. May 2021. URL: https://www.slideshare.net/mcourtot/ontologies-for-life-sciences-examples-from-the-gene-ontology.

[27] NCBO BioPortal. *Welcome to the NCBO BioPortal*. [Online; accessed 5. May 2021]. May 2021. URL: https://bioportal.bioontology.org.

[28] Ontology Xref Service. *Ontology Lookup Service < EMBL-EBI*. [Online; accessed 5. May 2021]. May 2021. URL: https://www.ebi.ac.uk/ols/index.

[29] Jason R. Swedlow. "The Open Microscopy Environment: A Collaborative Data Modeling and Software Development Project for Biological Image Informatics". In: *Imaging Cellular and Molecular Biological Functions*. Berlin, Germany: Springer, 2007, pp. 71–92. ISBN: 978-3-540-71330-2. DOI: 10.1007/978-3-540-71331-9_3.

[30] Jason R. Swedlow et al. "Informatics and Quantitative Analysis in Biological Imaging". In: *Science* 300.5616 (2003), pp. 100–102. ISSN: 0036-8075. DOI: 10.1126/science.1082602.

[31] Ilya G. Goldberg et al. "The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging". In: *Genome Biol.* 6.5 (May 2005), pp. 1–13. ISSN: 1474-760X. DOI: 10.1186/gb-2005-6-5-r47.

[32] OME. *OME Data Model and File Formats 6.0.1 Documentation*. [Online; accessed 18. Mar. 2021]. May 2019. URL: https://docs.openmicroscopy.org/ome-model/6.0.1.

[33] Mathias Hammer et al. "Towards community-driven metadata standards for light microscopy: tiered specifications extending the OME model". In: *bioRxiv* (Apr. 2021), p. 2021.04.25.441198. URL: https://doi.org/10.1101/2021.04.25.441198.

[34] Norio Kobayashi et al. *OME Core Ontology: An OWL-based Life Science Imaging Data Model*. [Online; accessed 22. Sep. 2021]. Apr. 2021. URL: http://ceur-ws.org/Vol-2849/paper-25.pdf.

[35] Norio Kobayashi et al. "RIKEN MetaDatabase: A Database Platform for Health Care and Life Sciences as a Microcosm of Linked Open Data Cloud". In: *IJSWIS* 14.1 (Jan. 2018), pp. 140–164. DOI: 10.4018/IJSWIS.2018010106.

[36] OME. *The OME-TIFF format — OME Data Model and File Formats 6.2.2 Documentation*. [Online; accessed 6. May 2021]. Dec. 2020. URL: https://docs.openmicroscopy.org/ome-model/6.2.2/ome-tiff/index.html.

[37] OME. *Bio-Formats Documentation — Bio-Formats 6.6.1 Documentation*. [Online; accessed 14. May 2021]. Mar. 2021. URL: https://docs.openmicroscopy.org/bio-formats/6.6.1.

[38] OME. *OMERO clients overview — OMERO 5.6.3 Documentation*. [Online; accessed 14. May 2021]. Sept. 2020. URL: https://docs.openmicroscopy.org/omero/5.6.3/users/clients-overview.html.

[39] OME. *Groups and permissions system — OMERO 5.6.3 Documentation*. [Online; accessed 9. Aug. 2021]. Sept. 2020. URL: https://docs.openmicroscopy.org/omero/5.6.3/sysadmins/server-permissions.html.

[40] OME. *Import metadata using the Command Line Interface (CLI) — OMERO guide 0.2.0 Documentation*. [Online; accessed 10. Aug. 2021]. July 2021. URL: https://omero-guides.readthedocs.io/en/latest/upload/docs/metadata.html.

[41] OME. *The Open Microscopy Environment*. [Online; accessed 12. Aug. 2021]. Aug. 2021. URL: https://www.openmicroscopy.org/omero/developers.

[42] Chris Allan et al. "OMERO: flexible, model-driven data management for experimental biology - Nature Methods". In: *Nat. Methods* 9 (Mar. 2012), pp. 245–253. ISSN: 1548-7105. DOI: 10.1038/nmeth.1896.

[43] Colin Blackburn et al. "The Open Microscopy Environment: open image informatics for the biological sciences". In: *Software and Cyberinfrastructure for Astronomy IV*. Ed. by Gianluca Chiozzi and Juan C. Guzman. Vol. 9913. International Society for Optics and Photonics. SPIE, 2016, pp. 823–830. URL: https://doi.org/10.1117/12.2232291.

[44] NGINX. *Beginner's Guide*. [Online; accessed 18. Aug. 2021]. July 2021. URL: http://nginx.org/en/docs/beginners_guide.html.

[45] WSGI. *Documentation*. [Online; accessed 9. Sep. 2021]. Jan. 2021. URL: https://wsgi.readthedocs.io/en/latest.

[46] OME. *OMERO.web framework — OMERO 5.6.3 documentation*. [Online; accessed 13. Aug. 2021]. Sept. 2020. URL: https://docs.openmicroscopy.org/omero/5.6.3/developers/Web.html.

[47] DFG. *Leitlinien zum Umgang mit Forschungsdaten*. [Online; accessed 28. Aug. 2021]. Aug. 2021. URL: https://www.dfg.de/download/pdf/foerderung/grundlagen_dfg_foerderung/forschungsdaten/leitlinien_forschungsdaten.pdf.

[48] OME. *Workshop 2018*. [Online; accessed 31. Aug. 2021]. June 2018. URL: https://downloads.openmicroscopy.org/presentations/2018/ELMI-Dublin/OMERO-Workshop/#/3.

[49] Susanne Kunis. *Using OMERO.mde to edit ome metadata — OMERO guide 0.2.0 documentation*. [Online; accessed 16. Sep. 2021]. Sept. 2021. URL: https://omero-guides.readthedocs.io/en/latest/mde/docs/mde_editStandard.html.

[50] Douglas P. W. Russell and Peter K. Sorger. "Maintaining the provenance of microscopy metadata using OMERO.forms software". In: *bioRxiv* (Feb. 2017), p. 109199. DOI: 10.1101/109199.

[51] OME. *omero-metadata*. [Online; accessed 21. Sep. 2021]. Sept. 2021. URL: https://github.com/ome/omero-metadata.

[52] Johnny Hay et al. "PyOmeroUpload: A Python toolkit for uploading images and metadata to OMERO". In: *Wellcome Open Research* 5 (Aug. 2020), p. 96. DOI: 10.12688/wellcomeopenres.15853.2.

[53] Josh Moore et al. “On Bringing Bioimaging Data into the Open (World)”. In: Dec. 2019, pp. 44–53. URL: http://ceur-ws.org/Vol-2849/#paper-06.

Class Diagrams MDEmic A

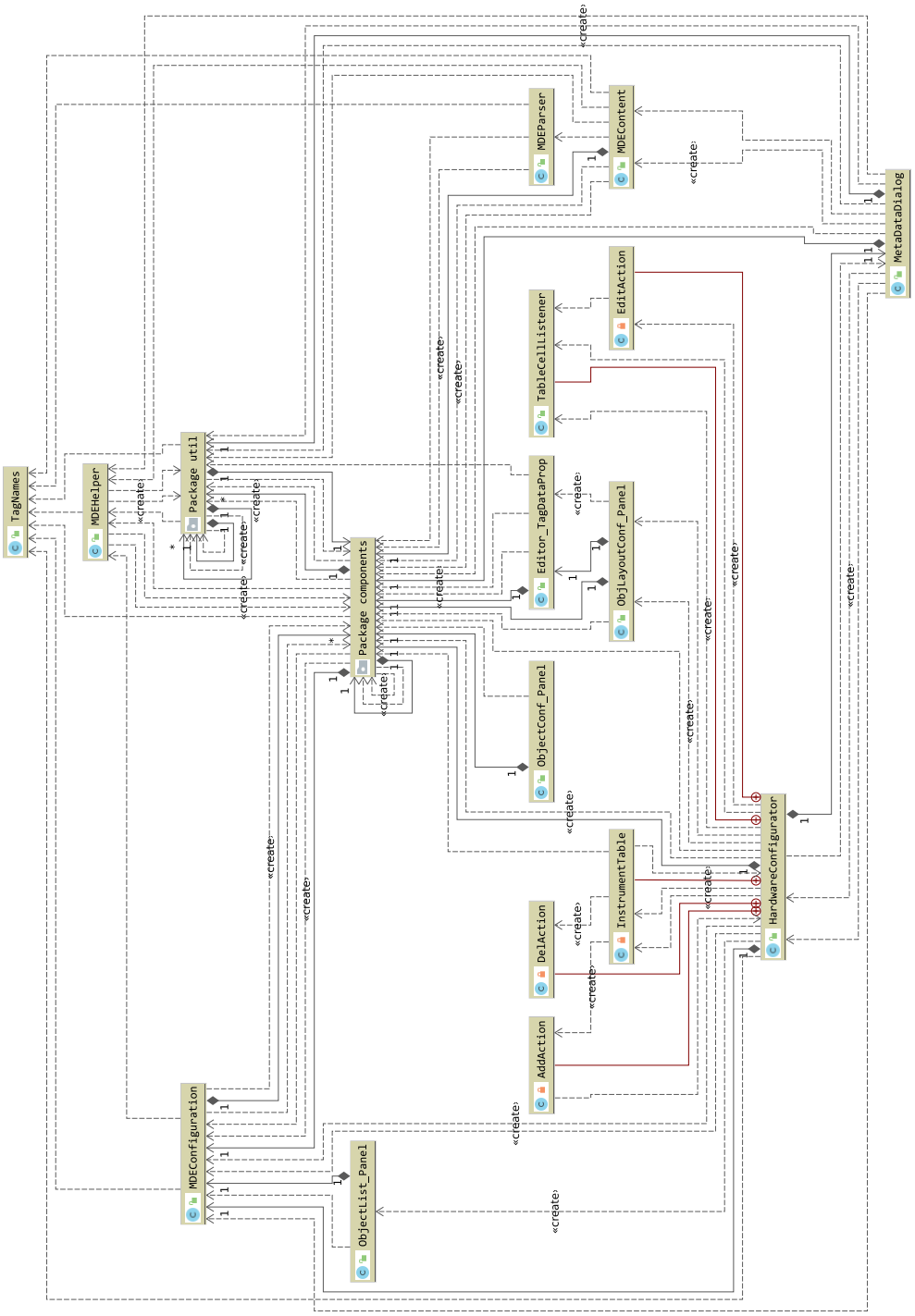

Figure A.1. Class diagram MDEmic package mde.

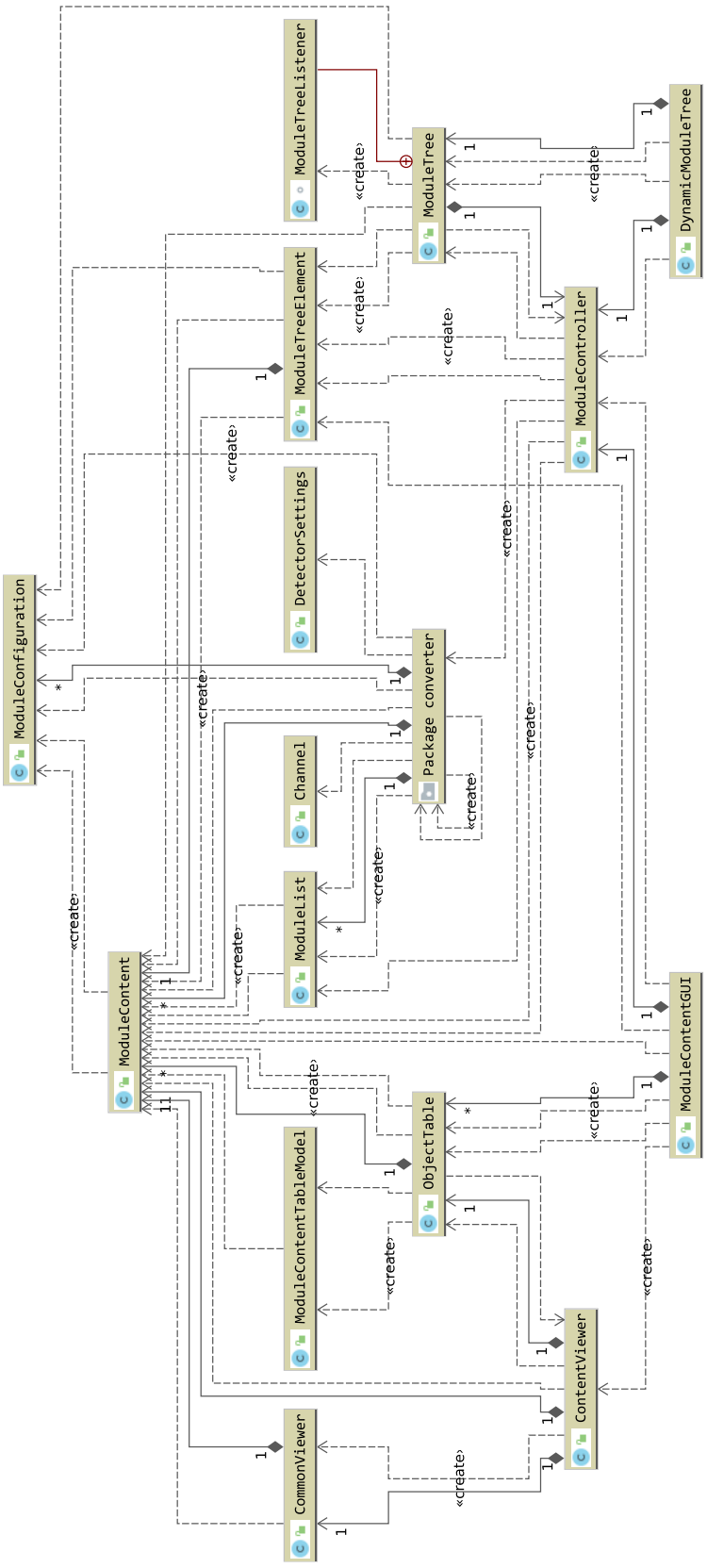

Figure A.2. Class diagram MDEmic package components.

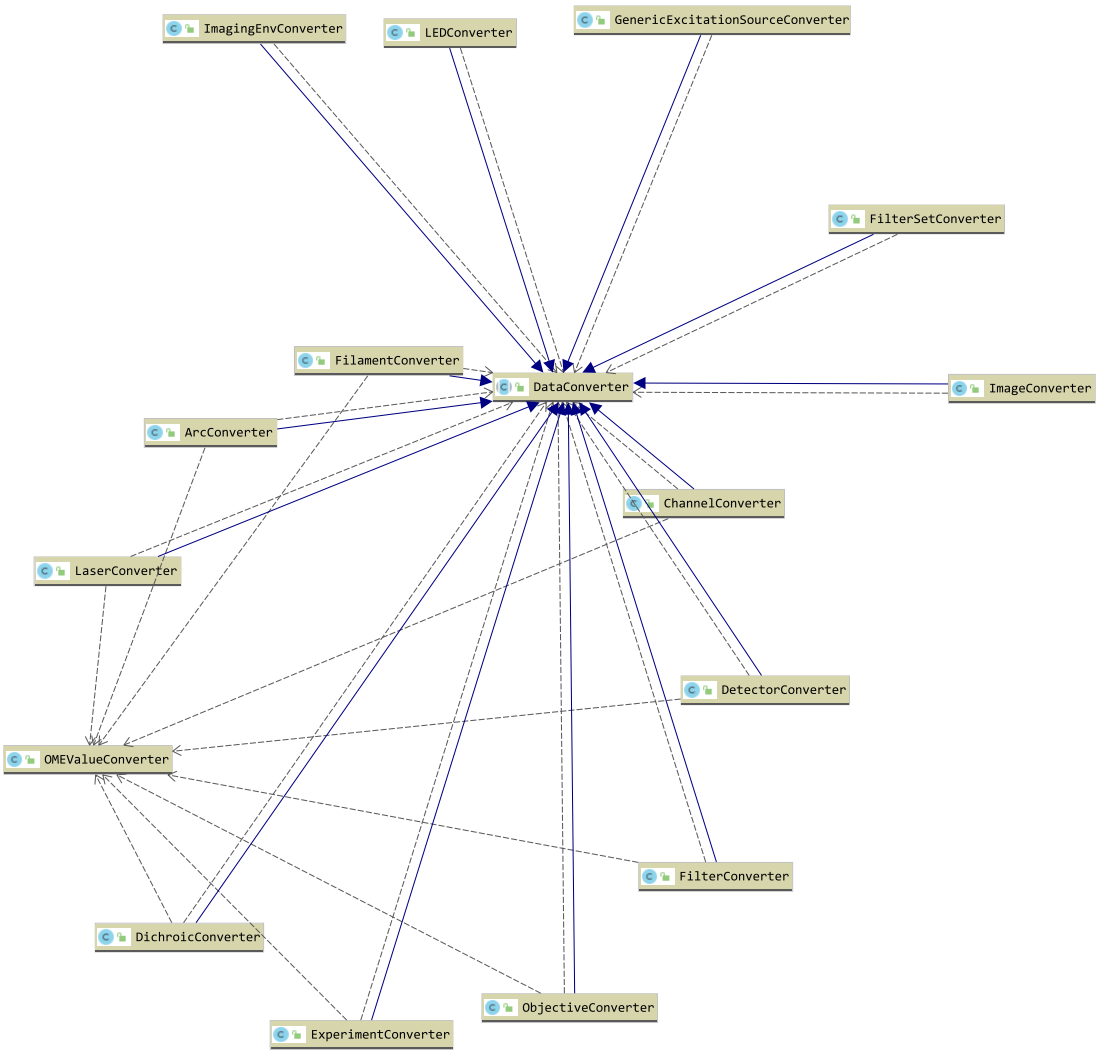

Figure A.3. Class diagram MDEmic package converter.

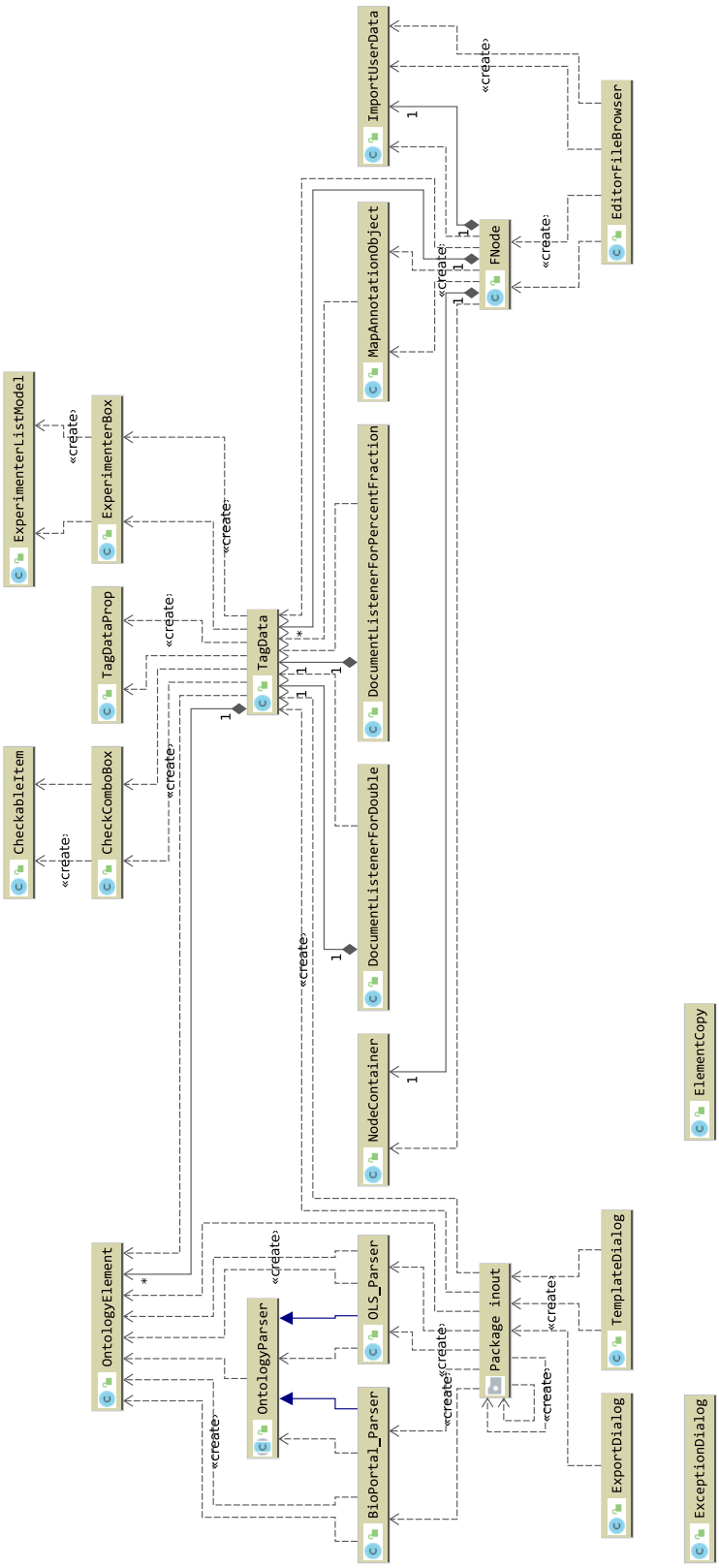

Figure A.4. Class diagram MDEmic package util.

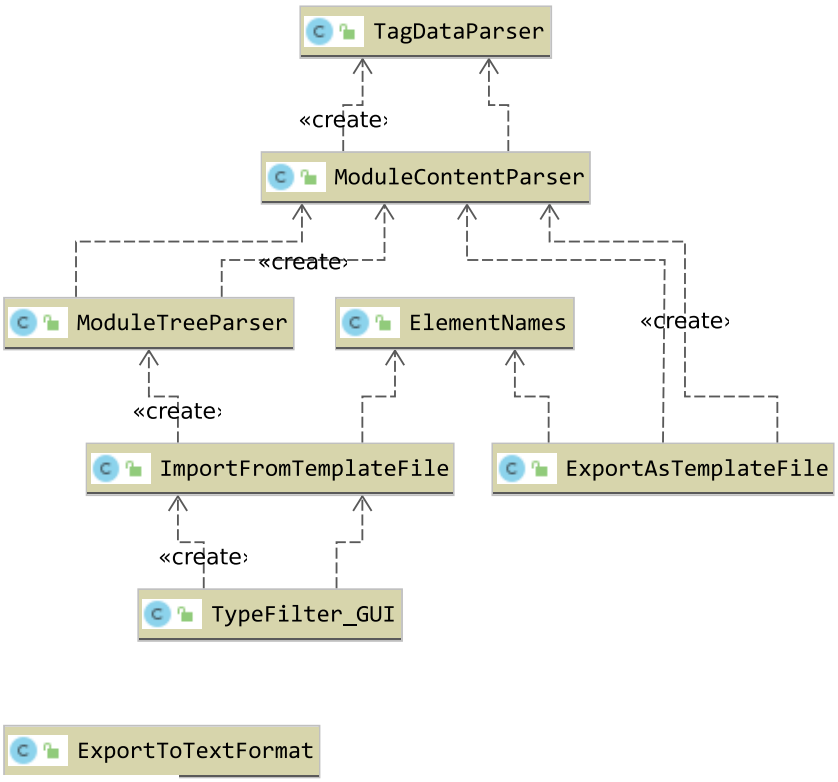

Figure A.5. Class diagram MDEmic package inout.

Example Configuration File MDEmic

B

```
<?xml version="1.0" encoding="UTF-8"?>
<MDEConfiguration>
    <MDEPredefinitions>
        <SetupPre Name="Universal" />
        <SetupPre Name="MyCustomSetup">
            <ObjectPre Type="MyCustomObject">
                <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
                         Unit="" Value="Example Value 1A" Visible="true" />
                <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
                         Unit="" Value="Example Value 2A" Visible="true" />
            </ObjectPre>
            <ObjectPre Type="MyCustomObject">
                <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
                         Unit="" Value="Example Value 1B" Visible="true" />
                <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
                         Unit="" Value="Example Value 2B" Visible="true" />
            </ObjectPre>
        </SetupPre>
        <SetupPre Name="Example Setup: Fields">
            <ObjectPre Type="Available InputFields" >
                <TagData DefaultValues="2" Name="Tag of Type ArrayField"
                         Type="ArrayField" Unit="" Value="3,4" Visible="true" />
                <TagData DefaultValues="3" Name="Tag of Type ArrayField with Unit"
                         Type="ArrayField" Unit="s" Value="3,4,6" Visible="true" />
                <TagData DefaultValues="" Name="Tag of Type TextArea" Type="
                    TextArea"
                         Unit="" Value="this is a text" Visible="true" />
                <TagData DefaultValues="" Name="Tag of Type TextField" Type="
                    TextField"
                         Unit="" Value="this is also a text" Visible="true" />
                <TagData DefaultValues="" Name="Tag of Type TextField with Unit"
                    Type="TextField"
                         Unit="mm" Value="millimeter value" Visible="true" />
                <TagData DefaultValues="Value1,Value2,Value3" Name="Tag of Type
                    ComboBox" Type="ComboBox"
                         Unit="" Value="Value1" Visible="true" />
                <TagData DefaultValues="" Name="Tag of Type ComboBox val from
                    ontology href" Type="ComboBox"
                         Unit="" Value="" Visible="true">
                    <Ontology URL_restapi="http://data.bioontology.org" Acronym="
                        EFO" ID_href="http://purl.obolibrary.org/obo/OBI_0000272" /
                        >
                </TagData>
                <TagData DefaultValues="Value1,Value2,Value3" Name="Tag of Type
                    CheckComboBox" Type="CheckComboBox"
                         Unit="" Value="Value1" Visible="true" />
                <TagData DefaultValues="" Name="Tag of Type TimeStamp" Type="
                    TimeStamp"
                         Unit="" Value="2020-01-01" Visible="true" />
            </ObjectPre>
            <ObjectPre ID="" Type="OME:Objective">
                <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
                         Value="" Visible="true" />
```

```
45                <TagData DefaultValues="" Name="Model" Type="TextField"
46                         Unit="" Value="HCX PL APO CS 40x/0.75-1.25" Visible="true"
                               />
47                <TagData DefaultValues="" Name="Manufacturer"
48                         Type="TextField" Unit="" Value="Leica" Visible="true" />
49                <TagData DefaultValues="" Name="Nominal Magnification"
50                         Type="TextField" Unit="" Value="40.0" Visible="true" />
51                <TagData DefaultValues="" Name="Calibration Magnification"
52                         Type="TextField" Unit="" Value="40.0" Visible="true" />
53                <TagData DefaultValues="" Name="Lens NA" Type="TextField"
54                         Unit="" Value="" Visible="true" />
55                <TagData
56                         DefaultValues="Oil,Water,WaterDipping,Air,Multi,Glycerol,
                             Other"
57                         Name="Immersion" Type="ComboBox" Unit="" Value="Oil"
                             Visible="true" />
58                <TagData
59                         DefaultValues="UV,PlanApo,PlanFluor,SuperFluor,
                             VioletCorrected,Achro,Achromat,Fluor,Fl,Fluar,Neofluar,
                             Fluotar,Apo,PlanNeofluar,Other"
60                         Name="Correction" Type="ComboBox" Unit="" Value="PlanApo"
61                         Visible="true" />
62                <TagData DefaultValues="" Name="Working Distance"
63                          Type="TextField" Unit=" m " Value="220.0" Visible="true" /
                               >
64                <TagData DefaultValues="" Name="Iris" Type="TextField"
65                          Unit="" Value="true" Visible="true" />
66                <TagData DefaultValues="" Name="User::Refraction Index"
67                         Type="TextField" Unit="" Value="" Visible="true" />
68                <TagData DefaultValues="Air,Oil,Water,Glycerol,Other"
69                         Name="User::Medium" Type="ComboBox" Unit="" Value=""
                              Visible="true" />
70                <TagData DefaultValues="" Name="User::Correction Collar"
71                         Type="TextField" Unit="" Value="" Visible="true" />
72            </ObjectPre>
73        </SetupPre>
74    </MDEPredefinitions>
75    <MDEObjects>
76        <Definitions>
77            <ObjectDef Type="Study Info">
78                <TagData DefaultValues="" Name="Study Type" Type="TextField"
79                         Unit="" Value="" Visible="true" />
80                <TagData DefaultValues="Arabidopsis thaliana,Danio rerio,Drosophila
                              melanogaster,Gallus gallus,Homo sapiens,Mus musculus, Mus
                             musculus x Mus spretus,Saccharomyces cerevisiae,
                             Schizosaccharomyces pombe" Name="Organism" Type="ComboBox"
81                         Unit="" Value="Homo sapiens" Visible="true" />
82                <TagData DefaultValues="" Name="Experiment Type" Type="TextField"
83                         Unit="" Value="" Visible="true" />
84                <TagData DefaultValues="" Name="Imaging Method" Type="TextField"
85                         Unit="" Value="" Visible="true" />
86                <TagData DefaultValues="" Name="Publication Title" Type="TextField"
87                         Unit="" Value="" Visible="true" />
88                <TagData DefaultValues="" Name="Publication Authors" Type="TextArea
                             "
89                         Unit="" Value="" Visible="true" />
90                <TagData DefaultValues="" Name="PubMed ID" Type="TextField"
91                         Unit="" Value="" Visible="true" />
92                <TagData DefaultValues="" Name="PMC ID" Type="TextField"
93                         Unit="" Value="" Visible="true" />
94                <TagData DefaultValues="" Name="Publication DOI" Type="TextField"
95                         Unit="" Value="" Visible="true" />
96                <TagData DefaultValues="" Name="License" Type="TextField"
97                         Unit="" Value="" Visible="true" />
98                <TagData DefaultValues="" Name="Data Publisher" Type="TextField"
99                         Unit="" Value="" Visible="true" />
100               <TagData DefaultValues="" Name="Data DOI" Type="TextField"
101                        Unit="" Value="" Visible="true" />
102               <Parents Values="OME-Model" />
103           </ObjectDef>
```

```
104             <ObjectDef Type="OME:Detector">
105                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
106                          Value="" Visible="true" />
107                 <TagData DefaultValues="" Name="Model" Type="TextField"
108                          Unit="" Value="" Visible="true" />
109                 <TagData DefaultValues="" Name="Manufacturer"
110                          Type="TextField" Unit="" Value="" Visible="true" />
111                 <TagData
112                         DefaultValues="CCD,IntensifiedCCD,AnalogVideo,PMT,
                                Photodiode,Spectroscopy,LifetimeImaging,
                                CorrelationSpectroscopy,FTIR,EMCCD,APD,CMOS,EBCCD,Other
                                "
113                         Name="DetectorType" Type="ComboBox" Unit="" Value=""
                                Visible="true" />
114                 <TagData DefaultValues="" Name="Zoom" Type="TextField"
115                          Unit="" Value="" Visible="true" />
116                 <TagData DefaultValues="" Name="AmplificationGain"
117                          Type="TextField" Unit="" Value="" Visible="true" ></
                              TagData>
118                 <TagData DefaultValues="" Name="User::Voltage"
119                          Type="TextField" Unit="V" Value="" Visible="true" />
120                 <TagData DefaultValues="" Name="User::Offset"
121                          Type="TextField" Unit="" Value="" Visible="true" />
122                 <TagData DefaultValues="" Name="User::Gain" Type="TextField"
123                          Unit="" Value="" Visible="true" />
124                 <TagData DefaultValues="" Name="User::Confocal Zoom"
125                          Type="TextField" Unit="" Value="" Visible="true" />
126                 <TagData DefaultValues="1x1,2x2,4x4,8x8,Other"
127                          Name="User::Binning" Type="ComboBox" Unit="" Value=""
128                          Visible="true" />
129                 <TagData DefaultValues="" Name="User::Subarray"
130                          Type="TextField" Unit="" Value="" Visible="true" />
131                 <Parents Values="OME:Channel" />
132             </ObjectDef>
133             <ObjectDef Type="OME:Laser">
134                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
135                          Value="" Visible="true" />
136                 <TagData DefaultValues="" Name="Manufacturer"
137                          Type="TextField" Unit="" Value="" Visible="true" />
138                 <TagData DefaultValues="" Name="Model" Type="TextField"
139                          Unit="" Value="" Visible="true" />
140                 <TagData DefaultValues="" Name="Power" Type="TextField"
141                          Unit="mW" Value="" Visible="true" />
142                 <TagData
143                         DefaultValues="Excimer,Gas,MetalVapor,SolidState,Dye,
                                Semiconductor,FreeElectron,Other"
144                         Name="L_Type" Type="ComboBox" Unit="" Value="" Visible="
                                true" />
145                 <TagData
146                         DefaultValues="Cu,Ag,ArFl,ArCl,KrFl,KrCl,XeFl,XeCl,XeBr,N,
                                Ar,Kr,Xe,HeNe,HeCd,CO,CO2,H2O,HFl,NdGlass,NdYAG,ErGlass
                                ,ErYAG,HoYLF,HoYAG,Ruby,TiSapphire,Alexandrite,
                                Rhodamine6G,CoumarinC30,GaAs,GaAlAs,EMinus,Other"
147                         Name="Laser Medium" Type="ComboBox" Unit="" Value=""
                                Visible="true" />
148                 <TagData DefaultValues="" Name="Frequency Multiplication"
149                          Type="TextField" Unit="" Value="" Visible="true" />
150                 <TagData DefaultValues="true,false" Name="Tunable"
151                          Type="ComboBox" Unit="" Value="" Visible="true" />
152                 <TagData
153                         DefaultValues="CW,Single,QSwitched,Repetitive,ModeLocked,
                                Other"
154                         Name="Pulse" Type="ComboBox" Unit="" Value="" Visible="true
                                " />
155                 <TagData DefaultValues="true,false" Name="Pockel Cell"
156                          Type="ComboBox" Unit="" Value="" Visible="true" />
157                 <TagData DefaultValues="" Name="Repititation Rate"
158                          Type="TextField" Unit="MHz" Value="" Visible="true" />
159                 <TagData DefaultValues="" Name="Pump" Type="TextField"
160                          Unit="" Value="" Visible="true" />
```

```
161                 <TagData DefaultValues="" Name="Wavelength" Type="TextField"
162                          Unit="nm" Value="" Visible="true" />
163                 <TagData DefaultValues="" Name="User::Wavelength"
164                          Type="TextField" Unit="nm" Value="" Visible="true" />
165                 <TagData DefaultValues="" Name="User::Attenuation"
166                          Type="TextField" Unit="" Value="" Visible="true" />
167                 <Parents Values="OME:LightSource" />
168             </ObjectDef>
169             <ObjectDef Type="OME:Arc">
170                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
171                          Value="" Visible="true" />
172                 <TagData DefaultValues="" Name="Manufacturer"
173                          Type="TextField" Unit="" Value="" Visible="true" />
174                 <TagData DefaultValues="" Name="Model" Type="TextField"
175                          Unit="" Value="" Visible="true" />
176                 <TagData DefaultValues="Hg,Xe,HgXe,Other" Name="A_Type"
177                          Type="ComboBox" Unit="" Value="" Visible="true" />
178                 <TagData DefaultValues="" Name="User::Wavelength"
179                          Type="TextField" Unit="nm" Value="" Visible="true" />
180                 <TagData DefaultValues="" Name="User::Attenuation"
181                          Type="TextField" Unit="" Value="" Visible="true" />
182                 <Parents Values="OME:LightSource" />
183             </ObjectDef>
184             <ObjectDef Type="OME:Objective">
185                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
186                          Value="" Visible="true" />
187                 <TagData DefaultValues="" Name="Model" Type="TextField"
188                          Unit="" Value="" Visible="true" />
189                 <TagData DefaultValues="" Name="Manufacturer"
190                          Type="TextField" Unit="" Value="" Visible="true" />
191                 <TagData DefaultValues="" Name="Nominal Magnification"
192                          Type="TextField" Unit="" Value="" Visible="true" />
193                 <TagData DefaultValues="" Name="Calibration Magnification"
194                          Type="TextField" Unit="" Value="" Visible="true" />
195                 <TagData DefaultValues="" Name="Lens NA" Type="TextField"
196                          Unit="" Value="" Visible="true" />
197                 <TagData
198                         DefaultValues="Oil,Water,WaterDipping,Air,Multi,Glycerol,
                                Other"
199                         Name="Immersion" Type="ComboBox" Unit="" Value="" Visible="
                                true" />
200                 <TagData
201                         DefaultValues="UV,PlanApo,PlanFluor,SuperFluor,
                                VioletCorrected,Achro,Achromat,Fluor,Fl,Fluar,Neofluar,
                                Fluotar,Apo,PlanNeofluar,Other"
202                         Name="Correction" Type="ComboBox" Unit="" Value="" Visible=
                                "true" />
203                 <TagData DefaultValues="" Name="Working Distance"
204                          Type="TextField" Unit=" m " Value="" Visible="true" />
205                 <TagData DefaultValues="" Name="Iris" Type="TextField"
206                          Unit="" Value="" Visible="true" />
207                 <TagData DefaultValues="" Name="User::Refraction Index"
208                          Type="TextField" Unit="" Value="" Visible="true" />
209                 <TagData DefaultValues="Air,Oil,Water,Glycerol,Other"
210                          Name="User::Medium" Type="ComboBox" Unit="" Value=""
                                 Visible="true" />
211                 <TagData DefaultValues="" Name="User::Correction Collar"
212                          Type="TextField" Unit="" Value="" Visible="true" />
213                 <Parents Values="OME:Image" />
214             </ObjectDef>
215             <ObjectDef Type="OME:EmissionFilter">
216                 <Parents Values="OME:LightPath" />
217             </ObjectDef>
218             <ObjectDef Type="OME:Filter">
219                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
220                          Value="" Visible="true" />
221                 <TagData DefaultValues="" Name="Model" Type="TextField"
222                          Unit="" Value="" Visible="true" />
223                 <TagData DefaultValues="" Name="Manufacturer"
224                          Type="TextField" Unit="" Value="" Visible="true" />
```

```
225                 <TagData
226                         DefaultValues="Dichroic,LongPass,ShortPass,BandPass,
                                MultiPass,NeutralDensity,Tuneable,Other"
227                         Name="FilterType" Type="ComboBox" Unit="" Value="" Visible=
                                "true" />
228                 <TagData DefaultValues="" Name="Filterwheel"
229                          Type="TextField" Unit="" Value="" Visible="true" />
230                 <TagData DefaultValues="" Name="CutIn" Type="TextField"
231                          Unit="nm" Value="" Visible="true" />
232                 <TagData DefaultValues="" Name="CutOut" Type="TextField"
233                          Unit="nm" Value="" Visible="true" />
234                 <Parents Values="OME:EmissionFilter,OME:ExcitationFilter" />
235             </ObjectDef>
236             <ObjectDef Type="OME:ImagingEnvironment">
237                 <TagData DefaultValues="" Name="Temperature"
238                          Type="TextField" Unit=" C " Value="" Visible="true" />
239                 <TagData DefaultValues="" Name="Air Pressure"
240                          Type="TextField" Unit="Mbar" Value="" Visible="true" />
241                 <TagData DefaultValues="" Name="Humidity %" Type="TextField"
242                          Unit="" Value="" Visible="true" />
243                 <TagData DefaultValues="" Name="CO2 Percent %"
244                          Type="TextField" Unit="" Value="" Visible="true" />
245                 <Parents Values="OME:Image" />
246             </ObjectDef>
247             <ObjectDef Type="OME:ExcitationFilter">
248                 <Parents Values="OME:LightPath" />
249             </ObjectDef>
250             <ObjectDef Type="OME:Dichroic">
251                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
252                          Value="" Visible="true" />
253                 <TagData DefaultValues="" Name="Model" Type="TextField"
254                          Unit="" Value="" Visible="true" />
255                 <TagData DefaultValues="" Name="Manufacturer"
256                          Type="TextField" Unit="" Value="" Visible="true" />
257                 <Parents Values="OME:EmissionFilter,OME:ExcitationFilter,
                            OME:LightPath" />
258             </ObjectDef>
259             <ObjectDef Type="OME:LightEmittingDiode">
260                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
261                          Value="" Visible="true" />
262                 <TagData DefaultValues="" Name="Manufacturer"
263                          Type="TextField" Unit="" Value="" Visible="true" />
264                 <TagData DefaultValues="" Name="Model" Type="TextField"
265                          Unit="" Value="" Visible="true" />
266                 <TagData DefaultValues="" Name="Description" Type="TextArea"
267                          Unit="" Value="" Visible="true" />
268                 <TagData DefaultValues="" Name="User::Wavelength"
269                          Type="TextField" Unit="nm" Value="" Visible="true" />
270                 <TagData DefaultValues="" Name="User::Attenuation"
271                          Type="TextField" Unit="" Value="" Visible="true" />
272                 <Parents Values="OME:LightSource" />
273             </ObjectDef>
274             <ObjectDef Type="OME:Channel">
275                 <TagData DefaultValues="" Name="Name" Type="TextField"
276                          Unit="" Value="" Visible="true" />
277                 <TagData DefaultValues="" Name="Color" Type="TextField"
278                          Unit="" Value="" Visible="true" />
279                 <TagData DefaultValues="" Name="Fluorophore"
280                          Type="TextField" Unit="" Value="" Visible="true" />
281                 <TagData
282                         DefaultValues="Transmitted,Epifluorescence,Oblique,
                                NonLinear,Other"
283                         Name="Illumination Type" Type="ComboBox" Unit="" Value=""
284                         Visible="true" />
285                 <TagData DefaultValues="" Name="Exposure Time"
286                          Type="TextField" Unit="s" Value="" Visible="true" />
287                 <TagData DefaultValues="" Name="Excitation Wavelength"
288                          Type="TextField" Unit="nm" Value="" Visible="true" />
289                 <TagData DefaultValues="" Name="Emission Wavelength"
290                          Type="TextField" Unit="nm" Value="" Visible="true" />
```

```
291                 <TagData
292                         DefaultValues="WideField,LaserScanningConfocalMicroscopy,
                                SpinningDiskConfocal,SlitScanConfocal,
                                MultiPhotonMicroscopy,StructuredIllumination,
                                SingleMoleculeImaging,TotalInternalReflection,
                                FluorescenceLifetime,SpectralImaging,
                                FluorescenceCorrelationSpectroscopy,
                                NearFieldScanningOpticalMicroscopy,
                                SecondHarmonicGenerationImaging,PALM,STORM,STED,TIRF,
                                FSM,LCM,Other,BrightField,SweptFieldConfocal,SPIM"
293                         Name="Imaging Mode" Type="ComboBox" Unit="" Value=""
                                Visible="true" />
294                 <TagData
295                         DefaultValues="Brightfield,Phase,DIC,HoffmanModulation,
                                ObliqueIllumination,PolarizedLight,Darkfield,
                                Fluorescence,Other"
296                         Name="Contrast Method" Type="ComboBox" Unit="" Value=""
297                         Visible="true" />
298                 <TagData DefaultValues="" Name="ND Filter" Type="TextField"
299                          Unit="" Value="" Visible="true" />
300                 <TagData DefaultValues="" Name="Pinhole Size"
301                          Type="TextField" Unit=" m " Value="" Visible="true" />
302                 <Parents Values="OME:Image" />
303             </ObjectDef>
304             <ObjectDef Type="OME:LightSource">
305                 <Parents Values="OME:Channel" />
306             </ObjectDef>
307             <ObjectDef Type="OME:Experiment">
308                 <TagData DefaultValues="" Name="Description" Type="TextArea"
309                          Unit="" Value="" Visible="true" />
310                 <TagData
311                         DefaultValues="FP,FRET,TimeLapse,FourDPlus,Screen,
                                Immunocytochemistry,Immunofluorescence,FISH,
                                Electrophysiology,IonImaging,Colocalization,
                                PGIDocumentation,FluorescenceLifetime,SpectralImaging,
                                Photobleaching,SPIM,Other"
312                         Name="ExperimentType" Type="ComboBox" Unit="" Value=""
313                         Visible="true" />
314                 <TagData DefaultValues="" Name="Experimenter Name"
315                          Type="TextField" Unit="" Value="" Visible="true" />
316                 <Parents Values="OME-Model" />
317             </ObjectDef>
318             <ObjectDef Type="OME:Filament">
319                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
320                          Value="" Visible="true" />
321                 <TagData DefaultValues="" Name="Manufacturer"
322                          Type="TextField" Unit="" Value="" Visible="true" />
323                 <TagData DefaultValues="" Name="Model" Type="TextField"
324                          Unit="" Value="" Visible="true" />
325                 <TagData DefaultValues="" Name="Power" Type="TextField"
326                          Unit="mW" Value="" Visible="true" />
327                 <TagData DefaultValues="Incandescent,Halogen,Other"
328                          Name="F_Type" Type="ComboBox" Unit="" Value="" Visible="
                                 true" />
329                 <TagData DefaultValues="" Name="User::Wavelength"
330                          Type="TextField" Unit="nm" Value="" Visible="true" />
331                 <TagData DefaultValues="" Name="User::Attenuation"
332                          Type="TextField" Unit="" Value="" Visible="true" />
333                 <Parents Values="OME:LightSource" />
334             </ObjectDef>
335             <ObjectDef Type="OME:Generic_Excitation_Src">
336                 <TagData DefaultValues="" Name="ID" Type="TextField" Unit=""
337                          Value="" Visible="true" />
338                 <TagData DefaultValues="" Name="Manufacturer"
339                          Type="TextField" Unit="" Value="" Visible="true" />
340                 <TagData DefaultValues="" Name="Model" Type="TextField"
341                          Unit="" Value="" Visible="true" />
342                 <TagData DefaultValues="" Name="Power" Type="TextField"
343                          Unit="mW" Value="" Visible="true" />
344                 <TagData DefaultValues="" Name="Map" Type="TextField"
```

```
345                     Unit="" Value="" Visible="true" />
346                 <TagData DefaultValues="" Name="User::Wavelength"
347                     Type="TextField" Unit="nm" Value="" Visible="true" />
348                 <TagData DefaultValues="" Name="User::Attenuation"
349                     Type="TextField" Unit="" Value="" Visible="true" />
350                 <Parents Values="OME:LightSource" />
351             </ObjectDef>
352             <ObjectDef Type="OME:LightPath">
353                 <Parents Values="OME:Channel" />
354             </ObjectDef>
355             <ObjectDef Type="OME:Image">
356                 <TagData DefaultValues="" Name="Name" Type="TextField"
357                     Unit="" Value="" Visible="true" />
358                 <TagData DefaultValues="" Name="Description"
359                     Type="TextField" Unit="" Value="" Visible="true" />
360                 <TagData DefaultValues="" Name="Acquisition Time"
361                     Type="TimeStamp" Unit="" Value="" Visible="true" />
362                 <TagData DefaultValues="2" Name="Dim X x Y"
363                     Type="ArrayField" Unit="" Value="" Visible="true" />
364                 <TagData DefaultValues="" Name="Pixel Depth"
365                     Type="TextField" Unit="" Value="" Visible="true" />
366                 <TagData DefaultValues="2" Name="Pixel Size (XY)"
367                     Type="ArrayField" Unit=" m " Value="" Visible="true" />
368                 <TagData DefaultValues="3" Name="Dim Z x T x C"
369                     Type="ArrayField" Unit="" Value="" Visible="true" />
370                 <TagData DefaultValues="" Name="Time Increment"
371                     Type="TextField" Unit="ms" Value="" Visible="true" />
372                 <TagData DefaultValues="2" Name="Stage Label (XY)"
373                     Type="ArrayField" Unit="reference frame" Value="" Visible=
                        "true" />
374                 <Parents Values="OME-Model" />
375             </ObjectDef>
376             <ObjectDef Type="Available InputFields">
377                 <TagData DefaultValues="2" Name="Tag of Type ArrayField"
378                     Type="ArrayField" Unit="" Value="" Visible="true" />
379                 <TagData DefaultValues="3" Name="Tag of Type ArrayField with Unit"
380                     Type="ArrayField" Unit="s" Value="" Visible="true" />
381                 <TagData DefaultValues="" Name="Tag of Type TextArea" Type="
                        TextArea"
382                     Unit="" Value="" Visible="true" />
383                 <TagData DefaultValues="" Name="Tag of Type TextField" Type="
                        TextField"
384                     Unit="" Value="" Visible="true" />
385                 <TagData DefaultValues="" Name="Tag of Type TextField with Unit"
                        Type="TextField"
386                     Unit="nm" Value="" Visible="true" />
387                 <TagData DefaultValues="Value1,Value2,Value3" Name="Tag of Type
                        ComboBox" Type="ComboBox"
388                     Unit="" Value="Value1" Visible="true" />
389                 <TagData DefaultValues="Value1,Value2,Value3" Name="Tag of Type
                        CheckComboBox" Type="CheckComboBox"
390                     Unit="" Value="" Visible="true" />
391                 <TagData DefaultValues="" Name="Tag of Type TimeStamp" Type="
                        TimeStamp"
392                     Unit="" Value="" Visible="true" />
393                 <TagData DefaultValues="" Name="Tag of Type ComboBox val from
                        ontology href" Type="ComboBox"
394                     Unit="" Value="" Visible="true">
395                     <Ontology URL_restapi="http://data.bioontology.org" Acronym="
                            EFO" ID_href="http://purl.obolibrary.org/obo/OBI_0000272" /
                            >
396                 </TagData>
397                 <Parents Values="OME-Model" />
398             </ObjectDef>
399             <ObjectDef Type="MyCustomObject">
400                 <TagData DefaultValues="" Name="ExampleKey_1" Type="TextField"
401                     Unit="" Value="" Visible="true" />
402                 <TagData DefaultValues="" Name="ExampleKey_2" Type="TextField"
403                     Unit="" Value="" Visible="true" />
404                 <Parents Values="OME:Image" />
```

```
405             </ObjectDef>
406         </Definitions>
407         <Configurations>
408             <SetupConf Name="Example Setup: Fields">
409                 <ObjectConf Type="OME:Image">
410                     <TagDataProp Name="Name" Unit="" Visible="true" />
411                     <TagDataProp Name="Description" Unit="" Visible="true" />
412                     <TagDataProp Name="Acquisition Time" Unit=""
413                                  Visible="false" />
414                     <TagDataProp Name="Dim X x Y" Unit="" Visible="true" />
415                     <TagDataProp Name="Pixel Depth" Unit="" Visible="true" />
416                     <TagDataProp Name="Pixel Size (XY)" Unit=" m "
417                                  Visible="true" />
418                     <TagDataProp Name="Dim Z x T x C" Unit="" Visible="true" />
419                     <TagDataProp Name="Time Increment" Unit="s"
420                                  Visible="true" />
421                     <TagDataProp Name="Stage Label (XY)"
422                                  Unit="reference frame" Visible="true" />
423                 </ObjectConf>
424                 <ObjectConf Type="OME:Objective">
425                     <TagDataProp DefaultValues="" Name="ID" Type="TextField" Unit="
                                "
426                              Value="" Visible="false" />
427                     <TagDataProp DefaultValues="" Name="Model" Type="TextField"
428                              Unit="" Value="" Visible="false" />
429                     <TagDataProp DefaultValues="" Name="Manufacturer"
430                              Type="TextField" Unit="" Value="" Visible="false" />
431                     <TagDataProp DefaultValues="" Name="Nominal Magnification"
432                              Type="TextField" Unit="" Value="" Visible="false" />
433                     <TagDataProp DefaultValues="" Name="Calibration Magnification"
434                              Type="TextField" Unit="" Value="" Visible="false" />
435                     <TagDataProp DefaultValues="" Name="Lens NA" Type="TextField"
436                              Unit="" Value="" Visible="false" />
437                     <TagDataProp
438                             DefaultValues="Oil,Water,WaterDipping,Air,Multi,
                                    Glycerol,Other"
439                             Name="Immersion" Type="ComboBox" Unit="" Value=""
                                    Visible="false" />
440                     <TagDataProp
441                             DefaultValues="UV,PlanApo,PlanFluor,SuperFluor,
                                    VioletCorrected,Achro,Achromat,Fluor,Fl,Fluar,
                                    Neofluar,Fluotar,Apo,PlanNeofluar,Other"
442                             Name="Correction" Type="ComboBox" Unit="" Value=""
                                    Visible="false" />
443                     <TagDataProp DefaultValues="" Name="Working Distance"
444                              Type="TextField" Unit=" m " Value="" Visible="false" /
                                >
445                     <TagDataProp DefaultValues="" Name="Iris" Type="TextField"
446                              Unit="" Value="" Visible="false" />
447                     <TagDataProp Name="User::Refraction Index" Unit=""  Visible="
                            true" />
448                     <TagDataProp Name="User::Medium"  Unit="" Visible="true" />
449                     <TagDataProp Name="User::Correction Collar" Unit="" Visible="
                            true" />
450                 </ObjectConf>
451
452                 <ObjectConf Type="Available InputFields" Insert="true" InsertPoint=
                        "OME-Model">
453                     <TagDataProp Name="Tag of Type TextField" Unit="" Visible="true
                            " Required="true"/>
454
455                     <TagDataProp Name="Tag of Type ArrayField" Unit=""  Visible="
                            false" />
456                     <TagDataProp Name="Tag of Type ArrayField with Unit" Unit="s"
                            Visible="true" />
457                     <TagDataProp Name="Tag of Type TextArea" Unit="" Visible="false
                            " />
458                 </ObjectConf>
459             </SetupConf>
460             <SetupConf Name="Example Setup: Study Info">
```

```
461                 <ObjectConf Type="OME:Image"/>
462                 <ObjectConf Type="Study Info" Insert="true" InsertPoint="OME-Model"
                        />
463             </SetupConf>
464             <SetupConf Name="MyCustomSetup">
465                 <ObjectConf Type="OME:Image">
466                     <TagDataProp Name ="Description" Unit ="" Visible ="false"/>
467                 </ObjectConf>
468                 <ObjectConf Type="OME:Objective"/>
469                 <ObjectConf Type="MyCustomObject" Insert="true" InsertPoint="
                        OME:Image"/>
470             </SetupConf>
471         </Configurations>
472     </MDEObjects>
473 </MDEConfiguration>
```

Listing B.1 Full example of mdeConfiguration.xml

Related Publications C

Publication and unpublished manuscripts related to this thesis

- Susanne Kunis, Sebastian Hänsch, Christian Schmidt, Frances Wong, Caterina Strambio-De-Castillia, and Stefanie Weidtkamp-Peters. "MDEmic: a metadata annotation tool to facilitate management of FAIR image data in the bioimaging community". In: *Nat. Methods* 18 (Dec. 2021), pp. 1416–1417. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01288-z (Author)

- Glyn Nelson et al. "QUAREP-LiMi: A community-driven initiative to establish guidelines for quality assessment and reproducibility for instruments and images in light microscopy". In: *J. Microsc.* 284.1 (Oct. 2021), pp. 56–73. ISSN: 0022-2720. DOI: 10.1111/jmi.13041 (Co-Author)

- Alex Rigano et al. "Micro-Meta App: an interactive tool for collecting microscopy metadata based on community specifications". In: *Nat. Methods* 18 (Dec. 2021), pp. 1489–1495. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01315-z (Co-Author)

Conferences

- European Light Microscopy Initiative 2021 - Poster:
  Susanne Kunis, Thomas Zobel, Stefanie Weidtkamp-Peters, and Monica Valencia-S. *RDM4mic working group– Research Data Management for microscopy data as a community task*. [Online; accessed 20. Sep. 2021]. Sept. 2021. DOI: 10.22443/rms.elmi2021.187 (Author)

- SWAT4HCLS 2019 - Conference paper:
  Josh Moore, Norio Kobayashi, Susanne Kunis, Shuichi Onami, and Jason R. Swedlow. "On Bringing Bioimaging Data into the Open (World)". In: Dec. 2019, pp. 44–53. URL: http://ceur-ws.org/Vol-2849/#paper-06 (Co-Author)

Erklärung über die Eigenständigkeit der erbrachten wissenschaftlichen Leistung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise entgeltlich / unentgeltlich geholfen.

1 .....

2 .....

3 .....

Weitere Personen waren an der inhaltlichen materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

(Ort, Datum) (Unterschrift)